Hadoop Common / HADOOP-1148

re-indent all code

⌄ Details

Type:	⬆ Improvement		Status:	**CLOSED**
Priority:	⬇ Minor		Resolution:	Fixed
Affects Version/s:	None		Fix Version/s:	0.13.0
Component/s:	None			
Labels:	None			

⌄ Description

We should re-indent all code to consistently use 2-spaces per level. This will not invalidate outstanding patches: one can use the '-l' option to ignore whitespace differences in patches.

⌄ Attachments

</>	checkstyle-errors.html	515 kB	19/Apr/07 20:39
▤	indents2.patch	152 kB	17/Apr/07 19:42
▤	whitespace.patch	844 kB	19/Apr/07 20:39

⌄ Activity

↑

◉ **Doug Cutting** created issue - 22/Mar/07 20:19

⌄ ◉ **David Bowen** added a comment - 22/Mar/07 20:44

I agree that consistency in this regard would be good.

I don't want to start an argument over something so unimportant, but I was just wondering - is there a consensus in this developer community that two spaces are better than four?

⌄ ◉ **Konstantin Shvachko** added a comment - 22/Mar/07 20:54

I agree we should re-indent uniformly to 2 spaces.

⌄ ◉ **Doug Cutting** added a comment - 22/Mar/07 21:01

> is there a consensus in this developer community that two spaces are better than four?

To make a change, we'd need a consensus that four are better than two. Two has been the stated goal since the project's inception.

I prefer two because it permits more code per line while staying with 80 columns and using descriptive identifiers. I always edit in 80-column windows. Four spaces per indent level can quickly use nearly half the columns, with nested classes, anonymous methods, try/catch blocks, etc. It's nice when, e.g., 3-parameter method calls, where the method name and each parameter name might have ten characters, can be written on a single line, without wrapping each parameter to a separate line. I've never found two-space indentation to make code any less readable.

⌄ ◉ **Dhruba Borthakur** added a comment - 22/Mar/07 22:45

I like the idea of having 80 column line width. Makes code easily readable. But some portions of existing code have longer line lengths. Does this issue aim to address this issue too?

⌄ ◉ **Doug Cutting** added a comment - 23/Mar/07 00:14

> Does this issue aim to address this issue too?

No. Line-breaking would have a bigger affect on patches. Perhaps we might do that separately, but, for now, I was just thinking of re-indenting.

Arun Murthy added a comment - 23/Mar/07 10:41

As much as I'd love to have two spaces everywhere I'm worried about the adverse impact on 'svn annotate' ... there, that's my best impression of the devil's advocate.

If everyone does agree on doing it anyway, I'd argue this is the right time to fix the line-lengths and also impose stricter rules on coding-conventions and adoption of tools like checkstyle in one shot.

Thomas White added a comment - 23/Mar/07 13:27

+1

My feeling is that we should fix compile warnings (HADOOP-958) before fixing style warnings. However, this one is relatively unobtrusive, since there is a way of not invalidating current patches, so I would be happy for it to go before or in parallel with fixing warnings.

The checkstyle build target will check identation (as well as lots of other things), and I would encourage developers and committers to use it, especially for large chunks of new code. (I should confess at this point that I have yet to start using it in earnest as a part of the commit cycle!)

Doug Cutting added a comment - 23/Mar/07 18:27

> As much as I'd love to have two spaces everywhere I'm worried about the adverse impact on 'svn annotate'

That is indeed the biggest cost that I can see. To find out who last edited a line one would have to use 'svn annotate -r XXX' where XXX is the last revision before the re-indentation. So there is a workaround. Is that acceptable?

> I'd argue this is the right time to fix the line-lengths

That's more intrusive, as it would break outstanding patches. It's also harder to do well automatically.

Doug Cutting added a comment - 23/Mar/07 18:35

> My feeling is that we should fix compile warnings (HADOOP-958) before fixing style warnings.

I don't feel strongly about the ordering. Fixing those compilation warnings is a manual process, no? And it may break existing patches. So it is heavier weight, but I don't see why it must happen first.

In both cases, a patch may be impractical, as it might soon become outdated. These are cases where, once we reach agreement, a committer might simply make and commit the change.

Arun Murthy added a comment - 24/Mar/07 07:46

> That is indeed the biggest cost that I can see. To find out who last edited a line one would have to use 'svn annotate -r XXX' where XXX is the last revision before the re-indentation. So there is a workaround. Is that acceptable?

