CoCoA-5 - Feature #1305

Create a directory

02 Sep 2019 15:39 - John Abbott

Status:	New	**Start date:**	02 Sep 2019
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	10%
Category:	CoCoA-5 function: new	**Estimated time:**	0.00 hour
Target version:	CoCoA-5.?.?	**Spent time:**	0.35 hour

Description

Ange asked about a command in CoCoA for creating a directory.
Do we want to offer such a command?
Perhaps OpenOFile should automatically create all necessary directories (perhaps depending on an arg?)?

Related issues:

Related to CoCoALib - Design #1225: Move to C++14 (skipping C++11)	**In Progress**	**06 Sep 2018**

History

#1 - 02 Sep 2019 15:45 - John Abbott

It seems that C++17 includes a portable command for creating directories (create_directory), so this would make portable impl easy.

BOOST also offers a command for creating directories.

If we impl a separate command the users would write code like this:

```
CreateDirectory("NewDir");
output := OpenOFile("NewDir/filename");
println answer on output;
```

If we let OpenOFile automatically create all dirs then the users would write code like this:

```
output := OpenOFile("NewDir/filename", "CreateDirs");
println answer on output;
```

Comments, ideas, criticisms?

#2 - 02 Sep 2019 15:56 - Anna Maria Bigatti

I agree it would be nice.
Do you want to do it now, or wait a bit before going all the way to C++17?

#3 - 02 Sep 2019 15:56 - Anna Maria Bigatti

- Related to Design #1225: Move to C++14 (skipping C++11) added

#4 - 02 Sep 2019 16:15 - John Abbott

I prefer not to jump straight to C++17 (as it is only 2019).
We can probably use the BOOST impl since CoCoA-5 requires BOOST.

Which interface do you prefer? A separate function or an option to OpenOFile?
I think it does not make much sense to left OpenIFile create new directories.

#5 - 02 Sep 2019 16:31 - Anna Maria Bigatti

- % Done changed from 0 to 10

I much prefer having CreateDirectory.
In fact, I'd like it to give a warning if the directory already exists,
(and maybe also have CreateDirectoryNoWarning, for the silent version, **if needed** ;-))

It's easier to make two functions (oneliners) than make optional arguments, and I also think it is easier to document and find in the documentation.