

1. Copyright.

Copyright © Dave Bone 1998 - 2014

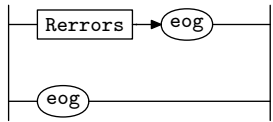
2. *o2_err_hdlr* grammar.

Grammar that processes for printing posted errors from yacc2.

3. Fsm Ctestout_err_hdlr class.

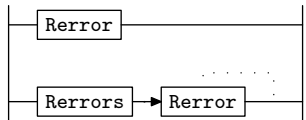
4. *Rtestout_err_hdlr* rule.

Rtestout_err_hdlr



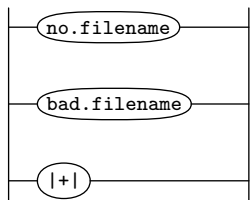
5. *Rerrors* rule.

Rerrors



6. *Rerror* rule.

Rerror



7. Rerror user-declaration directive.

⟨Rerror user-declaration directive 7⟩ ≡

```

void error_where(CAbs_lr1_sym * E_sym)
{
    std :: string ext_fle;
    if (E_sym-tok_co_ords...external_file_id__ < yacco2 :: FILE_TBL__.size()) {
        ext_fle = yacco2 :: FILE_TBL__[E_sym-tok_co_ords...external_file_id__];
    }
    else {
        ext_fle = "File_not_found";
    }
    std :: string line_of_data;
    std :: ifstream ifile;
    ifile.open(ext_fle.c_str());
    if (ifile.good()) {
        yacco2 :: UINT lno(1);
        yacco2 :: UINT dlno(E_sym-tok_co_ords...line_no__);
        for (; lno ≤ dlno; ++lno) {
            getline(ifile, line_of_data);
            if (lno ≡ dlno) break;
            line_of_data.clear();
        }
    }
    std :: string space(" ");
    std :: string :: size_type f = line_of_data.find_first_of('\\t');
    for (; f ≠ std :: string :: npos; ) {
        line_of_data.replace(f, 1, space);
        f = line_of_data.find_first_of('\\t');
    }
    yacco2 :: lrlog << "Error_in_file#:_" << E_sym-tok_co_ords...external_file_id__ << "\\\" <<
        ext_fle.c_str() << "\\\" << std :: endl;
    yacco2 :: lrlog << line_of_data.c_str() << std :: endl;
    std :: cout << "Error_in_file#:_" << E_sym-tok_co_ords...external_file_id__ << "\\\" <<
        ext_fle.c_str() << "\\\" << std :: endl;
    std :: cout << line_of_data.c_str() << std :: endl;
    for (int pos = 1; pos < E_sym-tok_co_ords...pos_in_line__; ++pos) {
        yacco2 :: lrlog << ' ';
        std :: cout << ' ';
    }
    yacco2 :: lrlog << '^' << std :: endl;
    yacco2 :: lrlog << "\\tfpos:_" << E_sym-tok_co_ords...rc_pos__ << "_line#:" <<
        E_sym-tok_co_ords...line_no__ << "_cpos:_" << E_sym-tok_co_ords...pos_in_line__ << std :: endl;
    std :: cout << '^' << std :: endl;
    std :: cout << "\\tfpos:_" << E_sym-tok_co_ords...rc_pos__ << "_line#:" <<
        E_sym-tok_co_ords...line_no__ << "_cpos:_" << E_sym-tok_co_ords...pos_in_line__ << std :: endl;
    if (E_sym-tok_co_ords...who_file__ ≠ 0) {
        yacco2 :: lrlog << "\\twho_thru_it:_" << E_sym-tok_co_ords...who_file__ << "_line#:" <<
            E_sym-tok_co_ords...who_line_no__ << std :: endl;
        std :: cout << "\\twho_thru_it:_" << E_sym-tok_co_ords...who_file__ << "_line#:" <<
            E_sym-tok_co_ords...who_line_no__ << std :: endl;
    }
}
ifile.close();

```

```
}
;
```

8. Error's subrule 1.

```
|no.filename|
1
```

```
<Error subrule 1 op directive 8> ≡
error_where(sf-p1-);
yacco2::lrclg << "\t" << sf-p1--id- << std::endl;
std::cout << "\t" << sf-p1--id- << std::endl;
```

9. Error's subrule 2.

```
|bad.filename|
1
```

```
<Error subrule 2 op directive 9> ≡
error_where(sf-p1-);
yacco2::lrclg << "\t" << sf-p1--id- << "_filename:\\" << sf-p1--file_name()-c_str() <<
"\_does\_not\_exist" << std::endl;
std::cout << "\t" << sf-p1--id- << "_filename:\\" << sf-p1--file_name()-c_str() <<
"\_does\_not\_exist" << std::endl;
```

10. Error's subrule 3.

```
|(|+)|
1
```

```
<Error subrule 3 op directive 10> ≡
error_where(sf-p1-);
yacco2::lrclg << "\t" << sf-p1--id- << std::endl;
std::cout << "\t" << sf-p1--id- << std::endl;
```

