WSDL2RPG – FAQ

FAQ How to Troubleshoot the "Not all data was written" Error Message

Status of this Document

Date: 02.02.2012
Version: 1.2
Credits: John Fox

Question

The web service stub crashes with an "Not all data was written" error message?

Answer

The "16: recvchunk: saveproc: Not all data was written!" error message is sent by the HTTP API when a user written save procedure failed to save the data that was just received from the server.

The WSDL2RPG web service stubs use their own special save procedure to analyze the data received from the server. Depending on the format of that data, the procedure either forwards the data to the MIME decoder, or directly to the eXpat parser. This is done by procedure `HTTP_receiveResponse()`.

Therefore the "Not all data was written!" error message is just the final message that you see. But it does not explain what went wrong. For example the core problem might be one of these possible errors, or something completely different:

- Range of subscript value or character string error.
- Division by zero
- Value too large
- Invalid numeric number format
- Invalid xml document received from server
- Unexpected data received from server
- Unexpected Soap element received from the web service
- etc.

General Troubleshooting

In order to determine the actual problem you should review the job log. If there is nothing that points to the problem, you should enable the HTTP API debug log, recompile your program and call the web service again:

```
WebServicePortName_Port_setHttpDebug(
      *ON: 'PathToYourDebugLogFile.log')
```

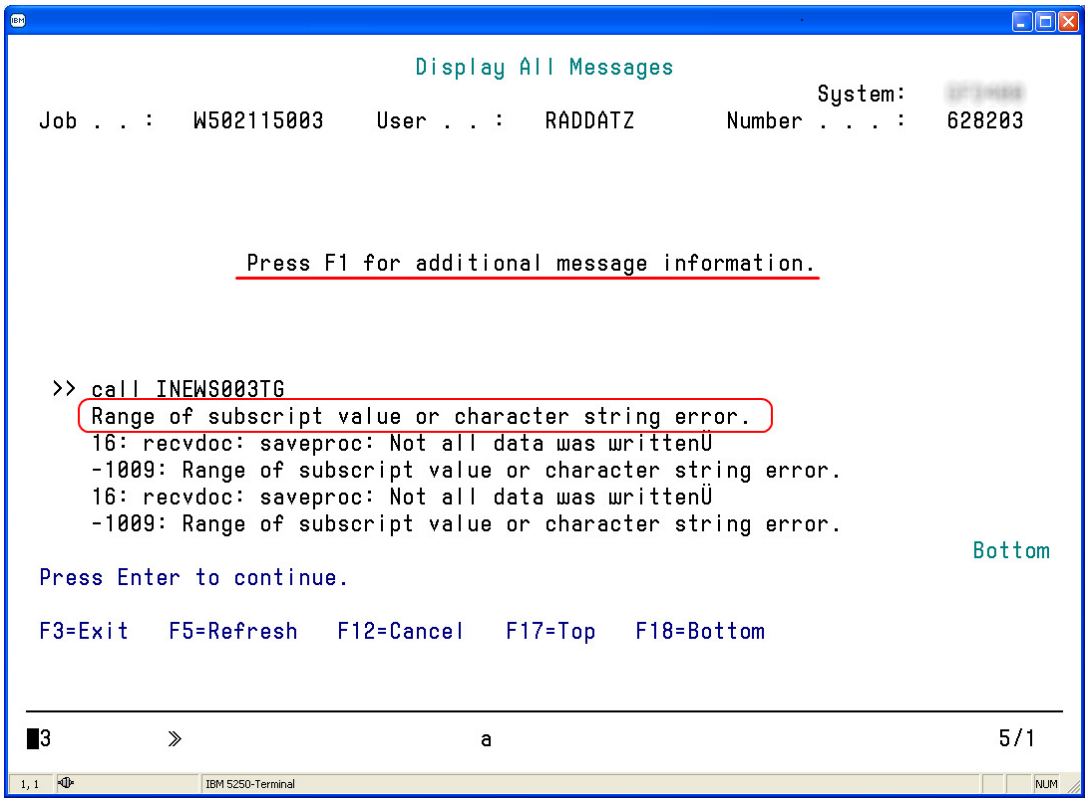
Hopefully, the log will reveal the cause of the problem.

