

The Ciao Internals

The Ciao Compiler, Engine, and Runtime libraries

REFERENCE MANUAL

The Ciao Documentation Series

<https://ciao-lang.org/>

Generated/Printed on: 22 July 2023

Table of Contents

1	Summary	1
2	Introduction.....	3
3	Dynamic loading and compilation.....	5
3.1	Usage and interface.....	5
3.2	Documentation on exports	5
	make_po/1 (pred)	5
	make_wam/1 (pred)	5
	ensure_loaded/1 (pred).....	5
	ensure_loaded/2 (pred).....	5
	use_module/1 (pred).....	5
	use_module/2 (pred).....	5
	use_module/3 (pred).....	5
	unload/1 (pred).....	6
	set_debug_mode/1 (pred).....	6
	set_nodebug_mode/1 (pred)	6
	set_debug_module/1 (pred).....	6
	set_nodebug_module/1 (pred).....	6
	set_debug_module_source/1 (pred)	6
	mode_of_module/2 (pred)	6
	module_of/2 (pred)	6
3.3	Documentation on imports	6
4	exemaker (library)	7
4.1	Usage and interface.....	7
4.2	Documentation on exports	7
	make_exec/2 (pred).....	7
	force_lazy/1 (pred)	7
	undo_force_lazy/1 (pred)	7
	dynamic_search_path/1 (pred)	7
	needs_interface/2 (pred).....	7
	create_interfaces/0 (pred)	7
4.3	Documentation on multifiles.....	8
	define_flag/3 (pred).....	8
4.4	Documentation on imports	8
5	Compiler frontend	9
5.1	Usage and interface.....	9
5.2	Documentation on exports	9
	compiler_version/1 (pred)	9
	compiler_name/1 (pred).....	9
	opt_suffix/2 (pred)	10
	defines_module/2 (pred)	10
	def_multifile/4 (pred)	10
	decl/2 (pred).....	10
	exports/5 (pred)	10
	uses/2 (pred)	10

adds/2 (pred)	10
uses_file/2 (pred)	10
imports_pred/7 (pred)	11
imports_all/2 (pred)	11
includes/2 (pred)	11
loads/2 (pred)	11
clause_of/7 (pred)	11
package/2 (pred)	11
imports_nocheck/4 (pred)	12
defines_pred/3 (pred)	12
meta_pred/4 (pred)	12
dyn_decl/4 (pred)	12
defines/5 (pred)	12
comp_defines/1 (pred)	12
restore_defines/5 (pred)	12
restore_imports/5 (pred)	12
restore_multifile/4 (pred)	12
imports/5 (pred)	13
meta_args/2 (pred)	13
(multifile)/3 (pred)	13
defines/3 (pred)	13
(discontiguous)/3 (pred)	13
add_module_check/1 (pred)	13
ensure_imported/4 (pred)	13
define_ops/0 (pred)	13
processed/2 (pred)	13
process_files_from/7 (pred)	14
process_file/7 (pred)	14
old_file_extension/2 (pred)	14
cleanup_c_itf_data/0 (pred)	14
cleanup_itf_cache/0 (pred)	14
base_name/2 (pred)	14
file_data/3 (pred)	14
compute_base_name/4 (pred)	14
module_from_base/2 (pred)	14
exports_pred/3 (pred)	14
module_error/0 (pred)	14
module_error/1 (pred)	15
expand_module_decl/5 (pred)	15
read_sentence/3 (pred)	15
default_package/1 (pred)	15
handle_exc/1 (pred)	15
false/1 (pred)	15
make_object1/2 (pred)	15
make_po_file/1 (pred)	15
activate_translation/3 (pred)	15
end_goal_trans/1 (pred)	15
module_expansion/9 (pred)	15
c_itf_internal_pred/2 (pred)	16
c_itf_internal_pred_decl/1 (pred)	16
incore_mode_of/2 (pred)	16
pred_module/2 (pred)	16
location_t/1 (regtype)	16
location/3 (pred)	16
location/1 (pred)	16
set_ciaopp_expansion/1 (pred)	16

