GammaLib - Bug #828

GLog prepends date in block of text

04/10/2013 03:38 PM - Knödlseder Jürgen

Status:	Feedback	**Start date:**	04/10/2013
Priority:	Low	**Due date:**	
Assigned To:		**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

In the following output

```
2013-04-10T09:46:21:  Initial iteration: func=9.27457e+06, Lambda=0.001
2013-04-10T09:46:25:  Iteration 1: func=9.02055e+06, Lambda=0.0001, delta=254025, max(grad)=1.39084e+06 [2]
2013-04-10T09:46:25:    Parameter "Prefactor" drives optimization step (step=0.944368)
2013-04-10T09:46:25:    Parameter "2013-04-10T09:46:25:  Prefactor" hits minimum: 1e-07 < 1e-07 (1)
2013-04-10T09:46:30:  Iteration 2: func=8.82274e+06, Lambda=1e-05, delta=197809, max(grad)=1.24019e+06 [2]
2013-04-10T09:46:30:    Parameter "Prefactor" does not drive optimization step anymore.
2013-04-10T09:46:30:    Parameter "Index" drives optimization step (step=8.90739e-06)
2013-04-10T09:46:30:    Parameter "Prefactor" hits minimum: -7.4137e-07 < 1e-07 (2)
```

the GLog method prepends a date before the Prefactor text. This does not happen systematically.

Here the code where it happens in GOptimizerLM::step_size():

```
        // Signal if a parameter is driving the optimization step
        if (ipar_bnd != -1) {
            m_hit_boundary[ipar_bnd] = true;
            if (m_logger != NULL) {
                *m_logger << "  Parameter \"";
                *m_logger << pars.par(ipar_bnd).name();
                *m_logger << "\" drives optimization step (step=";
                *m_logger << step << ")" << std::endl;
            }
        }
```

which calls basically the GLog::append() method:

```
void GLog::append(std::string arg)
{
    // If the buffer is empty or if the last charater is a \n, prepend a
    // prefix at the beginning of the string to be inserted.
    if (m_buffer.size() == 0 || m_buffer[m_buffer.size()-1] == '\n') {

        // Prepend prefix
        arg.insert(0, prefix());
    }

    // Search the first CR (\n)
    std::size_t pos = arg.find_first_of("\n",0);

    // Search all \n characters. Ignore the last CR.
    while (pos != std::string::npos && pos < arg.size()-1) {

        // Prepend prefix
        std::string pre = prefix();
        arg.insert(pos+1, pre);
```

```
      // Search next CR
      pos = arg.find_first_of("\n",pos+1+pre.size());

    } // endwhile

    // Add string to buffer
    m_buffer.append(arg);

    // Flush Buffer
    flush();

    // Return
    return;
}
```

The problem comes maybe from the first if statement, which prepends the date each time that the buffer is empty. The buffer may however be empty in the middle of a line, which does not require to prepend a date. This occurs if the buffer flush occurs when the buffer is full.

History

#1 - 01/22/2014 10:04 PM - Knödlseder Jürgen

- Status changed from New to Feedback

- % Done changed from 0 to 100

I think this has been fixed in release 00-08-00, but still should be verified.