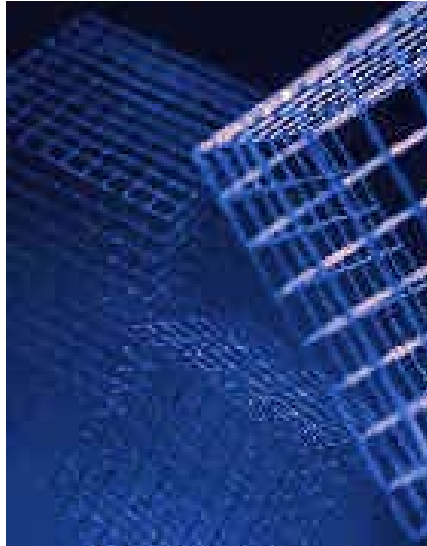


Installation Guide C/C++ Precompiler: SAP DB



Version 7.3








Copyright

© Copyright 2002 SAP AG.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation.

For more information on the GNU Free Documentaton License see <http://www.gnu.org/copyleft/fdl.html#SEC4>.

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Typographic Conventions

Type Style	Description
<i>Example text</i>	Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths and options. Cross-references to other documentation.
Example text	Emphasized words or phrases in body text, titles of graphics and tables.
EXAMPLE TEXT	Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example, SELECT and INCLUDE.
Example text	Screen output. This includes file and directory names and their paths, messages, source code, names of variables and parameters as well as names of installation, upgrade and database tools.
EXAMPLE TEXT	Keys on the keyboard, for example, function keys (such as F2) or the ENTER key.
Example text	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries.

Installation Guide C/C++ Precompiler: SAP DB 7.3.....	5
Installation Guide for Developers.....	5
Installing the C/C++ Precompiler.....	5
Calling the C/C++ Precompiler.....	6
Specifying the Database Instance and the User ID: -d and -u	7
Override Connection to the Database Instance: -Hnocheck	8
Override the C/C++ Compiler: -c.....	8
Linking a C/C++ Program to the Runtime Library	8
Executing a C/C++ Program	9
Example for Compiling a File.....	9
Installation Guide for Users.....	10
Installing the C/C++ Precompiler Runtime Library.....	10
Example: Executing a Precompiler Application	11
Deinstallation.....	12
Using IRTRACE	12
Displaying the Current Trace Settings	12
Modifying the Trace Settings.....	13
Using XUSER.....	13
Using IRCONF	13
Registering the Interface Programs' Runtime Library on the Client.....	13
IRCONF: Options	14
Specify Installation Path: -p	14
Display registered versions of the runtime library: -s	15
Delete runtime library registration: -r	15
Override Driver Name: -d.....	16
Register Runtime Library Version: -i.....	16
Override Version Check: -v.....	17
Specify Additional Key: -k.....	18
Display available options and help: -h.....	18
IRCONF Error Messages	19



Installation Guide C/C++ Precompiler: SAP DB 7.3

[Installation Guide for Developers \[Page 5\]](#)

[Installation Guide for Users \[Page 10\]](#)



This installation guide is **not** applicable to standard installations of SAP systems. The installation of the SAP DB software relevant for SAP systems is described in SAP specific installation guides. For these guides see the SAP Library.

General information on the SAP DB database system can be found in the [User Manual: SAP DB \[Extern\]](#).



Installation Guide for Developers

This installation guide is a brief introduction to using the SAP DB C/C++ Precompiler, written for developers.

If you want to use the SAP DB C/C++ Precompiler exclusively to carry out the installation of precompiler applications, use the [Installation Guide for Users \[Page 10\]](#).

For more detailed information on the features and options included in the precompiler, **see:** *C/C++ Precompiler: SAP DB 7.3*.



Installing the C/C++ Precompiler

Use

You want to install the following installation packages:

- *Environment*
(sapdb-ind-7.3.00<build_no.>-1.i386.rpm)
- *C/C++ Precompiler for Developers*
(sapdb-ifpc-7.3.00<build_no.>-1.i386.rpm)



Always use the highest available build number to install the most up-to-date version of the C/C++ Precompiler.

Procedure

To install the packages, log on with the user `root`.

Start the installation process with the following command:

```
rpm -i <package-name>
```

Add the path `/opt/sapdb/interfaces/precompiler/bin` to the environment variable `PATH`.