Fair enough, I guess we have to bite the bullet at some point...

I'd like to propose we put up a reasonable svn 'tag' **before** (and after?) making this change so as to ease finding out the last revision i.e. help ease the process of finding the 'last revision before this change'. Does that sound reasonable? (I'm assuming svn has this feature, having used this in cvs)

Doug Cutting added a comment - 16/Apr/07 21:47

I just re-indented all the .java files, in revision 529410.

Patches can still be applied by using the -l flag, however the patched code will then need to be re-indented before it is committed. It's easy for me to automatically re-indent code with emacs, so if you have a patch that you'd like re-indented, I'd be happy to do that; just ask.

Doug Cutting made changes - 16/Apr/07 21:47

Field	Original Value	New Value
Fix Version/s		0.13.0 [12312348]
Resolution		Fixed [1]
Status	Open [1]	Resolved [5]

⌄ ☺ Raghu Angadi added a comment - 16/Apr/07 22:58

Doug,

Could you attach your indentation settings on your editor (emacs?)?
Does some one have config file for eclipse that sets the same settings?

⌄ ⊙ Doug Cutting added a comment - 16/Apr/07 23:11

(add-hook 'find-file-hooks 'turn-on-auto-fill) ; always use fill
(add-hook 'text-mode-hook 'turn-on-auto-fill) ; always use fill

(defun my-code-mode-hook ()
(setq fill-column 79)
(setq comment-column 50)
(setq c-basic-offset 2)
(setq indent-tabs-mode nil) ; untabified indenting
)

(add-hook 'lisp-mode-hook 'my-code-mode-hook)
(add-hook 'c++-mode-hook 'my-code-mode-hook)
(add-hook 'java-mode-hook 'my-code-mode-hook)

⌄ ☺ Raghu Angadi added a comment - 17/Apr/07 04:24

Thanks. I think default behavior of 2 space indentation in one editor won't be same as 2 space indentation in another editor. Are those differences ok or will all new patches be re-indented by these settings?

⌄ ⊙ Hadoop QA added a comment - 17/Apr/07 11:21

Integrated in Hadoop-Nightly #60 (See http://lucene.zones.apache.org:8080/hudson/job/Hadoop-Nightly/60/)

⌄ ⊙ Doug Cutting added a comment - 17/Apr/07 16:40

> I think default behavior of 2 space indentation in one editor won't be same as 2 space indentation in another editor.

Right. The biggest variation seems to be around parameter indentation, whether a call is indented as: {{
foo(a,
b,
c);
}} as by emacs, or as {{
foo(a,
b,
c);
}}
as by eclipse.

Owen says he knows a way to get eclipse to use the former. I prefer the former, but would certainly not bounce a patch for using the latter. I'm not sure which the style checker prefers, if any.

⌄ ⊙ Doug Cutting added a comment - 17/Apr/07 17:15

Lemme try that again, with *fancy* **wiki** formatting — *enabled*:

... whether a call is indented as:

```
foo(a,
    b,
    c);
```

as by emacs, or as something like

```
foo(a,
  b,
  c);
```

as by eclipse.

🔵 Raghu Angadi added a comment - 17/Apr/07 18:08

True. I like th emacs argument alignment... but could not easily find where to set that in eclipse. I usually manually set it.

Also either in emacs or eclipse, I manually override this alignment to keep the text with in 80 columns. I think in your automated conversion some of these have crossed the 80 columns.

I think in patch submission, the enforcement should be semi-strict and code should not be auto indented.

🔘 Doug Cutting added a comment - 17/Apr/07 18:32

> I manually override this alignment to keep the text with in 80 columns.

A better way to fix that, if no other options are available, is to put the newline before the parenthesis:

```
aVeryVeryLongVariableName.aVeryVeryLongMethodName
   (firstVeryVeryLongParameterName,
    secondVeryVeryLongParameterName,
    thirdVeryVeryLongParameterName);
```

> I think in your automated conversion some of these have crossed the 80 columns.

Yes, but I don't think we should fix 80-column violations wholesale, since that would break outstanding patches. With indentation-only changes, existing patches can still be easily applied.

🔘 Doug Cutting added a comment - 17/Apr/07 19:42

Somehow yesterday I didn't manage to re-indent quite all the files. I'm not sure what happened. I've attached a patch of what would happen if I re-indent things now. Should I go ahead and do this?

Some files are really badly indented, others only indent parameters in the "eclipse" style, described above.

