

gcc.print Plugin

Table of contents

1 gcc.print plugin.....	2
2 Parameters.....	2
3 Export gcc.print.current-function.name.....	2
4 Export gcc.print.function.xml.....	2
5 Export gcc.print.function.xml.brief.....	2
6 Export gcc.print.loop.xml.....	2
7 Export gcc.print.loop.xml.brief.....	3
8 Export gcc.print.basic-block.xml.....	3
9 Export gcc.print.basic-block.xml.brief.....	3
10 Export gcc.print.tree.xml.....	3
11 Export gcc.print.tree.xml.brief.....	3
12 Export gcc.print.rtx.xml.....	3
13 Export gcc.print.rtx.xml.brief.....	4
14 Export gcc.print.rtl.....	4

1. gcc.print plugin

This plugin provides print functions, suitable for the message plugin.

2. Parameters

Parameter	Value	Notes
Id	gcc.print	
Version processing instruction	<?gcc version="4.2.0"?>	
Version	0.1.0	
Location	gcc.print-0.1.0	
Lazy	true	
Author	Hugh Leather	

3. Export gcc.print.current-function.name

This export is a function of signature: `void (*)(FILE* file)`

It will print the name of the current function to the given file (which must be open)

4. Export gcc.print.function.xml

This export is a function of signature: `void (*)(FILE* file, void* value)`

It assumes that value points to `struct function*` and will print it, in detail, as xml to the given file (which must be open)

5. Export gcc.print.function.xml.brief

This export is a function of signature: `void (*)(FILE* file, void* value)`

It assumes that value points to `struct function*` and will print it, in briefly, as xml to the given file (which must be open)

6. Export gcc.print.loop.xml

This export is a function of signature: `void (*)(FILE* file, void* value)`

It assumes that `value` points to `struct loop*` and will print it, in detail, as xml to the given file (which must be open)

7. Export `gcc.print.loop.xml.brief`

This export is a function of signature: `void (*)(FILE* file, void* value)`

It assumes that `value` points to `struct loop*` and will print it, in briefly, as xml to the given file (which must be open)

8. Export `gcc.print.basic-block.xml`

This export is a function of signature: `void (*)(FILE* file, void* value)`

It assumes that `value` points to `basic_block` and will print it, in detail, as xml to the given file (which must be open)

9. Export `gcc.print.basic-block.xml.brief`

This export is a function of signature: `void (*)(FILE* file, void* value)`

It assumes that `value` points to `basic_block` and will print it, in briefly, as xml to the given file (which must be open)

10. Export `gcc.print.tree.xml`

This export is a function of signature: `void (*)(FILE* file, void* value)`

It assumes that `value` points to `tree` and will print it, in detail, as xml to the given file (which must be open)

11. Export `gcc.print.tree.xml.brief`

This export is a function of signature: `void (*)(FILE* file, void* value)`

It assumes that `value` points to `tree` and will print it, in briefly, as xml to the given file (which must be open)

12. Export `gcc.print.rtx.xml`

This export is a function of signature: `void (*)(FILE* file, void* value)`

It assumes that `value` points to `rtx` and will print it, in detail, as xml to the given file (which

must be open)

13. Export gcc.print.rtx.xml.brief

This export is a function of signature: `void (*)(FILE* file, void* value)`

It assumes that value points to `rtx` and will print it, in briefly, as xml to the given file (which must be open)

14. Export gcc.print.rtl

This export is a function of signature: `void (*)(FILE* file, void* value)`

It assumes that value points to `rtl` and will print it, in detail, as xml to the given file (which must be open). The entire instruction chain will be printed in detail