If you have more than one precompiler installed on one computer, you must register the version you want to use. Use the tool [IRCONF \[Page 13\]](#) to do this. ([Registering the Interface Programs' Runtime Library on the Client \[Page 13\]](#)).

Result

The installation package *Environment* has been installed in the directory `/opt/sapdb/indep_prog`.

The installation package *C/C++ Precompiler for Developers* has been installed in the directory `/opt/sapdb/interfaces/precompiler`.



Calling the C/C++ Precompiler

Use

The C/C++ precompiler only compiles files with the file extension `.cpc`. From the file `<file_name>.cpc`, it generates the file `<file_name>.c`, which the C Compiler or the C++ Compiler can then convert.

Prerequisites

The environment variable `PATH` contains the path `/opt/sapdb/interfaces/precompiler/bin`.

Procedure

Use the following command to call the C/C++ precompiler:

```
cpc [-d <servername> -u <userid>,<password>] <file_name>
```

<code><server node></code>	The name of the database instance
<code><userid></code>	User name
<code><password></code>	User password
<code><file_name></code>	The name of the file to be compiled without the file extension



```
cpc -d DB -u Armstrong,Secret testfile
```

Options

The following options, among others, are available to you when calling the C/C++ precompiler:

Specify the database instance [Page 7]	-d
User ID [Page 7]	-u
Override connection to the database instance [Page 8]	-Hnocheck
Override the C/C++ Compiler [Page 8]	-c

For more information about the available options **see:** *C/C++ Precompiler: SAP DB 7.3*.



Specifying the Database Instance and the User ID: -d and -u

Use

Specifying the database instance and user ID when [calling the C/C++-Precompiler \[Page 6\]](#) is optional. There are three ways in which you can define the database instance and user:

1. You can specify the name of the database instance, the user name, and the user password in the command to call the C/C++ precompiler. The C/C++ precompiler will then use these values.



```
cpc -d DB -u Armstrong,Secret testfile
```

2. You can define a database instance and user in the file `<file_name>.cpc` in the SET_SERVERDB command and/or CONNECT statement (**see:** *C/C++ Precompiler: SAP DB 7.3*). As long as you do not add any other values when you call the C/C++ precompiler, the C/C++ precompiler will then use the values from the command or statement.
3. You can specify a database instance and a user using the tool XUSER. The C/C++ precompiler will use the values you enter with XUSER in that particular session (see [Using XUSER \[Extern\]](#)) as long as no other entries are made when the C/C++ precompiler is called.

Note that if you specify a database instance and a user in the command when calling the precompiler, this will override any values you entered with the tool XUSER or any entries you made in the file `<file_name>.cpc`.

If you have not defined a user and database instance in one of the three ways described above, the system will terminate the compiling operation and display an error message. One way of avoiding this would be to use the precompiler option [-Hnocheck \[Page 8\]](#).

Override Connection to the Database Instance: -Hnocheck

Use

If you have not entered the database instance and user definition [when calling the C/C++-Precompiler, \[Page 6\]](#) and want to avoid the system terminating the compilation process with an error message, use the option `-hnocheck`. If you use this option, the system will not attempt to connect to a database instance so it can compile without a database running.



```
cpc -hnocheck testfile
```

Override the C/C++ Compiler: -c

Use

Once it has executed the command, the precompiler calls the C/C++ Compiler and generates the object file `<file_name>.o`.

You can stop the C/C++ compiler being called by specifying the option `-c` when calling the C/C++ precompiler. When you use this option, the precompiler generates the file `<file_name_>.c`.



```
cpc -c testfile
```

Linking a C/C++ Program to the Runtime Library

Use

You are linking the file generated by [calling the C/C++ Precompiler \[Page 6\]](#) command to the required object files and precompiler libraries. This results in an executable program `<main>`.

Prerequisites

The environment variable `PATH` contains the path
`/opt/sapdb/interfaces/precompiler/bin`.

Procedure

Use the following command:

```
cpclnk <main> [<external_objects_or_libs>]
```

<main>	The name of the object file (without the file extension .o) containing the function <code>main</code>
<external_objects_or_libs>	Files with objects and precompiler libraries in any sequence. Specification of the file extension .o for objects and .a for libraries is optional. If you do not specify anything here, <main> will only be linked with the precompiler's runtime library.



