
Subject: Re: the type parser

Posted by [marc schellens\[1\]](#) on Mon, 16 May 2005 09:26:24 GMT

[View Forum Message](#) <> [Reply to Message](#)

IDL scans the strings.

Its not really difficult but due to the many ways constants can be defined

in IDL it gets a little bit complicated.

Here is the relevant GDL

(<http://sourceforge.net/projects/gnudatalanguage>)

code (in antlr, <http://www.antlr.org>) wich is 100% compatible to IDL.

The assignment to `_ttype` sets the determined type.

If you translate this into IDL you got what you need.

HDH,

marc

```
protected
```

```
D
```

```
: ('0'..'9')
```

```
;
```

```
protected
```

```
L
```

```
: ('a'..'z'|'_')
```

```
;
```

```
protected
```

```
H
```

```
: ('a'..'f'|'0'..'9')
```

```
;
```

```
protected
```

```
O
```

```
: ('0'..'7')
```

```
;
```

```
protected
```

```
EXP
```

```
: ('e' ('+'|'-)? (D)+)
```

```
;
```

```
protected
```

```
DBL_E
```

```
: 'd' { $setText( "E");}
```

```
;
```

```
protected
```

```

DBL
: (DBL_E (('+'|'|'-')? (D)+)?)
;

protected
CONSTANT_HEX_BYTE;;
protected
CONSTANT_HEX_LONG;;
protected
CONSTANT_HEX_LONG64;;
protected
CONSTANT_HEX_I;; // integer or larger
protected
CONSTANT_HEX_INT;;
protected
CONSTANT_HEX_ULONG;;
protected
CONSTANT_HEX_ULONG64;;
protected
CONSTANT_HEX_UI;;
protected
CONSTANT_HEX_UINT;;
protected
CONSTANT_BYTE;;
protected
CONSTANT_LONG;;
protected
CONSTANT_LONG64;;
protected
CONSTANT_I;; // integer or larger if necessary
protected
CONSTANT_INT;;
protected
CONSTANT_ULONG;;
protected
CONSTANT_ULONG64;;
protected
CONSTANT_UI;;
protected
CONSTANT_UINT;;
protected
CONSTANT_OCT_BYTE;;
protected
CONSTANT_OCT_LONG;;
protected
CONSTANT_OCT_LONG64;;
protected
CONSTANT_OCT_I;; // integer or larger if necessary

```

```

protected
CONSTANT_OCT_INT;;
protected
CONSTANT_OCT_ULONG;;
protected
CONSTANT_OCT_ULONG64;;
protected
CONSTANT_OCT_UI;;
protected
CONSTANT_OCT_UINT;;
protected
CONSTANT_FLOAT;;
protected
CONSTANT_DOUBLE;;
protected
STRING_LITERAL;;
protected
DOT;;

```

```

CONSTANT_OR_STRING_LITERAL
// could be a string, but octals have priority
: (\'"(O)+ ( 'b' | 's' | "us" | "ub" | 'l' | 'u' | "ul" )?) =>
( \'"! (O)+ { _ttype=CONSTANT_OCT_I; } // DEFINT32
( 's'! { _ttype=CONSTANT_OCT_INT; }
| 'b'! { _ttype=CONSTANT_OCT_BYTE; }
| 'u'! { _ttype=CONSTANT_OCT_UI; } // DEFINT32
| "us"! { _ttype=CONSTANT_OCT_UINT; }
| "ub"! { _ttype=CONSTANT_OCT_BYTE; }
| 'l'! { _ttype=CONSTANT_OCT_LONG; }
| "ll"! { _ttype=CONSTANT_OCT_LONG64; }
| "ul"! { _ttype=CONSTANT_OCT_ULONG; }
| "ull"! { _ttype=CONSTANT_OCT_ULONG64; }
)?)
| (\'"(H)+\'" ( 'x' | "xs" | "xb" | "xl" | "xu" | "xus" | "xub" |
"xul" )) =>
( \'"! (H)+ \'! 'x'!
( { _ttype=CONSTANT_HEX_I; } // DEFINT32
| 's'! { _ttype=CONSTANT_HEX_INT; }
| 'b'! { _ttype=CONSTANT_HEX_BYTE; }
| 'u'! { _ttype=CONSTANT_HEX_UI; } // DEFINT32
| "us"! { _ttype=CONSTANT_HEX_UINT; }
| "ub"! { _ttype=CONSTANT_HEX_BYTE; }
| 'l'! { _ttype=CONSTANT_HEX_LONG; }
| "ll"! { _ttype=CONSTANT_HEX_LONG64; }
| "ul"! { _ttype=CONSTANT_HEX_ULONG; }
| "ull"! { _ttype=CONSTANT_HEX_ULONG64; }
))
| (\'"(O)+\'" ( 'o' | "os" | "ol" | "ou" | "oul" )) =>

```

```

('"! (O)+ \'! 'o!
( { _ttype=CONSTANT_OCT_I; } // DEFINT32
| 's! { _ttype=CONSTANT_OCT_INT; }
| 'b! { _ttype=CONSTANT_OCT_BYTE; }
| 'u! { _ttype=CONSTANT_OCT_UI; } // DEFINT32
| "us"! { _ttype=CONSTANT_OCT_UINT; }
| "ub"! { _ttype=CONSTANT_OCT_BYTE; }
| 'l! { _ttype=CONSTANT_OCT_LONG; }
| "ll"! { _ttype=CONSTANT_OCT_LONG64; }
| "ul"! { _ttype=CONSTANT_OCT_ULONG; }
| "ull"! { _ttype=CONSTANT_OCT_ULONG64; }
))
// strings in IDL do not need trailing " or '
| \'! (~(\'"|\r|\n)| \' \'! )*
( \'!
|
) { _ttype=STRING_LITERAL; }
| \'! (~(\'"|\r|\n)| \' \'! )*
( \'!
|
) { _ttype=STRING_LITERAL; }
| (((D)+ (DBL | '.(D)*(DBL))) | '.(D)+(DBL)) =>
(
(
(D)+
(DBL
| '.(D)*(DBL)
)
)
| '.(D)+(DBL))
{ _ttype=CONSTANT_DOUBLE; }
| (((D)+ (EXP | '.(D)*(EXP?)) | '.(D)+(EXP?)) =>
(
(
(D)+
(EXP
| '.(D)*(EXP?)
)
)
| '.(D)+(EXP?)
{ _ttype=CONSTANT_FLOAT; }
| '.' { _ttype=DOT; }
| (D)+ { _ttype=CONSTANT_I; }
| 's! { _ttype=CONSTANT_INT; }
| 'b! { _ttype=CONSTANT_BYTE; }
| 'u('s')?! { _ttype=CONSTANT_UINT; }
| "ub"! { _ttype=CONSTANT_BYTE; }
| 'l! { _ttype=CONSTANT_LONG; }

```

```
| "l"!    { _ttype=CONSTANT_LONG64; }  
| "ul"!   { _ttype=CONSTANT_ULONG; }  
| "ull"!  { _ttype=CONSTANT_ULONG64; }  
)?  
;
```