	(multifile)/1 (pred)	17
	module_loaded/4 (pred)	17
	static_module/1 (pred)	17
	static_base/1 (pred)	17
	use_mod/3 (pred)	17
	use_mod_user/2 (pred)	17
	abolish_module/1 (pred)	17
	unload_mod/1 (pred)	17
	interpret_file/1 (pred)	17
	interpret_module/1 (pred)	17
	interpret_srcdbg/1 (pred)	17
5.3	Documentation on multifiles	18
	define_flag/3 (pred)	18
	do_on_abolish/1 (pred)	18
5.4	Documentation on imports	18
6	pl2wam (library)	19
6.1	Usage and interface	19
6.2	Documentation on exports	19
	reset_counter/1 (pred)	19
	set_compiler_mode_out/2 (pred)	19
	set_compiler_mode/1 (pred)	19
	set_compiler_out/1 (pred)	19
	compile_clause/2 (pred)	19
	compile_clause_in_mode/3 (pred)	19
	proc_declaration/4 (pred)	19
	proc_declaration_in_mode/5 (pred)	19
	cleanup_compilation_data/0 (pred)	20
6.3	Documentation on imports	20
7	Emulator description language	21
7.1	Usage and interface	21
8	The Ciao engine	23
	References	25
	Library/Module Index	27
	Predicate Index	29
	Property Index	31
	Regular Type Index	33
	Declaration Index	35
	Concept Index	37
	Author Index	39
	Global Index	41

1 Summary

This (semi-automatically generated) manual is still incomplete. It is however being updated regularly as the machine-readable documentation of the different libraries is completed.

This is the internal manual for the Ciao compiler, engine, and runtime libraries. It documents modules not included in the reference manual.

Please consult the *alldocs* bundle for the Ciao reference manual.

2 Introduction

This (semi-automatically generated) manual is still incomplete. It is however being updated regularly as the machine-readable documentation of the different libraries is completed.

This is the internal manual for the Ciao compiler, engine, and runtime libraries. It documents modules not included in the reference manual.

Please consult the *alldocs* bundle for the Ciao reference manual.

3 Dynamic loading and compilation

Author(s): The Ciao Development Team.

This module provides a simplified interface to the Ciao compiler front-end (see `toplevel` for details).

3.1 Usage and interface

- **Library usage:**
`:- use_module(library(compiler)).`
- **Exports:**
 - *Predicates:*
`make_po/1, make_wam/1, ensure_loaded/1, ensure_loaded/2, use_module/1, use_module/2, use_module/3, unload/1, set_debug_mode/1, set_nodebug_mode/1, set_debug_module/1, set_nodebug_module/1, set_debug_module_source/1, mode_of_module/2, module_of/2.`

3.2 Documentation on exports

make_po/1:	PREDICATE
No further documentation available for this predicate.	
make_wam/1:	PREDICATE
No further documentation available for this predicate.	
ensure_loaded/1:	PREDICATE
No further documentation available for this predicate.	
ensure_loaded/2:	PREDICATE
No further documentation available for this predicate.	
use_module/1:	PREDICATE
No further documentation available for this predicate.	
use_module/2:	PREDICATE
No further documentation available for this predicate. <i>Meta-predicate</i> with arguments: <code>use_module(? , addmodule(?)).</code>	

use_module/3: No further documentation available for this predicate.	PREDICATE
unload/1: No further documentation available for this predicate.	PREDICATE
set_debug_mode/1: No further documentation available for this predicate.	PREDICATE
set_nodebug_mode/1: No further documentation available for this predicate.	PREDICATE
set_debug_module/1: No further documentation available for this predicate.	PREDICATE
set_nodebug_module/1: No further documentation available for this predicate.	PREDICATE
set_debug_module_source/1: No further documentation available for this predicate.	PREDICATE
mode_of_module/2: No further documentation available for this predicate.	PREDICATE
module_of/2: No further documentation available for this predicate.	PREDICATE

3.3 Documentation on imports

This module has the following direct dependencies:

- *Application modules:*
datafacts_rt, system, c_itf, exemaker.
- *Packages:*
prelude, initial, condcomp, assertions, assertions/assertions_basic, datafacts.

4 exemaker (library)

4.1 Usage and interface

- **Library usage:**
`:- use_module(library(compiler/exemaker)).`
- **Exports:**
 - *Predicates:*
`make_exec/2, force_lazy/1, undo_force_lazy/1, dynamic_search_path/1, needs_interface/2, create_interfaces/0.`
 - *Multifiles:*
`define_flag/3.`

4.2 Documentation on exports

make_exec/2:	PREDICATE
No further documentation available for this predicate.	
force_lazy/1:	PREDICATE
No further documentation available for this predicate.	
undo_force_lazy/1:	PREDICATE
No further documentation available for this predicate.	
dynamic_search_path/1:	PREDICATE
No further documentation available for this predicate. The predicate is of type <i>data</i> .	
needs_interface/2:	PREDICATE
No further documentation available for this predicate. The predicate is of type <i>data</i> .	
create_interfaces/0:	PREDICATE
No further documentation available for this predicate.	