Executing a C/C++ Program

Runtime Options

You can govern how a C/C++ program is executed using various runtime options. Set these options in the environment variable `SQLOPT`. For more information, [see: C/C++-Precompiler: SAP DB](#).



You can switch the precompiler trace on and off while the program is running. To do this, use the tool `IRTRACE` ([Using IRTRACE \[Page 12\]](#)).



Example for Compiling a File

The user has written a precompiler program (`test.cpc`).

Use the command `cpc -d DB -u Armstrong,Secret test`

to generate the file `test.o`. In this example, the database instance would be `DB`, the user `Armstrong`, and the password `Secret`.

Use the command `cpclnk test`

to link the file with the runtime library and to generate an executable program.

If you want a precompiler trace to be written when the `test` program is executed, you must accordingly set the environment variable `SQLOPT` to the value `-x`. In a bash shell, the command would be as follows:

```
export SQLOPT=-X
```

When the `test` program is executed, the precompiler trace is written to the file `test.pct`. This file is situated in the same directory as the `test` program.



To test the precompiler you have installed, you can use the SAP DB demo database at <http://www.sapdb.org/> → *Software with Sample Database* → *Linux* → *Create and Test a Sample Database* .

Once you have installed the relevant package (5kb), use `TST` as the database name, `TEST` as the user and `TEST` as the user password.

An example of a call you might use to compile the file `test.cpc`:

```
cpc -d TST -u TEST,TEST test
```



Installation Guide for Users

This installation guide is a brief introduction to using the C/C++ Precompiler, written for users.

If you want to use the C/C++ Precompiler exclusively to carry out the installation of precompiler applications, use the [Installation Guide for Developers \[Page 5\]](#).

For more detailed information on the features and options included in the C/C++ precompiler, please **see also**: *C/C++-Precompiler: SAP DB 7.3*.



Installing the C/C++ Precompiler Runtime Library

Use

You want to install the following installation packages:

- *Environment*
(sapdb-ind-7.3.00<build no.>-1.i386.rpm)
- *C/C++ Precompiler for Users*
(sapdb-precompiler-7.3.00<build no.>--1.i386.rpm)



Always use the highest available build number to install the most up-to-date version of the C/C++ Precompiler.

Procedure

To install the packages, log on with the user `root`.

Start the installation process with the following command:

```
rpm -i <package-name>
```

Add the path `/opt/sapdb/interfaces/precompiler/bin` to the environment variable `PATH`.



If you have more than one precompiler installed on one computer, you must register the version you want to use. Use the tool [IRCONF \[Page 13\]](#) to do this. ([Registering the Interface Programs' Runtime Library on the Client \[Page 13\]](#)).

Result

The installation package *Environment* has been installed in the directory `/opt/sapdb/indep_prog`.

The installation package *C/C++ Precompiler for Users* has been installed in the directory `/opt/sapdb/interfaces/precompiler`.



Example: Executing a Precompiler Application

You want to execute a precompiler application with the name `test`.

If you want a precompiler trace to be written when the `test` program is executed, you must accordingly set the environment variable `SQLOPT` to the value `-x`. In a bash shell, the command would be as follows:

```
export SQLOPT=-X
```

When the `test` program is executed, the precompiler trace is written to the file `test.pct`. This file is situated in the same directory as the `test` program.



To test the precompiler you have installed, you can use the SAP DB demo database at <http://www.sapdb.org/> → *Software with Sample Database* → *Linux* → *Create and Test a Sample Database*.

Once you have installed the relevant package (5kb), use `TST` as the database name, `TEST` as the user and `TEST` as the user password.

You could, for example, execute the following command in a bash shell:

```
export SQLOPT="-d TST -u TEST,TEST"
```

The result of this command is that a precompiler application uses the database instance `TST` and the user `TEST` with the password `TEST` during runtime.



Deinstallation

Use the following command to deinstall the C/C++ precompiler software:

```
rpm -e <package_name>
```



Using IRTRACE

IRTRACE is a tool for changing the settings of a precompiler trace and for displaying the current settings of a precompiler application.

All commands are entered at operating system level. IRTRACE and the precompiler application communicate via a shared memory segment.

