

Frequently Asked Questions of PROGRES Users

Prof. Dr. Andy Schürr
Real-Time Systems Lab
TU Darmstadt
Merckstr. 25
D-64283 Darmstadt, Germany

Ulrike Ranger
Lehrstuhl Informatik III
RWTH Aachen (University of Technology)
Ahornstr. 55
D-52074 Aachen, Germany

Contact Address: progres@i3.informatik.rwth-aachen.de

Version/Date: PROGRES Version 11.2, 09/03/04

Abstract: This document contains a list of frequently asked questions (arising problems) of PROGRES users around the world. We - the PROGRES development team - hope that the compiled answers for these questions are helpful and solve some of your problems. Please don't hesitate to contact us if you have any further questions and send an email to the contact address listed above. Maybe you will find your question and its answer in the next version of the FAQ.

The FAQ document is divided into three parts. Chapter 1 deals with all questions concerning the language PROGRES itself, Chapter 2 answers questions concerning the usage of its programming environment, whereas Chapter 3 lists those questions which arise when prototypes are compiled or used.

Please note that the identifier *\$PROGRESROOT* refers in the following to a central directory at your site, which contains the complete system installation. The identifier *\$DATAROOT* refers to a local directory of your own account, which contains all user-specific files of the system (written specifications, generated prototypes, window layout parameters, ...).

Download:

- This document as a gzipped Postscript file [FAQ.ps.gz](#).
- This document as an Acrobat reader file [FAQ.pdf](#).

1. The Programming Language FAQ

1.1 General Questions

1.1.1 Getting started

Please read the document *\$PROGRESROOT/man/ProgresForBeginners.ps*. It is a short introduction to the language PROGRES which uses the development of a software design tool as the running example. Please note that it makes no attempt to cover the whole language nor does it make any attempts to present a formal definition of language constructs. Furthermore, the document still uses the old graphical representation of graph schemata, instead of the slightly different UML-like version of new PROGRES release.

A new up-to-date language report, which uses the development of a library information system as the running example will be available soon.

1.1.2 Problems with specific language constructs

The whole language PROGRES is documented on the WWW-page [ProgresSyntax.html](#)

It defines the (text) syntax of all language constructs and explains its graphical representation as well as its semantics informally.

2. The Programming Environment FAQ

2.1 General Questions

2.1.1 Getting started

Please read the document `$PROGRESROOT/man/GuidedTour.ps` . It explains in the form of a guided tour how to install and use the system PROGRES.

2.1.2 Getting rid of input window title bars

We strongly recommend that you change the configuration files of your window manager such that PROGRES input windows have no title bars and borders. The document `$PROGRESROOT/man/GuidedTour.ps` explains for a number of window manager how to do that. Otherwise, text input windows never open precisely above the construct you are going to modify.

2.1.3 Printing specifications

Select a portion of your PROGRES document (specification) and execute `MISC:WriteFile:MIFFormat` . This command generates a printable document in Framemaker office document interchange format MIF. Framemaker is a text processing system sold by Adobe which may be used to edit the generated document and to translate into pdf- or ps-format. For further information about Framemaker please visit

<http://www.adobe.com/prodindex/framemaker/prodinfo.html>

Please note that you don't need the full Framemaker version for printing mif-documents. You may also use the FrameViewer tool which allows you to inspect and print mif-documents.

Furthermore, we offer a PROGRES specific mif- to latex-converter (with xfig figures). For further information about this tool please visit [this page](#).

2.1.4 Exchanging documents between different platforms

It is not possible to use the same project database on different platforms (Sun workstations, Linux PC) due to incompatibilities of the used binary formats. Whenever a PROGRES document written on one platform is needed on a different platform it has to be exchanged as a plain text file:

- *Export of text file* : open document, select the to be exported part (usually the whole document) and activate `MISC:WriteFile:WithLayout` .
- *Import of text file* : select command `EDIT:Import` of the project window to import a full document into your project database or select command `MISC:ReadFile` of the document window to insert a part of one document into another document of your project database.

2.2 Sun Workstation Installation Problems

2.2.1 Could not start ganimed on Sun workstations

The PROGRES system's underlying DBMS GRAS makes heavy use of interprocess communication. It consists of a single control process, called *ganimed* , and at least one database server processes. It may happen that the script

```
$PROGRESROOT/bin/StartGanimed or $PROGRESROOT/bin/Progres
```

fails with an error message of the form "cannot start ganimed"

On Sun workstations please check whether

rsh localhost "echo something" or rsh \$HOST "echo something"

works. Depending on your installation of the Solaris operating system it might be helpful to add an entry of the form *localhost \$USER* to the file *.rhosts* in your home directory.

2.3 Linux PC Installation Problems

2.3.1 RedHat 5.1/5.2 installation caveats

PROGRES should run without problems. Be sure to have the packages *egcs-1.0.*.rpm* , *egcs-c++-1.0.*.rpm* and *libstdc++-devel-2.8.*.rpm* as well as *portmap-4.*.i386.rpm* installed.