4.3 Documentation on multifiles

define_flag/3:

PREDICATE

Usage: `define_flag(Flag,FlagValues,Default)`

- *The following properties hold upon exit:*

Flag is an atom.

(atm/1)

Define the valid flag values

(flag_values/1)

The predicate is *multifile*.

4.4 Documentation on imports

This module has the following direct dependencies:

- *Application modules:*

`datafacts_rt`, `aggregates`, `c_itf`, `file_buffer`, `stream_utils`, `system`, `engine_path`,
`pl2wam`, `lists`, `ctrlcclean`, `build_foreign_interface`.

- *Packages:*

`prelude`, `initial`, `condcomp`, `assertions`, `assertions/assertions_basic`, `define_flag`, `datafacts`.

5 Compiler frontend

Author(s): The Ciao Development Team.

This module implements the Ciao compiler front-end.

The main entry point for the front-end is the `process_files_from/7` predicate, which process (Ciao) source files and all their dependencies. Source file processing is incremental and dependency extraction automatic from source declarations. For each processed file, a user-provided back-end (as higher-order arguments) is invoked.

Based on incremental file processing, this module implements source-to-bytecode compilation (using `pl2wam` and `wamql` as back-end) and basic support for dynamic module loading (`use_mod/3`).

5.1 Usage and interface

- **Library usage:**

```
:- use_module(library(compiler/c_itf)).
```

- **Exports:**

- *Predicates:*

```
compiler_version/1, compiler_name/1, opt_suffix/2, defines_module/2, def_multifile/4, decl/2, exports/5, uses/2, adds/2, uses_file/2, imports_pred/7, imports_all/2, includes/2, loads/2, clause_of/7, package/2, imports_noccheck/4, defines_pred/3, meta_pred/4, dyn_decl/4, defines/5, comp_defines/1, restore_defines/5, restore_imports/5, restore_multifile/4, imports/5, meta_args/2, multifile/3, defines/3, discontinuous/3, add_module_check/1, ensure_imported/4, define_ops/0, processed/2, process_files_from/7, process_file/7, old_file_extension/2, cleanup_c_itf_data/0, cleanup_itf_cache/0, base_name/2, file_data/3, compute_base_name/4, module_from_base/2, exports_pred/3, module_error/0, module_error/1, expand_module_decl/5, read_sentence/3, default_package/1, handle_exc/1, make_object1/2, make_pos_file/1, activate_translation/3, end_goal_trans/1, module_expansion/9, c_itf_internal_pred/2, c_itf_internal_pred_decl/1, incore_mode_of/2, pred_module/2, location/3, location/1, set_ciaopp_expansion/1, multifile/1, module_loaded/4, static_module/1, static_base/1, use_mod/3, use_mod_user/2, abolish_module/1, unload_mod/1, interpret_file/1, interpret_module/1, interpret_srcdbg/1.
```

- *Regular Types:*

```
location_t/1.
```

- *Multifiles:*

```
define_flag/3, do_on_abolish/1.
```

5.2 Documentation on exports

compiler_version/1:

PREDICATE

No further documentation available for this predicate.

compiler_name/1: No further documentation available for this predicate.	PREDICATE
opt_suffix/2: No further documentation available for this predicate.	PREDICATE
defines_module/2: Usage: <code>defines_module(Base,Module)</code> The source <code>Base.pl</code> defines module <code>Module</code> . The predicate is of type <i>data</i> .	PREDICATE
def_multifile/4: Usage: <code>def_multifile(Base,F,A,DynType)</code> The source <code>Base.pl</code> defines multifile predicate <code>F/A</code> , defined as <code>DynType</code> (static, dynamic, data or concurrent). The predicate is of type <i>data</i> .	PREDICATE
decl/2: Usage: <code>decl(Base,Decl)</code> The source <code>Base.pl</code> contains the declaration <code>Decl</code> as an itf-exported <code>new_declaration</code> . The predicate is of type <i>data</i> .	PREDICATE
exports/5: Usage: <code>exports(Base,F,A,DefType,Meta)</code> The source <code>Base.pl</code> exports predicate <code>F/A</code> , defined as <code>DefType</code> (static, implicit, dynamic, data or concurrent) and with meta-predicate declaration <code>Meta</code> (which can be 0 if it has not).	PREDICATE
uses/2: Usage: <code>uses(Base,File)</code> The source <code>Base.pl</code> imports from file <code>File</code> . The predicate is of type <i>data</i> .	PREDICATE
adds/2: Usage: <code>adds(Base,File)</code> The source <code>Base.pl</code> does <code>ensure_loaded/1</code> of file <code>File</code> . The predicate is of type <i>data</i> .	PREDICATE

uses_file/2: PREDICATE**Usage:** `uses_file(Base,File)`

The source `Base.pl` uses file `File` explicitly through `use_module/1` or `use_module/2` or implicitly.