If you change the trace settings of a precompiler application using IRTRACE, the system makes an entry in the shared memory segment. The precompiler application checks the entries in the shared memory at regular intervals and adjusts its trace settings accordingly. A precompiler application is assigned to the relevant entry in the shared memory using the application's process ID. The entry in the shared memory remains the same and can be called up as long as the relevant process is active.

When a shared memory segment is used, the system creates a synchronization file (`irtrace.shm`), which is used by the IRTRACE processes and the application processes to access the shared memory. Regardless of the release, the system then stores this file in the directory `/opt/sapdb/indep_prog/wrk`. To enable this, the system registers the path `/opt/sapdb/indep_prog` when it is installed and creates the sub-directory `wrk`. In addition, the system also assigns read/write authorizations to the IRTRACE tool and the precompiler application.



Displaying the Current Trace Settings

- Displaying the current trace settings of a precompiler application:

```
irtrace -p <process_id>
```

- Displaying an overview of the trace settings of all precompiler applications that have been changed using IRTRACE:

```
irtrace -p all
```



Modifying the Trace Settings

- To switch the trace on or off or to change the trace settings for a particular process:

```
irtrace -p <process_id> -t <trace_setting>
```

- To turn the trace on or off for **all** precompiler applications on the local server:

```
irtrace -p all -t <trace_setting>
```

Possible values for <trace_setting>:

<long>	Switch on long trace
<short>	Switch on short trace
<off>	Switch off trace



Using XUSER

You use the [XUSER tool \[Extern\]](#) to predefine user attributes to set up a [database session \[Extern\]](#) and save these to a user key (USER_KEY). You can access the required [XUSER data \[Extern\]](#) when you call the [Database Manager CLI \[Extern\]](#) or the C/C++ precompiler, or when you start the SAP systems by entering the relevant user key.



Using IRCONF

IRCONF is a tool for registering and displaying the runtime library versions that are installed on your system. All commands are entered at operating system level.



Registering the Interface Programs' Runtime

Library on the Client

Use

To enable the database server to be linked to the client and for them both to communicate, certain features must be installed, for example, the runtime library of the interface programs. This client software component is often updated. To ensure that you are able to continue to use and manage older applications, the different versions of the interface programs' runtime library must be installed in different directories.

You can manage all the different versions of the interface programs' runtime library using the tool IRCONF.

Prerequisites

The software for the database server and the client is installed.



IRCONF: Options

When using the [IRCONF \[Page 13\]](#) tool [Registering the Interface Programs' Runtime Library on the Client \[Page 13\]](#), the following options are available to you:

Specify installation path [Page 14]	-p <path>
Display registered versions of the runtime library [Page 15]	-s
Delete runtime library registration [Page 15]	-r
Override driver name [Page 16]	-d <CPC_driver>
Register runtime library version [Page 16]	-i
Override version check [Page 17]	-v <version>
Specify additional key [Page 18]	-k <key>
Display available options and help [Page 18]	-h



Specify Installation Path: -p

Use

You can define the installation path with the option `-p`. You only need to enter the main directory here. After the installation, the interface programs' runtime library is stored in a subdirectory of the specified directory.

- Under Windows NT: `.\pgm`
- Under UNIX (32 Bit): `./lib`

On 64-bit UNIX operating systems, the runtime library of interface programs is stored in the sub-directory `./lib/lib64`.

You can combine the option `-p` with the options `-i` ([Register runtime library version \[Page 16\]](#)) and/or `-v` ([Override version check \[Page 17\]](#)).



Display registered versions of the runtime library:

-s

Use

You can display the registered versions of the interface programs' runtime library on the client using the option `-s`.

Syntax

```
irconf -s
```

Result

The system displays the paths and versions of the interface programs' runtime library installed on the client.



```
/opt/sapdb/interfaces/precompiler/runtime/7.2 -> 7.2.04.00  
/opt/sapdb/interfaces/precompiler/runtime/7.3 -> 7.3.00.00
```



Delete runtime library registration: -r

Use

You delete the registration of an interface programs' runtime library on the client using the option `-r`. Specify the directory in which the client is installed.

Syntax

```
irconf -r -p <path>
```



```
irconf -r -p /opt/sapdb/interfaces/precompiler/runtime/7.2
```

Result

The system generates a message after successful deletion, specifying the path and the version of the installed interface programs' runtime library.



```
/opt/sapdb/interfaces/precompiler/runtime/7.2 -> 7.2.04.00
registration removed
```

Override Driver Name: -d

Use

The driver name helps you obtain information on the interface programs' runtime library version. The default value is `libpccr`.



