

**1. License.**

Generated date: January 30, 2015

Copyright © 1998-2015 Dave Bone

This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at <http://mozilla.org/MPL/2.0/>.

## 2. Summary Linker External parse routines.

These are the various procedures that parse Yacco2's grammar language and emit the grammar's c++ code and tex document with mpost generated diagrams. Each language construct has its appropriate external procedure that houses the monolithic grammar to start the parse. There is no namespace used to contain these routines as I felt that this was overkill. As this is a closed system, their grammars are not universal and cannot be re-cycled for others. Their only outside value are in teaching examples on "how to skin a cat" or is it "how to parse a lion?" ...

External routines ratatouille:

*o2linker.externs.w* - cweb generator file

*o2linker.externs.h* - header file

*o2linker.externs.cpp* - implementation

Dependency files from other Yacco2 sub-systems:

*yacco2.h* - basic definitions used by Yacco2

*yacco2.T.enumeration.h* - terminal enumeration for Yacco2's terminal grammar alphabet

*yacco2.err.symbols.h* - error terminal definitions from Yacco2's grammar alphabet

*yacco2.characters.h* - raw character definitions from Yacco2's grammar alphabet

*yacco2.k.symbols.h* - constant meta terminal defs from Yacco2's grammar alphabet

*yacco2.terminals.h* - regular terminal definitions from Yacco2's grammar alphabet

\*. *h* - assorted grammar definitions for Yacco2's parsing

*yacco2.stbl.h* - symbol table definitions

External procedures and other globals:

LINKER\_PARSE\_CMD\_LINE

### 3. Global definitions, External parse routines for Yacco2.

#### 4. Create header file.

```

<o2linker_externs.h 4> ≡
#ifdef o2linker_externs_
#define o2linker_externs_ 1
    <Preprocessor definitions>
    <Files for header 5>;
#endif

```

#### 5. Files for header.

```

<Files for header 5> ≡
#include "globals.h"
#include "o2linker_types.h"
#include "yacco2_stbl.h"
#include "o2_linker_opts.h"
#include "linker_pass3.h"
extern STBL_T_ITEMS.type STBL_T_ITEMS;
extern void GET_CMD_LINE(int argc, char *argv[], const char *File, yacco2 ::TOKEN_GAGGLE & Errors);
extern void LINKER_PARSE_CMD_LINE(const char *Fsc, std ::string & Cntl_file,
    yacco2 ::TOKEN_GAGGLE & Errors);
extern void DUMP_ERROR_QUEUE(yacco2 ::TOKEN_GAGGLE & Errors);
extern const char *DATE_AND_TIME();
extern void XLATE_SYMBOLS_FOR_cweave(const char *Sym_to_xlate, char *Xlated_sym);
extern yacco2 ::AST *CWEB_MARKER;

```

This code is used in section 4.

#### 6. Include Header file.

```

<Include Header file 6> ≡
#include "o2linker_externs.h"

```

This code is used in section 7.

#### 7. Yacco2 external routines blueprint. Output of the code.

```

<o2linker_externs.cpp 7> ≡
    <Include Header file 6>;
    <accrue source for emit 8>;

```

#### 8. Accrue source for emit.

```

<accrue source for emit 8> ≡
    yacco2 ::AST *CWEB_MARKER(0);

```

See also section 10.

This code is used in section 7.

**9. Local o2linker routines.****10. Linker Parse command line: LINKER\_PARSE\_CMD\_LINE.**

The parameters have been extracted from the program run environment and placed into Linker's holding file. The routine is suitable for public consumption if a filename parameter is needed.

Constraints:

- ip1: Linker's holding file containing the command line to compile
- ip2: Extracted fsc file to compile
- ip3: Error container for generated errors

Errors:

- 1) bad filename
- 2) parameters errors

(accrue source for emit 8) +=

```
extern void LINKER_PARSE_CMD_LINE
(const char *Cmd_file, std::string & Ctrl_file, yacco2::TOKEN_GAGGLE & Errors)
{
    using namespace NS_yacco2_err_symbols;
    using namespace yacco2;
    tok_can < std::ifstream > Cmd1_tokens(Cmd_file);
    if (Cmd1_tokens.file_ok() == NO) {
        yacco2::Delete_tokens(Cmd1_tokens.container());
        CAbs_lr1_sym * sym = new Err_bad_filename(Cmd_file);
        sym->set_external_file_id(1);
        sym->set_line_no(1);
        sym->set_pos_in_line(1);
        Errors.push_back(*sym);
        return;
    }
    using namespace NS_o2_linker_opts;
    TOKEN_GAGGLE linker_options_tokens;
    Co2_linker_opts opts_fsm;
    Parser_options(opts_fsm, &Cmd1_tokens, &linker_options_tokens, 0, &Errors, 0, 0);
    options.parse();
    yacco2::Delete_tokens(Cmd1_tokens.container());
    if (Errors.empty() != YES) return;
    Ctrl_file += opts_fsm.file_to_compile_;
    return;
}
```