imports_pred/7: PREDICATE**Usage:** `imports_pred(Base,ImpFile,F,A,DefType,Meta,EndFile)`

The source `Base.pl` imports from file `ImpFile` predicate `F/A`. Predicate is defined as `DefType` and has meta_predicate declaration `Meta` (possibly 0). `EndFile` is `'.'` if the predicate resides in `ImpFile`, otherwise it is the file in which the predicate resides (due to reexportations). Stored in itf file for dependency check.

The predicate is of type *data*.

imports_all/2: PREDICATE**Usage:** `imports_all(Base,ImpFile)`

The source `Base.pl` imports all predicates of `ImpFile`.

The predicate is of type *data*.

includes/2: PREDICATE**Usage:** `includes(Base,File)`

The source `Base.pl` includes file `File`. Stored in itf file for dependency check.

The predicate is of type *data*.

loads/2: PREDICATE**Usage:** `loads(Base,File)`

The source `Base.pl` does `load_compilation_module` of file `File`. Stored in itf file for dependency check.

The predicate is of type *data*.

clause_of/7: PREDICATE**Usage:** `clause_of(Base,Head,Body,VarNames,Source,Line0,Line1)`

We have read from `Base.pl` (or included files) the clause `Head :- Body`, which has variable names `VarNames`, and is located in source `Source` (that changes if clauses are from an included file) between lines `Line0` and `Line1`. In the special case that `Head` is a number, `Body` is the body of a declaration.

The predicate is of type *data*.

package/2: PREDICATE

The predicate is of type *data*.

imports_nocheck/4:	PREDICATE
Usage: <code>imports_nocheck(Base,Module,F,A)</code> The source <code>Base.pl</code> imports predicate <code>F/A</code> from module <code>Module</code> using <code>import/2</code> . The predicate is of type <i>data</i> .	
defines_pred/3:	PREDICATE
The predicate is of type <i>data</i> .	
meta_pred/4:	PREDICATE
The predicate is of type <i>data</i> .	
dyn_decl/4:	PREDICATE
The predicate is of type <i>data</i> .	
defines/5:	PREDICATE
Usage: <code>defines(Base,F,A,DefType,Meta)</code> The source <code>Base.pl</code> defines predicate <code>F/A</code> , defined as <code>DefType</code> (static, implicit, dynamic, data or concurrent) and with meta-predicate declaration <code>Meta</code> (which can be 0 if it has not). Generated by calling <code>comp_defines/1</code> . The predicate is of type <i>data</i> .	
comp_defines/1:	PREDICATE
Usage: <code>comp_defines(Base)</code> Can be used in the <code>TreatP</code> phase of the compilation process to generate facts of <code>defines/5</code> for source <code>Base.pl</code>	
restore_defines/5:	PREDICATE
Usage: <code>restore_defines(Base,F,A,DefType,Meta)</code> Given the same arguments obtained from <code>defines/5</code> , it generates (assert in <code>c_itf</code> DB) all necessary data to offer itf functionality.	
restore_imports/5:	PREDICATE
Usage: <code>restore_imports(M,IM,F,A,EndMod)</code> Given the same arguments obtained from <code>imports/5</code> , it restores <code>c_itf</code> imports DB.	
restore_multifile/4:	PREDICATE
Usage: <code>restore_multifile(M,IM,F,Def)</code> Given the same arguments obtained from <code>multifile/4</code> , it restores <code>c_itf</code> multifile DB.	

