Indentation

What is indentation?

*Indentation is used to format program source code in order to improve its readability* (wikipedia). See wikipedia article. There is nice Unix utility called `indent`, which reformats C code (C++ is not yet supported).

See effect of indentation of example code by `indent` command.

<table>
<thead>
<tr>
<th>Raw code</th>
<th>Indented code</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>int main ()</code></td>
<td><code>int main ()</code></td>
</tr>
<tr>
<td>`{</td>
<td>`    {</td>
</tr>
<tr>
<td>int i,x;</td>
<td>int i, x;</td>
</tr>
<tr>
<td>for( i =0 ; i&lt; 137; i++) {</td>
<td>for (i = 0; i &lt; 137; i++)</td>
</tr>
<tr>
<td>x += i*i; }</td>
<td>{</td>
</tr>
<tr>
<td>return 0; }</td>
<td>x += i * i;</td>
</tr>
<tr>
<td>}</td>
<td>return 0; }</td>
</tr>
</tbody>
</table>

This tool works well with C code. It might corrupt C++ code. Example:

Unindented code:

```cpp
void RPTrackCandidateCollectionFitter::beginJob (const edm::EventSetup &)
{
  if (verbosity_)
  {
    edm::LogInfo ("RPTrackCandidateCollectionFitter") << 
      "[RPTrackCandidateCollectionFitter::beginJob]";
  }
}
```

The same C++ indented with `indent` tool:

```cpp
void RPTrackCandidateCollectionFitter::beginJob (const edm::EventSetup &)
{
  if (verbosity_)
  {
    edm::LogInfo ("RPTrackCandidateCollectionFitter") << 
      "[RPTrackCandidateCollectionFitter::beginJob]";
  }
}
```

`edm::LogInfo` was splitet between two lines.

Which indentation shall we use?

CMSSW is written (mainly) using GNU style. That means:

- 2 spaces for one indentation level
- brackets in new line after definition

Popular editors and Coding Standards:
**Indentation tool for CMSSW config files and C++**

There is no standard tool for indentation of C++ code, but we can profit from the fact that some text editors have built in support for indentation. We will use indentation function from emacs editor.

This tool does not move brackets from one line to another. It will just adjust number of spaces/tabs at begging of line.

You can find this script on AFS:

- /afs/cern.ch/exp/totem/soft/cmssw/tools/indentation/indent.sh

**Sample usage of indentation script for C++ code:**

<table>
<thead>
<tr>
<th>Raw code</th>
<th>Indented code</th>
</tr>
</thead>
<tbody>
<tr>
<td>int main ()</td>
<td>int main ()</td>
</tr>
<tr>
<td>{</td>
<td>{</td>
</tr>
<tr>
<td>int i,x;</td>
<td>int i,x;</td>
</tr>
<tr>
<td>for( i =0 ; i&lt; 137; i++){ for( i =0 ; i&lt; 137; i++){</td>
<td></td>
</tr>
<tr>
<td>// C++ comment</td>
<td>// C++ comment</td>
</tr>
<tr>
<td>/* C comment */</td>
<td>/* C comment */</td>
</tr>
<tr>
<td>x += i*i; }</td>
<td>x += i*i; }</td>
</tr>
<tr>
<td>return 0;</td>
<td>return 0;</td>
</tr>
<tr>
<td>}</td>
<td>}</td>
</tr>
</tbody>
</table>

Command used for indentation:

```
(...)
Indenting region...
Indenting region... done
Wrote /home/grzanka/uglyCppCode.cc
Done
```

**Indented code and SVN**

Let us take code which is already correctly indented. When we call indent tools (indent command and emacs) for second time, they will update "last modification" timestamp. It is fine for us, but SVN client will treat such files and "modified" and will try to increase version number.

Be careful when using indentation tool. indent.sh script is checking if indentation will change file. If file will be not change, then it will preserve timestamp.

-- LeszekGrzanka - 19 Aug 2008