Sample 1: MCH0603 - Range of subscript value or character string error

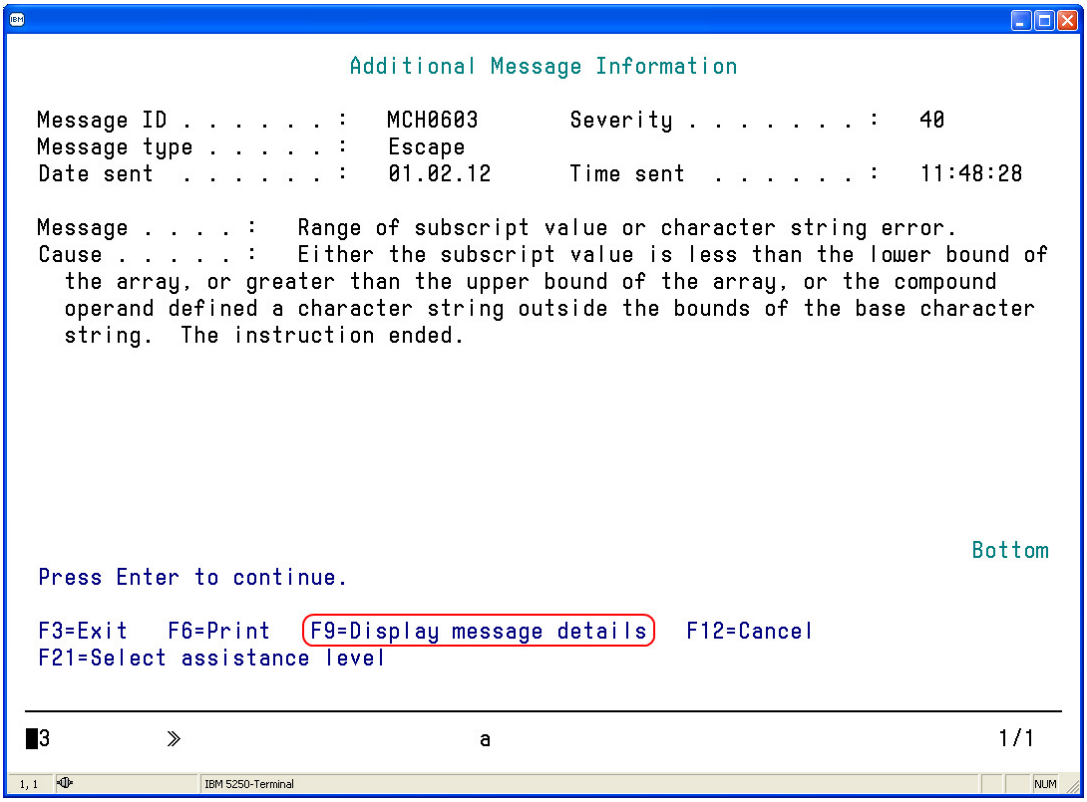
The first thing you should do, after having noticed the “Not all data was written” error message, is to have a close look at the job log:

```
DSPJOBLOG
```