imports/5:	PREDICATE
Usage: <code>imports(Mod,Mod2,F,A,EndMod)</code> Module <code>Mod</code> imports from module <code>Mod2</code> predicate <code>F/A</code> , which resides in module <code>EndMod</code> . <code>EndMod</code> can be different from <code>Mod2</code> due to reexportation. The predicate is of type <i>data</i> .	
meta_args/2:	PREDICATE
Usage: <code>meta_args(Mod,Meta)</code> Module <code>Mod</code> has meta-predicate declaration <code>Meta</code> . The predicate is of type <i>data</i> .	
multifile/3:	PREDICATE
No further documentation available for this predicate.	
defines/3:	PREDICATE
Usage: <code>defines(Mod,F,A)</code> Module <code>Mod</code> defines predicate <code>F/A</code> . The predicate is of type <i>data</i> .	
discontiguous/3:	PREDICATE
The predicate is of type <i>data</i> .	
add_module_check/1:	PREDICATE
Usage: <code>add_module_check(Pred)</code> Used by code expanders (loaded with <code>load_compilation_module/1</code> declarations) to provide additional checks to be executed before the current file is compiled. The predicate is called with the base name of the file processed as the first argument, and all solutions will be found. A fact <code>module_error/0</code> can be asserted if a condition which should stop the compilation is found. <i>Meta-predicate</i> with arguments: <code>add_module_check(pred(1))</code> .	
ensure_imported/4:	PREDICATE
No further documentation available for this predicate.	
define_ops/0:	PREDICATE
No further documentation available for this predicate.	

processed/2:	PREDICATE
The predicate is of type <i>data</i> .	
process_files_from/7:	PREDICATE
No further documentation available for this predicate. <i>Meta-predicate</i> with arguments: process_files_from(?,?,?,pred(1),pred(1),pred(1),pred(1)).	
process_file/7:	PREDICATE
No further documentation available for this predicate. <i>Meta-predicate</i> with arguments: process_file(?,?,?,pred(1),pred(1),pred(1),pred(1)).	
old_file_extension/2:	PREDICATE
No further documentation available for this predicate.	
cleanup_c_itf_data/0:	PREDICATE
No further documentation available for this predicate.	
cleanup_itf_cache/0:	PREDICATE
No further documentation available for this predicate.	
base_name/2:	PREDICATE
No further documentation available for this predicate.	
file_data/3:	PREDICATE
No further documentation available for this predicate. The predicate is of type <i>data</i> .	
compute_base_name/4:	PREDICATE
No further documentation available for this predicate.	
module_from_base/2:	PREDICATE
No further documentation available for this predicate.	
exports_pred/3:	PREDICATE
The predicate is of type <i>data</i> .	

module_error/0:	PREDICATE
No further documentation available for this predicate. The predicate is of type <i>data</i> .	
module_error/1:	PREDICATE
No further documentation available for this predicate. The predicate is of type <i>data</i> .	
expand_module_decl/5:	PREDICATE
No further documentation available for this predicate.	
read_sentence/3:	PREDICATE
No further documentation available for this predicate.	
default_package/1:	PREDICATE
No further documentation available for this predicate. The predicate is of type <i>data</i> .	
handle_exc/1:	PREDICATE
No further documentation available for this predicate.	
false/1:	PREDICATE
No further documentation available for this predicate.	
make_object1/2:	PREDICATE
No further documentation available for this predicate.	
make_po_file/1:	PREDICATE
No further documentation available for this predicate.	
activate_translation/3:	PREDICATE
No further documentation available for this predicate.	
end_goal_trans/1:	PREDICATE
No further documentation available for this predicate.	
module_expansion/9:	PREDICATE
No further documentation available for this predicate.	

c_itf_internal_pred/2: PREDICATE
 No further documentation available for this predicate.

c_itf_internal_pred_decl/1: PREDICATE
 No further documentation available for this predicate.

incore_mode_of/2: PREDICATE
 No further documentation available for this predicate. The predicate is of type *data*.

pred_module/2: PREDICATE
 No further documentation available for this predicate. The predicate is of type *data*.

location_t/1: REGTYPE
Usage:
 Identifies a source line range in a file.

```
location_t(loc(File,L1,L2)) :-
    atm(File),
    int(L1),
    int(L2).
```

location/3: PREDICATE
Usage:
 – *The following properties should hold upon exit:*
 Arg1 is an atom. (atm/1)
 Arg2 is an integer. (int/1)
 Arg3 is an integer. (int/1)
 The predicate is of type *data*.

location/1: PREDICATE
Usage: location(Loc)
 Unifies Loc with the locator of the current source line range. If this information can not be obtained, leaves Loc uninstantiated.
 – *The following properties should hold upon exit:*
 Identifies a source line range in a file.

```
location_t(loc(File,L1,L2)) :-
    atm(File),
    int(L1),
    int(L2).
```