**11. Bric-a-brac.**

**12. Index.**

*argc*: [5](#).  
*argv*: [5](#).  
*AST*: [5](#), [8](#).  
*CAbs\_lr1\_sym*: [10](#).  
*Cmd\_file*: [10](#).  
*Cmd1\_tokens*: [10](#).  
*Cntl\_file*: [5](#).  
*container*: [10](#).  
*Co2\_linker\_opts*: [10](#).  
*cpp*: [2](#).  
*Ctrl\_file*: [10](#).  
*CWEB\_MARKER*: [5](#), [8](#).  
*DATE\_AND\_TIME*: [5](#).  
*Delete\_tokens*: [10](#).  
*DUMP\_ERROR\_QUEUE*: [5](#).  
*empty*: [10](#).  
*Err\_bad\_filename*: [10](#).  
*Errors*: [5](#), [10](#).  
*File*: [5](#).  
*file\_ok*: [10](#).  
*file\_to\_compile\_*: [10](#).  
*Fsc*: [5](#).  
*GET\_CMD\_LINE*: [5](#).  
*ifstream*: [10](#).  
*linker\_options\_tokens*: [10](#).  
*LINKER\_PARSE\_CMD\_LINE*: [2](#), [5](#), [10](#).  
*NO*: [10](#).  
*NS\_o2\_linker\_opts*: [10](#).  
*NS\_yacco2\_err\_symbols*: [10](#).  
*options*: [10](#).  
*opts\_fsm*: [10](#).  
*o2linker\_externs*: [2](#).  
*o2linker\_externs\_*: [4](#).  
*parse*: [10](#).  
*Parser*: [10](#).  
*push\_back*: [10](#).  
*set\_external\_file\_id*: [10](#).  
*set\_line\_no*: [10](#).  
*set\_pos\_in\_line*: [10](#).  
*STBL\_T\_ITEMS*: [5](#).  
*STBL\_T\_ITEMS\_type*: [5](#).  
*std*: [5](#), [10](#).  
*string*: [5](#), [10](#).  
*sym*: [10](#).  
*Sym\_to\_xlate*: [5](#).  
*tok\_can*: [10](#).  
*TOKEN\_GAGGLE*: [5](#), [10](#).  
*XLATE\_SYMBOLS\_FOR\_cweave*: [5](#).  
*Xlated\_sym*: [5](#).  
**yacco2**: [2](#), [5](#), [8](#), [10](#).  
*yacco2\_characters*: [2](#).  
*yacco2\_err\_symbols*: [2](#).  
*yacco2\_k\_symbols*: [2](#).  
*yacco2\_stbl*: [2](#).  
*yacco2\_T\_enumeration*: [2](#).  
*yacco2\_terminals*: [2](#).  
*YES*: [10](#).

⟨ Files for header 5 ⟩ Used in section 4.  
⟨ Include Header file 6 ⟩ Used in section 7.  
⟨ accrue source for emit 8, 10 ⟩ Used in section 7.  
⟨ o2linker\_externs.cpp 7 ⟩  
⟨ o2linker\_externs.h 4 ⟩

# O2LINKER EXTERNS

	Section	Page
<b>License</b> .....	<a href="#">1</a>	1
<b>Summary Linker External parse routines</b> .....	<a href="#">2</a>	2
<b>Global definitions, External parse routines for Yacco2</b> .....	<a href="#">3</a>	3
Files for header .....	<a href="#">5</a>	3
<b>Local o2linker routines</b> .....	<a href="#">9</a>	4
Linker Parse command line: LINKER_PARSE_CMD_LINE .....	<a href="#">10</a>	4
<b>Bric-a-brac</b> .....	<a href="#">11</a>	5
<b>Index</b> .....	<a href="#">12</a>	6