The option `-d <CPC_Driver>` is used for support purposes only.

Syntax

```
irconf -i -d <CPC_driver>
```



```
irconf -i d <libpccr.old>
```

Register Runtime Library Version: -i

Use

You register a version of the interface programs' runtime library on the client using the option `-i`. Enter the directory in which the version of the library that you want to register is stored using the option `-p <path>`.

During the registration process, IRCONF identifies which version is installed in the directory (`libpccr.dll` on Windows NT or `libpccr.so` on UNIX) and checks it and registers it accordingly. In certain cases, you may need to skip the version check ([Override version check: -v \[Page 17\]](#)).

Syntax

```
irconf -i -p <path>
```



```
irconf -i -p /opt/sapdb/interfaces/precompiler/runtime/7.2
```

Result

The system generates a message after successful registration, specifying the path and the version of the installed interface programs' runtime library.



```
/opt/sapdb/interfaces/precompiler/runtime/7.2 -> 7.2.4.0
registered
```



Override Version Check: -v

Use

[IRCONF \[Page 13\]](#) tries to identify which version of the interface programs' runtime library is installed. In certain cases, you may override this version check using the option `-v <version>`. You can use this option when registering a version of the runtime library in order to save the relevant version under a version string that you have specified yourself.



You may want to register a 64-bit version of the runtime library with a 32-bit version of IRCONF. However, given that the 32-bit version of IRCONF is unable to identify a 64-bit version of the software, you must override the version check.



In certain support circumstances, it is necessary to register a version of the runtime library with a different name. You must therefore override the version check.

Syntax

```
irconf -i -v <version> -p <path>
```

The version specified in `<version>` comprises up to four numerical values, all separated by periods (##.##.##.##, for example, 7.3.00.00).



```
irconf -i -v 7.2.04.00 -p /usr/sapdb-if/runtime/7.2
```

Result

The system displays which version is installed in the specified path.



```
/usr/sapdb-if/runtime/7.2 -> 7.2.04.00
```



Specify Additional Key: -k

Use

When registering runtime libraries with IRCONF, it will not allow two identical versions of the same runtime library (in two different directories) to be identified on one computer. This may occur if you are running a test and production system on the same computer but with binaries from different directories.

If this is the case, use the option `-k <key>` to identify and locate the installations. Set the environment variable `SAPDBINSTKEY` to the value set in `<key>`, so that the applications use the version registered with this key.

Syntax

```
irconf -i -p <path> -k <key>
```



```
irconf -i -p /opt/sapdb/interfaces/precompiler/runtime/7.2 -
k db_instance_name
```

Result

The system displays the path under which the version specified with `<key>` was registered.



```
/opt/sapdb/interfaces/precompiler/runtime/7.2 -> 7.2.04.00
db_instance_name
```



Display available options and help: -h

Use

You can display a summary of the options that you can use with [IRCONF \[Page 13\]](#).

Syntax

```
irconf -h
```

Result

The system displays the following:

```
Usage : irconf -i[-v | -d | -p | -k] | -r -p | -s | -h
```

Options:

```
-i Registers a version
```

```
-r Unregisters a version
-s Displays all installed versions
-v <version> Overrides version number identified by the system
-d <CPC_driver> Default is 'libpcr'
-p <path> Installation path
-k <key> Installation name
-h Displays help
```



IRCONF Error Messages

IRCONF may display the following error messages:

Invalid option

You used an invalid command when calling IRCONF.

Invalid combination of options

This combination of options is not permitted.

Missing CPC Driver name

You used the option `-d` without specifying the relevant value.

Missing path name

You used the option `-p` without specifying the relevant value.

Missing installation name

You used the option `-k` without specifying the relevant value.

Missing version number

You used the option `-k` without specifying the relevant value.

Cannot get version info of CPC Driver

IRCONF was unable to identify which version of the runtime library of the interface programs is installed on the client.

The specified driver does not exist

The driver you specified either does not exist or could not be loaded.

Cannot find any installed version of CPC Driver

No runtime library installations are registered.

The specified installation does not exist

The installation you specified does not exist.