(location_t/1)

set_ciaopp_expansion/1: No further documentation available for this predicate.	PREDICATE
multifile/1: No further documentation available for this predicate.	PREDICATE
module_loaded/4: No further documentation available for this predicate. The predicate is of type <i>data</i> .	PREDICATE
static_module/1: No further documentation available for this predicate.	PREDICATE
static_base/1: No further documentation available for this predicate.	PREDICATE
use_mod/3: No further documentation available for this predicate.	PREDICATE
use_mod_user/2: No further documentation available for this predicate.	PREDICATE
abolish_module/1: No further documentation available for this predicate.	PREDICATE
unload_mod/1: No further documentation available for this predicate.	PREDICATE
interpret_file/1: No further documentation available for this predicate. The predicate is of type <i>data</i> .	PREDICATE
interpret_module/1: No further documentation available for this predicate. The predicate is of type <i>data</i> .	PREDICATE
interpret_srcdbg/1: No further documentation available for this predicate. The predicate is of type <i>data</i> .	PREDICATE

5.3 Documentation on multifiles

define_flag/3:	PREDICATE
Usage: <code>define_flag(Flag, FlagValues, Default)</code>	
– <i>The following properties hold upon exit:</i>	
Flag is an atom.	(atm/1)
Define the valid flag values	(flag_values/1)
The predicate is <i>multifile</i> .	
 do_on_abolish/1:	 PREDICATE
Usage: <code>do_on_abolish(G)</code>	
– <i>The following properties should hold at call time:</i>	
G is a term which represents a goal, i.e., an atom or a structure.	(cgoal/1)
The predicate is <i>multifile</i> .	

5.4 Documentation on imports

This module has the following direct dependencies:

- *Application modules:*
`datafacts_rt`, `aggregates`, `translation`, `pl2wam`, `srcdbg`, `global_module_options`,
`file_buffer`, `fastrw`, `complete_dict`, `terms_io`, `system`, `dynamic_rt`, `pathnames`,
`strings`, `stream_utils`, `ctrlcclean`, `terms`, `hiordlib`, `lists`, `read`, `operators`, `build_`
`foreign_interface`, `assrt_lib`, `rtchecks_tr`, `frontend_condcomp`, `exemaker`.
- *Packages:*
`prelude`, `initial`, `condcomp`, `assertions`, `assertions/assertions_basic`, `nortchecks`,
`hiord`, `define_flag`, `datafacts`.

6 pl2wam (library)

6.1 Usage and interface

- **Library usage:**
`:- use_module(library(compiler/pl2wam)).`
- **Exports:**
 - *Predicates:*
`reset_counter/1, set_compiler_mode_out/2, set_compiler_mode/1, set_compiler_out/1, compile_clause/2, compile_clause_in_mode/3, proc_declaration/4, proc_declaration_in_mode/5, cleanup_compilation_data/0.`

6.2 Documentation on exports

reset_counter/1:	PREDICATE
No further documentation available for this predicate.	
set_compiler_mode_out/2:	PREDICATE
No further documentation available for this predicate.	
set_compiler_mode/1:	PREDICATE
No further documentation available for this predicate.	
set_compiler_out/1:	PREDICATE
No further documentation available for this predicate.	
compile_clause/2:	PREDICATE
No further documentation available for this predicate.	
compile_clause_in_mode/3:	PREDICATE
No further documentation available for this predicate.	
proc_declaration/4:	PREDICATE
No further documentation available for this predicate.	

proc_declaration_in_mode/5:

PREDICATE

No further documentation available for this predicate.

cleanup_compilation_data/0:

PREDICATE

No further documentation available for this predicate.

6.3 Documentation on imports

This module has the following direct dependencies:

– *Application modules:*

datafacts_rt, iso_misc, sort, dict, terms, write, lists, pl2wam_tables, c_itf.

– *Packages:*

prelude, initial, condcomp, assertions, assertions/assertions_basic, nortchecks, dcg, datafacts.

7 Emulator description language

Author(s): Jose F. Morales.

This package implements a reduced version of the emulator generator and emulator description language described in [Mor10, MCH09].

7.1 Usage and interface

- **Library usage:**
 `:- use_package(compiler/emugen).`
 or
 `:- module(...,...,[compiler/emugen]).`
- **Implicit imports:**
 - *Packages:*
 `prelude, initial, condcomp, assertions, assertions/assertions_basic.`

8 The Ciao engine

Eventually, this should contain a description of the engine, as a chapter of the Reference Manual.

NOTE: After Ciao v0.8 and before switching to SVN and Git, changes to the engine C files where documented in the `Manifest/GlobalChangeLog` file.