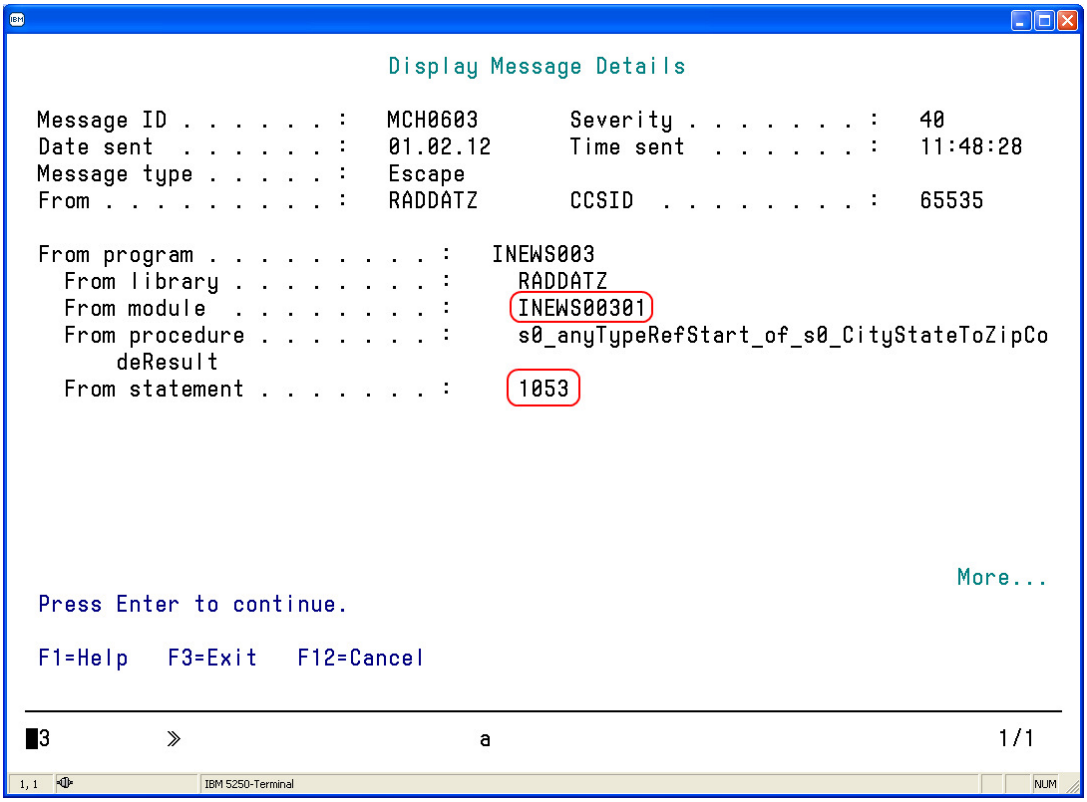
Then press F10 for “Display detailed messages” and F18 to get to the bottom of the list of messages. Now you should see something similar to this:



Now that you have identified the actual error, move your cursor to the error message and press F1 to get additional information for the message:



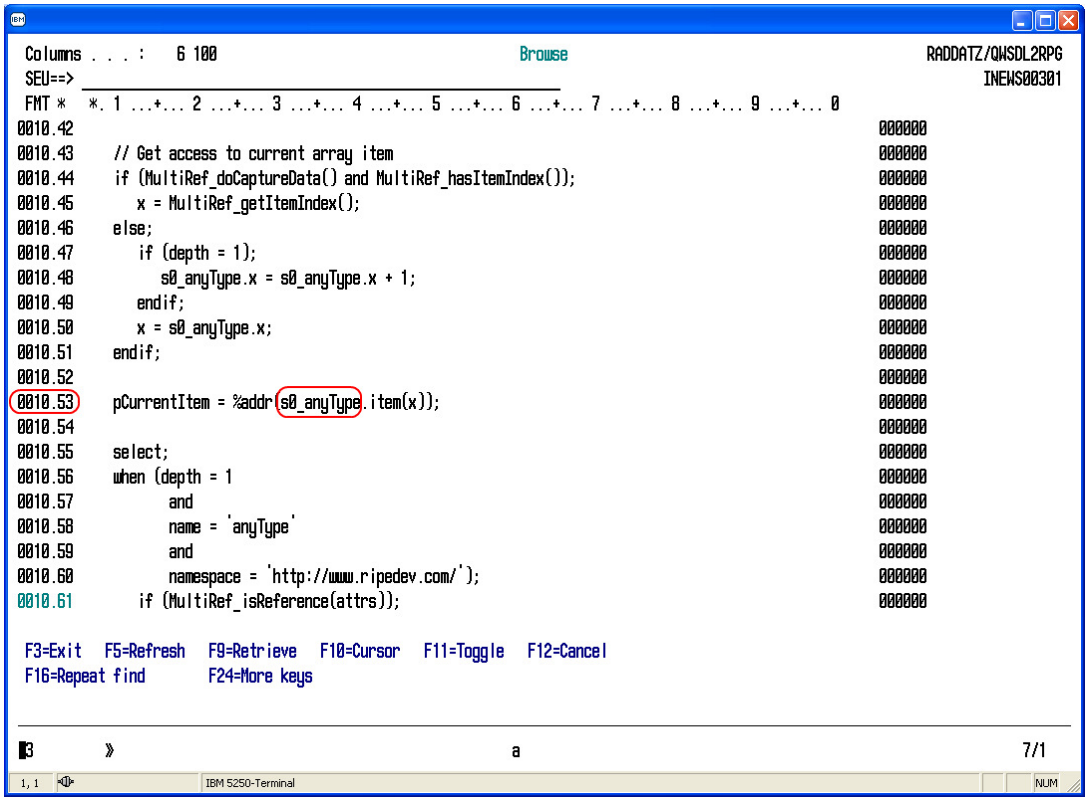
Since there is nothing interesting here you can go on with F9 to get the message details:



