CoCoA-5 - Feature #8

source region

20 Oct 2011 11:51 - Anna Maria Bigatti

Status:	Closed	**Start date:**	20 Oct 2011
Priority:	Urgent	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	CoCoA-5 function: new	**Estimated time:**	0.00 hour
Target version:	CoCoA-5.0.3	**Spent time:**	25.80 hours

Description

```
SOURCE <filename>;
SOURCE <riga1> (opt: ,<col1>) TO <riga2> (opt ,<col2>) IN <filename>;
```

2013-03-11 Clarified semantics: counts of lines/cols start from 1; starting char pos is INCLUDED, ending char pos is EXCLUDED.

Related issues:

Related to CoCoA-5 - Support #198: Appunti per (eventuale) tesi triennale Inf...	**New**	**29 Jun 2012**
Related to CoCoA-5 - Bug #344: basic_string::erase error in interpreter	**Closed**	**24 Apr 2013**
Related to CoCoA-5 - Feature #352: Should SourceRegion echo the "region"?	**Closed**	**21 May 2013**
Related to CoCoA-5 - Bug #164: CoCoA-5 emacs interface bug (minor): Source & ...	**Closed**	**20 May 2012**

History

#1 - 20 Oct 2011 14:21 - Anna Maria Bigatti

- Priority changed from Normal to Urgent

#2 - 08 Nov 2011 15:35 - Anna Maria Bigatti

- Category set to CoCoA-5 function: new

#3 - 09 Nov 2011 12:14 - Anna Maria Bigatti

- % Done changed from 0 to 30

#4 - 27 Feb 2013 17:27 - John Abbott

Any chance of having this within 2-3 weeks for this year's AlgComp course?

If I want it in a hurry then I know whom to assign it to? :-)

#5 - 27 Feb 2013 18:04 - Anna Maria Bigatti

John Abbott wrote:

> Any chance of having this within 2-3 weeks for this year's AlgComp course?

well, for emacs it is already available: just select the region and press C-c C-r (r for Region)

for doing it in the interpreter I actually have no clue. Even less fo rimplementing it in the GUI.
I think we have no hope :-(

#6 - 27 Feb 2013 19:10 - John Abbott

In a sense it works in emacs, but not as I would like it to work.

I would like it to work as though the region had been saved to a file and then that file had been sourced (except for line numbers in error messages). Instead it sends the text from the region as though it had been typed in by hand; this means that CoCoA-5 prints out a prompt for each line read... I would like these prompts not to appear (as they don't when a file is read by Source).

#7 - 05 Mar 2013 21:49 - John Abbott

- Status changed from New to In Progress

I mention an alternative possible meaning for "source region". It could specify the file and start and end character positions in that file. Char posns might be easier to handle in the UIs -- I doubt anyone would actually want to call SourceRegion manually (unless they're debugging it).

No idea what might happen if the file contains "wide chars".

2013-03-07 JAA now thinks "char posn" is a *bad idea* because it is not clear how many chars should be counted for a newline -- it looks like 1 char but may count as 2 on some systems (*e.g.* M$ abominations)

#8 - 07 Mar 2013 15:29 - John Abbott

- Assignee set to John Abbott

- Target version set to CoCoA-5.0.3

Why are we allowing so much flexibility with the "source region" command?
It'll never be called by hand!

I have started implementing, and the optional args are a real nuisance!

I now propose the simpler syntax:

```
SOURCE <riga1>,<col1> TO <riga2>,<col2> IN <filename>;
```

with the convention that a column index of 0 means the whole line. JAA expects that riga and col values are actually integer literals (though the first prototype accepts arbitrary expressions).

Perhaps we should also try implementing the Emacs side to see which values it can
easily produce.

#9 - 07 Mar 2013 16:55 - John Abbott

Let me rewaken the question of the command name for reading a (contiguous) region inside a file. Last time there was opposition to having a new name (why? perhaps to avoid having to create a new keyword).

After having completed part (half?) of the implementation, I discovered that it is not solely a question of aesthetic preference. The initial keyword of a command determines its internal type.

If we use **Source** as the keyword for a "source region" command then the internal representation must be that of a Source command. So several functions will have to be modified with an extra check: if (IamSourceRegion) do_this; else do_that; The code for a plain source command is fairly short andd simple; the code for a "source region" command is rather longer.

If we use a different keyword (*e.g.* **SourceRegion**) then the two impls are kept separate; the overall "complexity" of the code will remain much the same. Internally a new command type must be defined to represent the new command.

Opinions?

#10 - 07 Mar 2013 18:41 - John Abbott

- % Done changed from 30 to 50

It compiles and links! The keyword is **SourceRegion**.

There are sure to be some "off by 1" errors (*e.g.* do line numbers start at 0 or 1? Ditto for char positions in a line)

UPDATE I have now added a sanity check for the region limits -- it is in the ctor for FileRegionLineProvider

UPDATE the fn readAndEvaluate should perhaps take as 1st arg a LineProvider instead of a string?

#11 - 07 Mar 2013 23:31 - Anna Maria Bigatti

- % Done changed from 50 to 60

I now have the code for producing

```
SourceRegion 1,0 To 2,0 In "/Users/bigatti/0.99/optimized/CoCoALib-0.99/src/CoCoA-5/tests/anna.cocoa5";
```

even here we have to fix the "off-by-1"s.

2013-03-11 Everything now works, but we still need to discuss "semantics". Does the counting of lines start from 0 or 1? Similarly for character positions in a line. The region is given "numerical delimiters"; are these the posns of the first char in the region, and of the last char in the region? (or perhaps first char after the region?) Does it even matter that much?

#12 - 08 Mar 2013 12:01 - Anna Maria Bigatti

I played a bit with the emacs SourceRegion and I think that it might be useful to "echo" the region that has been sent, so that the users has the history of what has been executed.
I did write the code for emacs so that the sourced lines are printed (commented out), but, of course, the related numer of propts is then printed.

Now, I think the best solution (which would equally work for teh text and the GUI) is that is is **SourceRegion** itself to **output the lines before executing them**.

#13 - 11 Mar 2013 18:51 - John Abbott

- Status changed from In Progress to Feedback

- % Done changed from 60 to 90

Decided the fine details of the semantics; C++ and emacs code are now in agreement.
Modified Lexer.C and the LineProviders so that WHERE info is printed out for errors detected when processing a region sent by SourceRegion.

#14 - 21 May 2013 17:12 - John Abbott

- Status changed from Feedback to Closed

- % Done changed from 90 to 100

Apart from bug #344, no (design) problems have arisen; so I'm closing this issue.