🔘 Doug Cutting made changes - 17/Apr/07 19:42

Attachment	indents2.patch [12355714]

🔘 Thomas White added a comment - 17/Apr/07 20:03

Yes - please go ahead and finish the job. (BTW I've just committed ~~HADOOP-1190~~ unchecked warnings changes for dfs but they don't conflict with your patch.)

🔘 Hadoop QA added a comment - 18/Apr/07 11:04

Integrated in Hadoop-Nightly #61 (See http://lucene.zones.apache.org:8080/hudson/job/Hadoop-Nightly/61/)

🔘 Doug Cutting added a comment - 19/Apr/07 20:39

There were some problems with my earlier Emacs indentation scripting. It missed files that define multiple classes, and sometimes mis-indented methods with annotations.

I also:

- modified checkstyle's case indentation rule to use Sun's conventions;
- disabled checkstyle for JavaCC-generated files
- removed spaces before and after parens
- removed spaces before semicolons
- added spaces after commas

The number of checkstyle warnings are now radically reduced, with very few now that can be fixed by changing spacing alone (so that patches survive).

1115 Line is longer than 80 characters.
474 First sentence should end with a period.
414 'if' construct must use '{}'s.
215 Redundant 'public' modifier.
135 Array brackets at illegal position.
117 Line contains a tab character.
116 Must have at least one statement.

101 ',' is not followed by whitespace.

56 '}' should be on the same line.

40 '{' should be on the previous line.

38 ';' is not followed by whitespace.

34 'else' construct must use '{}'s.

33 'private' modifier out of order with the JLS suggestions.

28 '(' should be on the previous line.

27 'static' modifier out of order with the JLS suggestions.

24 Utility classes should not have a public or default constructor.

23 '{' is followed by whitespace.

22 'name' hides a field.

21 'for' construct must use '{}'s.

20 Inner assignments should be avoided.

Should I commit this?

Doug Cutting made changes - 19/Apr/07 20:39

| Attachment | checkstyle-errors.html [12355864] |
| Attachment | whitespace.patch [12355863] |

Thomas White added a comment - 19/Apr/07 20:50

+1

Owen O'Malley added a comment - 19/Apr/07 21:33

+1

Hadoop QA added a comment - 20/Apr/07 11:33

Integrated in Hadoop-Nightly #64 (See http://lucene.zones.apache.org:8080/hudson/job/Hadoop-Nightly/64/)

Raghu Angadi added a comment - 24/Apr/07 17:45

I am merging a largish patch made on 2-3 weeks old trunk. I think these changes are resulting in some rejections even with `patch
-l`. For e.g. I could not find any other reason for following rejection for FSDataset.java.

```
Index: src/java/org/apache/hadoop/dfs/FSDataset.java
===================================================================
--- src/java/org/apache/hadoop/dfs/FSDataset.java        (revision 527216)
+++ src/java/org/apache/hadoop/dfs/FSDataset.java        (working copy)

FSDataset.java.rej:

***************
*** 471,488 ****
      /**
       * Get a stream of data from the indicated block.
       */
-     public synchronized InputStream getBlockData(Block b) throws IOException {
          if (! isValidBlock(b)) {
              throw new IOException("Block " + b + " is not valid.");
          }
          // File should be opened with the lock.
-         return new FileInputStream(getFile(b));
      }

      /**
       * Start writing to a block file
       */
```

Doug Cutting added a comment - 24/Apr/07 18:06

Did you try changing patch's fuzz factor?

Another thing that sometimes works is to revert your workspace to a version where this patch applies, apply the patch, then use 'svn up' to sync with trunk. Subversion's internal patch system is generally better than patch.

⚙ Doug Cutting made changes - 08/Jun/07 20:40

Status	Resolved [5]	Closed [6]

Transition	Time In Source Status	Execution Times
⚙ Doug Cutting made transition - 16/Apr/07 21:47 `OPEN` ➡ `RESOLVED`	25d 1h 27m	1
⚙ Doug Cutting made transition - 08/Jun/07 20:40 `RESOLVED` ➡ `CLOSED`	52d 22h 52m	1

⌄ People

Assignee:

◯ Doug Cutting

Reporter:

◯ Doug Cutting

Votes:

0 Vote for this issue

Watchers:

0 Start watching this issue

⌄ Dates

Created:

22/Mar/07 20:19

Updated:

08/Jun/07 20:40

Resolved:

16/Apr/07 21:47