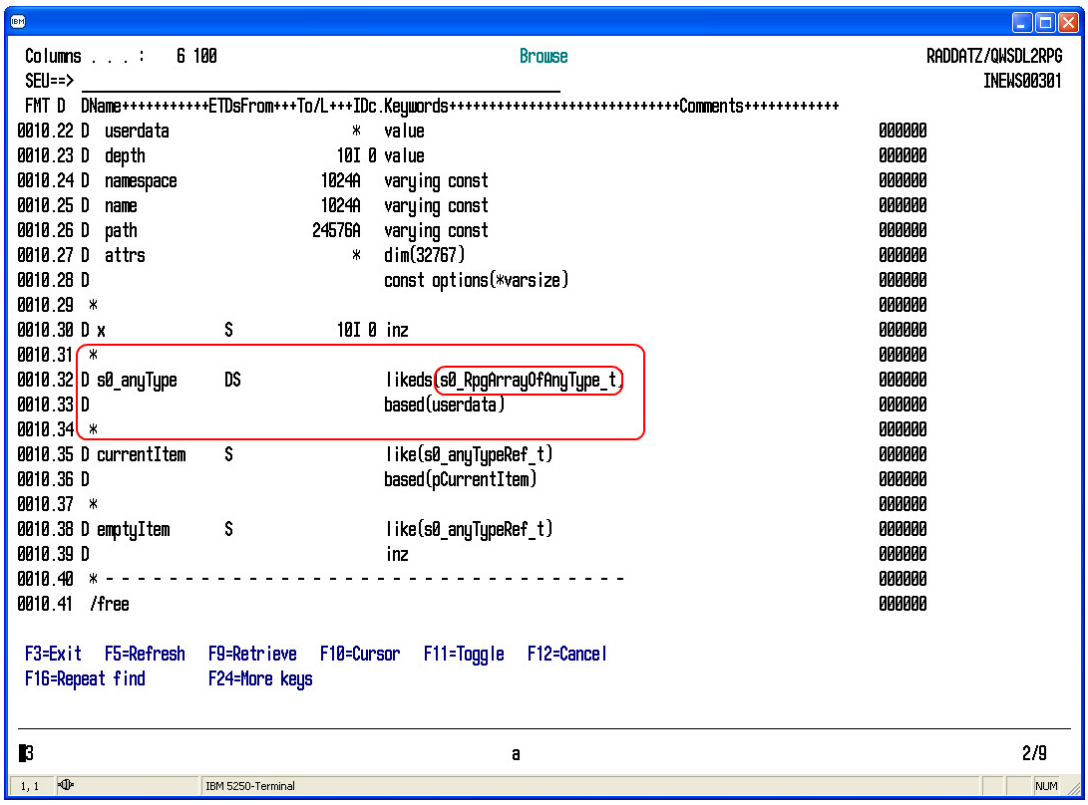
Here you get the next pieces of the puzzle:

- the name of the module the message came from
- the RPG statement number the message was sent from

