

## Re: True Name Spaces, when?

---

*Source:*

<http://coding.derkeiler.com/Archive/Delphi/borland.public.delphi.non-technical/2008-02/msg02127.html>

---

- *From:* Hans-Peter Diettrich <[DrDiettrich1@xxxxxxxx](mailto:DrDiettrich1@xxxxxxxx)>
  - *Date:* Tue, 19 Feb 2008 01:19:57 +0100
- 

L wrote:

Just to confuse people.. I want to clear (or not clear) something up about single pass parsers.

The whole "single pass" phrase is kind of a misleading term..IMO.

Well, I never found "single pass" applied to parsers, only to compilers.

In the early days of computing a complete compiler didn't fit into memory, so it had to be splitted into multiple passes. In this case every pass used the output of the preceding passes, and did \*not\* parse the source files again.

IMO, during parsing there are several "inner sub passes" to make sure syntax is right.. and things get fastforwarded/rewound while you are analyzing stuff after or during the pass.. so it is not really a single pass at all.

Then you'd have two distinct parsers, derived from distinct grammars. The first one checks for correct syntax only, the second one only processes syntactically correct source files.

Of course there must exist backtracking or other techniques, when a grammar requires lookahead in order to determine the applicable rules. The improper use of "productions" for parsing rules results from the concept of languages, whose grammars can be used to either produce or analyze sentences of the language. A parser instead does not produce sentences, it only processes given sentences.

There are many cases for example when someone uses a `StringReplace()` during a parse, which is actually is resetting part of the pass to not be a single pass any more... since you are rewinding when a replace is done.

A parser never modifies the source. What you mean is an application, that makes use of an parser. When that application modifies the input, it must resume parsing of the modified source at an appropriate place.

Re: True Name Spaces, when?

DoDi

.