References

- [MCH09] J.F. Morales, M. Carro, and M. V. Hermenegildo.
Description and Optimization of Abstract Machines in a Dialect of Prolog.
Technical Report CLIP4/2009.0, Technical University of Madrid (UPM), School of
Computer Science, UPM, October 2009.
- [Mor10] J.F. Morales.
Advanced Compilation Techniques for Logic Programming.
PhD thesis, Universidad Politécnica de Madrid (UPM), Facultad Informática UPM,
28660-Boadilla del Monte, Madrid-Spain, July 2010.

Library/Module Index

C

c_itf	9
compiler	5

E

emugen	21
exemaker	7

P

pl2wam	19
--------------	----

Predicate Index

A

abolish_module/1	17
activate_translation/3	15
add_module_check/1	13
adds/2	10

B

base_name/2	14
-------------------	----

C

c_itf_internal_pred/2	16
c_itf_internal_pred_decl/1	16
clause_of/7	11
cleanup_c_itf_data/0	14
cleanup_compilation_data/0	20
cleanup_itf_cache/0	14
comp_defines/1	12
compile_clause/2	19
compile_clause_in_mode/3	19
compiler_name/1	9
compiler_version/1	9
compute_base_name/4	14
create_interfaces/0	7

D

decl/2	10
def_multifile/4	10
default_package/1	15
define_flag/3	8, 18
define_ops/0	13
defines/3	13
defines/5	12
defines_module/2	10
defines_pred/3	12
discontiguous/3	13
do_on_abolish/1	18
dyn_decl/4	12
dynamic_search_path/1	7

E

end_goal_trans/1	15
ensure_imported/4	13
ensure_loaded/1	5
ensure_loaded/2	5
expand_module_decl/5	15
exports/5	10

exports_pred/3	14
----------------------	----

F

false/1	15
file_data/3	14
force_lazy/1	7

H

handle_exc/1	15
--------------------	----

I

imports/5	13
imports_all/2	11
imports_nocheck/4	12
imports_pred/7	11
includes/2	11
incore_mode_of/2	16
interpret_file/1	17
interpret_module/1	17
interpret_srcdbg/1	17

L

loads/2	11
location/1	16
location/3	16

M

make_exec/2	7
make_object1/2	15
make_po/1	5
make_po_file/1	15
make_wam/1	5
meta_args/2	13
meta_pred/4	12
mode_of_module/2	6
module_error/0	14
module_error/1	15
module_expansion/9	15
module_from_base/2	14
module_loaded/4	17
module_of/2	6
multifile/1	17
multifile/3	13

N

needs_interface/2 7

O

old_file_extension/2 14
opt_suffix/2 10

P

package/2 11
pred_module/2 16
proc_declaration/4 19
proc_declaration_in_mode/5 19
process_file/7 14
process_files_from/7 14
processed/2 13

R

read_sentence/3 15
reset_counter/1 19
restore_defines/5 12
restore_imports/5 12
restore_multifile/4 12

S

set_ciaopp_expansion/1 16
set_compiler_mode/1 19
set_compiler_mode_out/2 19
set_compiler_out/1 19
set_debug_mode/1 6
set_debug_module/1 6
set_debug_module_source/1 6
set_nodebug_mode/1 6
set_nodebug_module/1 6
static_base/1 17
static_module/1 17

U

undo_force_lazy/1 7
unload/1 6
unload_mod/1 17
use_mod/3 17
use_mod_user/2 17
use_module/1 5
use_module/2 5
use_module/3 5
uses/2 10
uses_file/2 10

Property Index

(Index is nonexistent)

Regular Type Index

`location_t/1`..... 16

Declaration Index

(Index is nonexistent)

Concept Index

(Index is nonexistent)

Author Index

J

Jose F. Morales 21

T

The Ciao Development Team 5, 9

Global Index

This is a global index containing pointers to places where concepts, predicates, modes, properties, types, applications, authors, etc., are referred to in the text of the document.