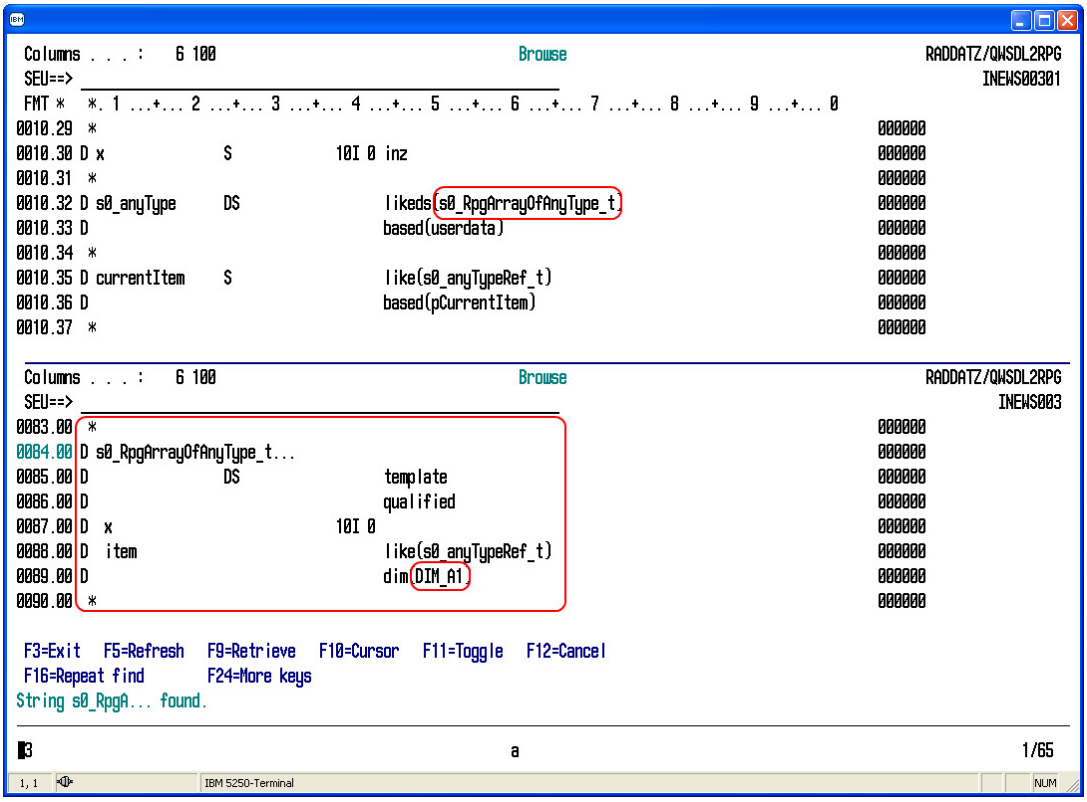
Having this information, please open the source member and go to the specified line number:



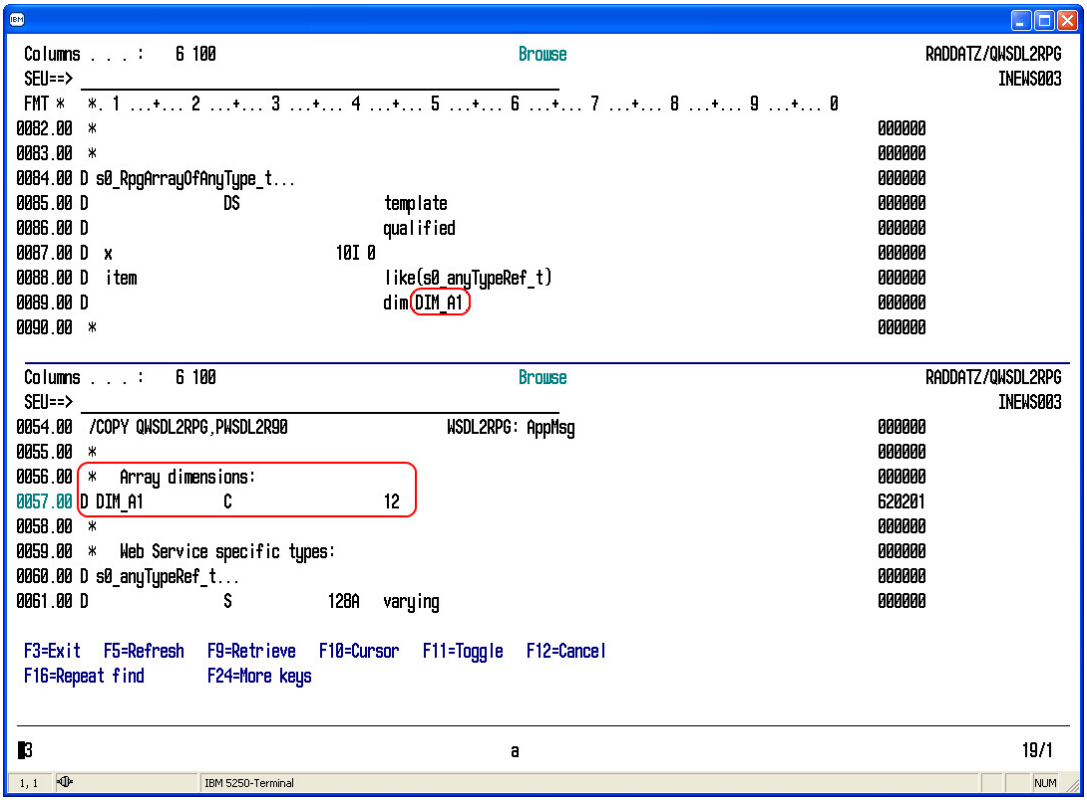
Now it is clear that "**`s0_anyType.item`**" is an array with too less elements. In other words: The web service sent more elements that the stub module could place into that array. Hence you have to slightly increase the number of elements to a more reasonable value. But first you have to find the reference field. Therefore go to the top of the procedure watching for the statement where "**`s0_anyType`**" is defined:



