



RELEASE NOTES

PRLGCC 5.0

January, 2017



IMPORTANT NOTICE

DISCLAIMER OF WARRANTY

The staff of Programming Research Ltd. have taken due care in preparing this document which is believed to be accurate at the time of printing. However, no liability can be accepted for errors or omissions nor should this document be considered as an expressed or implied warranty that the products described perform as specified within.

COPYRIGHT NOTICE

This document is copyrighted and may not, in whole or in part, be copied, reproduced, disclosed, transferred, translated, or reduced to any form, including electronic medium or machine-readable form, or transmitted by any means, electronic or otherwise, unless Programming Research Ltd consents in writing in advance. Copyright ©2015 *Programming Research Ltd.*

TRADEMARKS

PRQA, the PRQA logo , QA·C, QA·C++ and High Integrity C++ (HIC++) are trademarks of *Programming Research Ltd.*

"MISRA", "MISRA C" and "MISRA C++" are registered trademarks of MIRA Limited, held on behalf of the MISRA Consortium.

Yices is a registered trademark of SRI International.

Windows is a registered trademark of Microsoft Corporation.

CONTACTING PROGRAMMING RESEARCH LTD

For technical support, contact your nearest Programming Research Ltd authorized distributor or you can contact Programming Research's head office:

by telephone on +44 (0) 1932 888 080

by fax on +44 (0) 1932 888 081

or by webpage: www.programmingresearch.com/services/contact-support/



Contents

1	Introduction	1
2	PRLGCC Changes Overview	2
2.1	Functional Changes in PRLGCC 5.0	2
2.1.1	Improved support for GCC variants	2
2.1.2	Change to Version Detecting Algorithm	2
2.1.3	Additional Support	3
3	PRLGCC Ticket Summary	4
3.1	Ticket Summary for PRLGCC 5.0	4





List of Figures

List of Tables

3.1	Ticket Summary for PRLGCC 5.0	4
-----	---	---





1 Introduction

Version 5.0 release of PRLGCC is a full release.

This document provides information on the major feature additions as well as all feature fixes made in this release.



2 PRLGCC Changes Overview

2.1 Functional Changes in PRLGCC 5.0

2.1.1 Improved support for GCC variants

The GNU collection is often used as a backend code generate for 3rd party clients. These front ends will often provide additional defines and flags to the compiler to ensure that the code compiles correctly.

However, PRLGCC depends on specific command line options being supported by the tool and producing the output necessary to create a working personality or CCT. This version of PRLGCC changes how the compiler is queried with the core options being specified as part of the `-Wp` option, ie. passed to the preprocessor.

Three passes are then performed:

1. The command line as specified to PRLGCC is invoked and the includes and defines gathered.
2. Using the gathered includes and defines, the command line is then scanned to find and remove explicit `-I`, `-D` and `-include` options.
3. A final pass calls the top level tool with `-v` in an attempt to determine the Target compiler.

The resulting configuration will include only the options implicitly defined by the compiler, excluding those explicitly specified on the command line.

2.1.2 Change to Version Detecting Algorithm

Official releases of GNU GCC define a macro `__GLIBCXX__` with an ISO datestamp that, in most cases can be mapped to the version of library being used.

The list of versions and official releases can be found here:

<https://gcc.gnu.org/onlinedocs/libstdc++/manual/abi.html#abi.versioning>

The structure of the stubbed headers has been modified to make use of the `__GLIBCXX__` macro.

The main benefit of this approach being that the path through the headers is independent of the version reported by the compiler. This will reduce the effort required to work with 3rd party compilers that use the standard library headers provided by GNU GCC.

One consequence of this change is that by default, future C++ standard library versions



will not be supported until an entry is added for them in the following location:

```
.../config/cct/DATA/GNU_GCC-g++/Stub/gcc/bits/qacpp_glibcxx.h
```

Note: Some distributions ship a modified version of the C++ standard library with a changed datestamp. These can be handled by adding an entry for the version, see the `qacpp_glibcxx.h` header file for more information.

2.1.3 Additional Support

In addition to the listed tickets support has been added for the following versions:

- 4.8.5
- 4.9.4

3 PRLGCC Ticket Summary

3.1 Ticket Summary for PRLGCC 5.0

The following table summarizes the tickets that were closed in PRLGCC 5.0.

Tickets are categorized into 3 types:

- E** – Enhancement to an existing feature.
- F** – A fix of a bug or problem feature.
- N** – New functionality has been introduced.

Table 3.1: Ticket Summary for PRLGCC 5.0

Ticket	Type	Description
21938	E	The <code>__UNSIGNED_CHAR__</code> macro, if defined will be used to set the value of the <code>-u</code> macro in QA·C.
22549	E	Replace <code>__int128</code> with <code>long long</code> .
22871	F	Improved support for MinGW.
24471	E	Use <code>-Wp</code> to pass options through to the preprocessor allowing PRLGCC to be used with a wider set of GNU GCC variant compilers.
24672	E	Return constant expression from <code>__builtin?</code> functions to avoid errors parsing 'ratio'
24673	E	Work around incorrect lookup of <code>swap</code> in <code>priority_queue</code> in <code>noexcept</code> used in declaration