A

abolish_module/1 9, 17
 activate_translation/3 9, 15
 add_module_check/1 9, 13
 adds/2 9, 10
 aggregates 8, 18
 assertions 6, 8, 18, 20, 21
 assertions/assertions_basic 6, 8, 18, 20, 21
 assrt_lib 18

B

base_name/2 9, 14
 build_foreign_interface 8, 18

C

c_itf 6, 8, 9, 20
 c_itf_internal_pred/2 9, 16
 c_itf_internal_pred_decl/1 9, 16
 clause_of/7 9, 11
 cleanup_c_itf_data/0 9, 14
 cleanup_compilation_data/0 19, 20
 cleanup_itf_cache/0 9, 14
 comp_defines/1 9, 12
 compile_clause/2 19
 compile_clause_in_mode/3 19
 compiler 5
 compiler_name/1 9
 compiler_version/1 9
 complete_dict 18
 compute_base_name/4 9, 14
 condcomp 6, 8, 18, 20, 21
 create_interfaces/0 7
 ctrlcclean 8, 18

D

datafacts 6, 8, 18, 20
 datafacts_rt 6, 8, 18, 20
 dcg 20
 decl/2 9, 10
 def_multifile/4 9, 10
 default_package/1 9, 15
 define_flag 8, 18
 define_flag/3 7, 8, 9, 18
 define_ops/0 9, 13

defines/3 9, 13
 defines/5 9, 12
 defines_module/2 9, 10
 defines_pred/3 9, 12
 dict 20
 discontinuous/3 9, 13
 do_on_abolish/1 9, 18
 dyn_decl/4 9, 12
 dynamic_rt 18
 dynamic_search_path/1 7

E

emugen 21
 end_goal_trans/1 9, 15
 engine_path 8
 ensure_imported/4 9, 13
 ensure_loaded/1 5, 10
 ensure_loaded/2 5
 exemaker 6, 7, 18
 expand_module_decl/5 9, 15
 exports/5 9, 10
 exports_pred/3 9, 14

F

false/1 15
 fastrw 18
 file_buffer 8, 18
 file_data/3 9, 14
 force_lazy/1 7
 frontend_condcomp 18

G

global_module_options 18

H

handle_exc/1 9, 15
 hiord 18
 hiordlib 18

I

import/2	12
imports/5	9, 12, 13
imports_all/2	9, 11
imports_nocheck/4	9, 12
imports_pred/7	9, 11
includes/2	9, 11
incore_mode_of/2	9, 16
initial	6, 8, 18, 20, 21
interpret_file/1	9, 17
interpret_module/1	9, 17
interpret_srcdbg/1	9, 17
iso_misc	20

J

jfran-phd	21
Jose F. Morales	21

L

lists	8, 18, 20
load_compilation_module/1	13
loads/2	9, 11
location/1	9, 16
location/3	9, 16
location_t/1	9, 16

M

make_exec/2	7
make_object1/2	9, 15
make_po/1	5
make_po_file/1	9, 15
make_wam/1	5
meta_args/2	9, 13
meta_pred/4	9, 12
mode_of_module/2	5, 6
module_error/0	9, 13, 14
module_error/1	9, 15
module_expansion/9	9, 15
module_from_base/2	9, 14
module_loaded/4	9, 17
module_of/2	5, 6
morales09:improlog-tr	21
multifile/1	9, 17
multifile/3	9, 13

multifile/4	12
-------------	----

N

needs_interface/2	7
nortchecks	18, 20

O

old_file_extension/2	9, 14
operators	18
opt_suffix/2	9, 10

P

package/2	9, 11
pathnames	18
pl2wam	8, 9, 18, 19
pl2wam_tables	20
pred_module/2	9, 16
prelude	6, 8, 18, 20, 21
proc_declaration/4	19
proc_declaration_in_mode/5	19
process_file/7	9, 14
process_files_from/7	9, 14
processed/2	9, 13

R

read	18
read_sentence/3	9, 15
reset_counter/1	19
restore_defines/5	9, 12
restore_imports/5	9, 12
restore_multifile/4	9, 12
rtchecks_tr	18

S

set_ciaopp_expansion/1	9, 16
set_compiler_mode/1	19
set_compiler_mode_out/2	19
set_compiler_out/1	19
set_debug_mode/1	5, 6
set_debug_module/1	5, 6
set_debug_module_source/1	5, 6
set_nodebug_mode/1	5, 6
set_nodebug_module/1	5, 6

sort 20
 srcdbg 18
 static_base/1 9, 17
 static_module/1 9, 17
 stream_utils 8, 18
 strings 18
 system 6, 8, 18

T

terms 18, 20
 terms_io 18
 The Ciao Development Team 5, 9
 toplevel 5
 translation 18

U

undo_force_lazy/1 7
 unload/1 5, 6
 unload_mod/1 9, 17
 use_mod/3 9, 17
 use_mod_user/2 9, 17
 use_module/1 5, 11
 use_module/2 5, 11
 use_module/3 5
 uses/2 9, 10
 uses_file/2 9, 10

W

wamql 9
 write 20