11. First Set Language for O_2^{linker} .

```
/*
  File: testout_err_hdlr.fsc
  Date and Time: Fri Jan  2 09:49:46 2015
*/
transitive      n
grammar-name    "testout_err_hdlr"
name-space     "NS_testout_err_hdlr"
thread-name    "Ctestout_err_hdlr"
monolithic     y
file-name      "testout_err_hdlr.fsc"
no-of-T        269
list-of-native-first-set-terminals 4
  LR1_eog
  LR1_all_shift_operator
  Err_no_filename
  Err_bad_filename
end-list-of-native-first-set-terminals
list-of-transitive-threads 0
end-list-of-transitive-threads
list-of-used-threads 0
end-list-of-used-threads
fsm-comments
"Logic sequencer: Print out errors from testout grammars - pager_1."
```

12. Lr1 State Network.

\Rightarrow					State: 1 state type: ^s			
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA	
c	Rtestout_err_hdlr		1 2 1	eog			1 2 2	
c	Rerror		3 3 1	+			1 3 3	
c	Rerror		3 1 1	no filename			1 4 4	
c	Rerror		3 2 1	bad filename			1 5 5	
c	Rerrors		2 2 1	Rerrors <u>Rerror</u>			1 6 8	
c	Rtestout_err_hdlr		1 1 1	Rerrors <u>eog</u>			1 6 7	
c	Rerrors		2 1 1	Rerror			1 9 9	
\Rightarrow	<i>eog</i>				State: 2 state type: ^r			
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA	
t	Rtestout_err_hdlr		1 2 2				1 0 2 1	
\Rightarrow	+				State: 3 state type: ^r			
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA	
t	Rerror		3 3 2				1 0 3 2	
\Rightarrow	<i>nofilename</i>				State: 4 state type: ^r			
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA	
t	Rerror		3 1 2				1 0 4 2	
\Rightarrow	<i>badfilename</i>				State: 5 state type: ^r			
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA	
t	Rerror		3 2 2				1 0 5 2	
\Rightarrow	<i>Rerrors</i>				State: 6 state type: ^s			
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA	
t	Rtestout_err_hdlr		1 1 2	eog			1 7 7	
c	Rerror		3 3 1	+			6 3 3	
c	Rerror		3 1 1	no filename			6 4 4	
c	Rerror		3 2 1	bad filename			6 5 5	
t	Rerrors		2 2 2	Rerror			1 8 8	
\Rightarrow	<i>eog</i>				State: 7 state type: ^r			
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA	
t	Rtestout_err_hdlr		1 1 3				1 0 7 1	
\Rightarrow	<i>Rerror</i>				State: 8 state type: ^r			
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA	
t	Rerrors		2 2 3				1 0 8 2	
\Rightarrow	<i>Rerror</i>				State: 9 state type: ^r			
\leftarrow	rule	\rightarrow	R# sr# Po	\leftarrow	subrule element	\rightarrow	Brn Gto Red LA	
t	Rerrors		2 1 2				1 0 9 2	

13. Index.

|+|: 6.
bad filename: 6.
c_str: 7, 9.
CAbs_lr1_sym: 7.
clear: 7.
close: 7.
cout: 7, 8, 9, 10.
dlno: 7.
E_sym: 7.
endl: 7, 8, 9, 10.
eog: 4.
error_where: 7, 8, 9, 10.
ext_fle: 7.
external_file_id__: 7.
file_name: 9.
FILE_TBL__: 7.
find_first_of: 7.
getline: 7.
good: 7.
id__: 8, 9, 10.
ifile: 7.
ifstream: 7.
line_no__: 7.
line_of_data: 7.
lno: 7.
lrlog: 7, 8, 9, 10.
no filename: 6.
npos: 7.
open: 7.
o2_err_hdlr: 2.
pos: 7.
pos_in_line__: 7.
p1__: 8, 9, 10.
rc_pos__: 7.
replace: 7.
Rerror: 5.
Rerror: 6, 8, 9, 10.
Rerrors: 5.
Rerrors: 4, 5.
Rtestout_err_hdlr: 4.
sf: 8, 9, 10.
size: 7.
size_type: 7.
space: 7.
std: 7, 8, 9, 10.
string: 7.
tok_co_ords__: 7.
UINT: 7.
who_file__: 7.
who_line_no__: 7.
yacco2: 7, 8, 9, 10.

- ⟨ Error subrule 1 op directive 8 ⟩
- ⟨ Error subrule 2 op directive 9 ⟩
- ⟨ Error subrule 3 op directive 10 ⟩
- ⟨ Error user-declaration directive 7 ⟩

testout_err_hdlr Grammar

Date: January 2, 2015 at 11:27

File: testout_err_hdlr.lex

Ns: NS_testout_err_hdlr

Version: 1.0

Debug: false

Grammar Comments:

Type: Monolithic

Logic sequencer: Print out errors from testout grammars - pager.1.

	Section	Page
Copyright	1	1
<i>o2_err_hdlr</i> grammar	2	2
Fsm Ctestout_err_hdlr class	3	2
<i>Rtestout_err_hdlr</i> rule	4	2
<i>Rerrors</i> rule	5	2
<i>Rerror</i> rule	6	2
Error user-declaration directive	7	3
<i>Rerror</i> 's subrule 1	8	4
<i>Rerror</i> 's subrule 2	9	4
<i>Rerror</i> 's subrule 3	10	4
First Set Language for O_2^{linker}	11	5
Lr1 State Network	12	6
Index	13	7