2.3.2 SuSE 6.0 installation caveats

SuSE ships with *egcs-1.1.1* as default c++ compiler. Be sure to have *egcs.rpm* , *gpp.rpm* , *libgpp.rpm* and *nkitb.rpm* (contains portmapper) installed. The *egcs-1.1.1* compiler automatically links C++ programs against the version 2.9 of the *stdc++* library and its own *libgcc* . In order to run the PROGRES environment (of which parts are compiled with *egcs-1.0.3*), you have to make version 2.8 of the *stdc++* library available, a symbolic link to the 2.9 version might do as well. Due to an incompatibility of the *libgcc* of *egcs-1.0* and *egcs-1.1* , the linking of a prototype requires an additional symbol *__eh_pc* , defined as *void** .

2.3.3 Debian 2.0 installation caveats

Debian ships with *egcs-1.0.x* as default C++ compiler and version 2.8 of *libstdc++* . There is a problem with the *termcap* library which is only provided as a libc5-based library. In order to run progres, you have to recompile this library or get a binary from another distribution (we will try to remove the dependency on termcap in a future version).

Apart from this problem, Progres should run fine with the Debian distribution.

2.3.4 Could not start the browser

In the newer Linux distributions the interpreter browser doesn't start because of a missing *stdc++* library. In this cases you have to copy the library *libg++-libc6.2-2.so* into *\$PROGRESROOT/lib/LINUXLIBC6*. Perhaps you need *libstdc++.so.2.8*, too.

2.3.5 Could not start ganimed on Linux PC

The PROGRES system's underlying DBMS GRAS makes heavy use of interprocess communication. It consists of a single control process, called *ganimed* , and at least one database server processes. It may happen that the script

```
$PROGRESROOT/bin/StartGanimed or $PROGRESROOT/bin/Progres
```

fails with an error message of the form

```
"cannot start ganimed"
```

In the case of Linux please check whether the portmapper is installed and running (cf. [problem 2.3.1](#), and [problem 2.3.2](#)).

3. Generating Prototypes FAQ

This chapter lists all those questions and their answers dealing with the development of PROGRES prototypes. These prototypes are generated C programs which have the same functionality as a given graph transformation specification plus an interactive user interface. Any specified graph transformation defines a menu entry of the generated prototype plus an appropriate parameter input mask. A huge number of property sheets may be used to customize the representation (layout) of the manipulated class of graphs.

3.1 General Questions

3.1.1 Getting started

Please read the document `$PROGRESROOT/man/GuidedPrototyping.ps` . It explains in the form of a guided tour how to generate, customize, and use a prototype. Two examples of generated prototypes are delivered separately from the PROGRES system installation (source code plus binaries for Solaris). Please inspect the WWW page

www-i3.informatik.rwth-aachen.de/research/projects/progres/

for further details how to download these examples.

3.1.2 Problems with multi-user access

The generated prototype is very, very slow or crashes even when you several prototype instances access the same graph database at the same time. Probably the prototype has been compiled and linked using the makefile option

`MULTIUSER = FALSE` .

Change the makefile appropriately, touch the dummy file `DateOfGeneration` and start `make` again.

3.2 Problems on Sun Workstations (Solaris 2.x)

3.2.1 Problems with make

Check whether you are using GNU make. The standard make delivered with the Solaris operating system is not able to process the delivered Makefile.

3.2.2 Problems with compiling prototypes

Check whether you have installed the GNU compiler `gcc` version 2.8.1 together with the library `libstdc++` .

3.2.3 Missing include files or libraries

The compilation of the generated prototype stops with the error message

`iostream.h, stream.h, _G_config.h: No such file or directory`

and/or with the error message

`cstdlib, cstring, cstdio, cerrno: No such file or director` .

In this case gcc does not find the include files for the library `libstdc++`. Please install this library as follows:

1. download the files `libstdc++-2.8.1.1.tar.gz` (and `gcc-2.8.1.tar.gz`) from the GNU archive or any of its mirrors around the world as e.g. via anonymous ftp from `ftp.informatik.rwth-aachen.de/pub/gnu/libg++-2.8.1.1a`
2. create a new directory: `mkdir builddir; cd builddir`
3. extract the library with `gunzip < /where/you/store/downloadfiles/libstdc++-2.8.1.1.tar.gz | tar xf -`
4. go to the directory `cd libstdc++-2.8.1.1` and `g unzip < /where/you/store/downloadfiles/gcc-2.8.1.tar.gz | tar xf -`
5. finally execute `mv gcc-2.8.1 gcc; mkdir objdir; cd objdir`
`./configure --enable-shared; make bootstrap; make install`

The very last step installs the builded library and compiler, i.e. it moves it to `/usr/local` and subdirectories

of */usr/local* . Therefore you need read, write and execute permissions for those directories. We recommend to perform the last step as *root* .

3.3 Problems with Linux

Until now we are not aware of any specific problems which are not already addressed by [problem 2.3.](#) and [problem 3.2.](#) above.