U++ - Task #1479
Core get ride of "extra ;" warning with pedantic mode

06/25/2016 02:26 PM - Zbigniew Rebacz

Status:	Patch ready	**Start date:**	06/25/2016
Priority:	Low	**Due date:**	
Assignee:	Miroslav Fidler	**% Done:**	0%
Category:	Core	**Estimated time:**	0.00 hour
Target version:	Release 2021.1 - NTH	**Spent time:**	0.00 hour

Description

Compile with GCC and -Wpedantic to detects all warning related to "extra ;"

History

#1 - 06/25/2016 02:26 PM - Zbigniew Rebacz

- Tracker changed from Bug to Task

#2 - 02/05/2017 11:21 AM - Zbigniew Rebacz

- Target version changed from Release 2017.1 - NTH to Release 2017.2 - NTH

#3 - 12/24/2017 06:36 PM - Zbigniew Rebacz

- Target version changed from Release 2017.2 - NTH to Release 2018.1 - NTH

#4 - 04/03/2018 10:00 PM - Zbigniew Rebacz

- Target version changed from Release 2018.1 - NTH to Release 2019.1 - NTH

#5 - 06/24/2019 01:29 AM - Zbigniew Rebacz

- Target version changed from Release 2019.1 - NTH to Release 2019.2 - NTH

#6 - 10/08/2019 05:07 PM - Sender Ghost

- File uppsrc.diff added

The GCC 8.3.0 compiler identified following cases with extra ';' character:

```
namespace * {*};
extern "C" {*};
INITIALIZE(*);
INITBLOCK {*};
rval_default(*);
CH_VALUE(*);
CH_IMAGE(*);
CH_COLOR(*);
CH_INT(*);
NTL_MOVEABLE(*);
SVO_FN(*);
VALUE_XMLIZE(*);
__Expand(*);
INI_TYPE(*);
INI_BOOL(*);
INI_INT(*);
INI_STRING(*);
```

and where INI_TYPE define was used manually:

```
    INI_INT64(*);
    INI_DOUBLE(*);
```

Patch attached.
Please use "svn patch" (or patch, etc.) utility to apply patches, because current (at 13633 revision) version of TheIDE's "Assist -> Display/apply patch.." function doesn't recognize "\ No newline at end of file" line(s) in patches, created by diff, "git diff" or "svn diff" programs.

#7 - 10/09/2019 04:55 AM - Sender Ghost

- Status changed from New to Patch ready

#8 - 10/09/2019 10:04 AM - Miroslav Fidler

Yeah, well, but I would like to actually have ';' after many of these macros, for two reasons:

- it is more readable (; being a natural item ending character in C++)
- futureproofing - it is not unlikely that some day in the future implementation changes. well, unlikely, but still

Maybe we should instead find a way how to require ';' there?

#9 - 10/09/2019 10:15 AM - Sender Ghost

I think, the namespace and extern are valid cases. Possible to create another patch, which excludes changes for macros.

#10 - 10/09/2019 01:41 PM - Sender Ghost

- File uppsrc_excluded_macros.diff added

Attached new patch with excluded changes for macros.

#11 - 11/01/2019 11:21 AM - Zbigniew Rebacz

- Target version changed from Release 2019.2 - NTH to Release 2020.1 - NTH

#12 - 05/24/2020 12:48 PM - Zbigniew Rebacz

- Target version changed from Release 2020.1 - NTH to Release 2020.2 - NTH

#13 - 10/21/2020 11:34 PM - Zbigniew Rebacz

- Target version changed from Release 2020.2 - NTH to Release 2021.1 - NTH

Files

uppsrc.diff	47 KB	10/08/2019	Sender Ghost
uppsrc_excluded_macros.diff	18 KB	10/09/2019	Sender Ghost