Since all reference fields are defined in the stub module, you have to open it to get to the reference field:



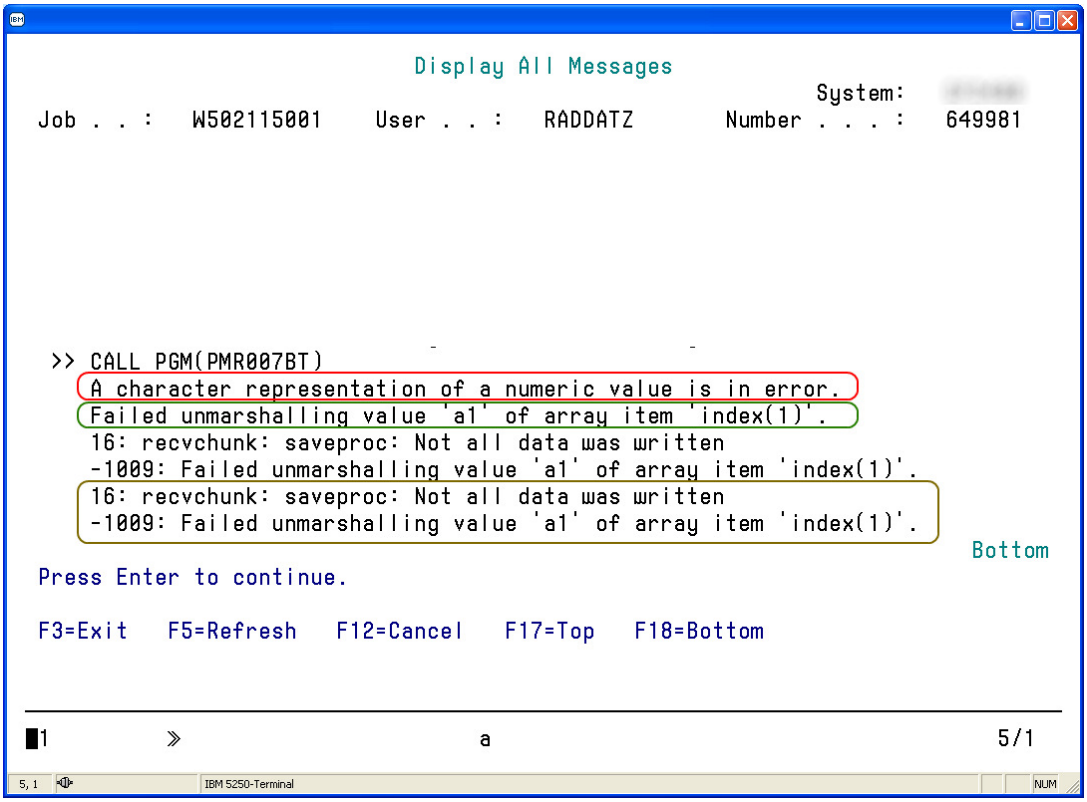
Voila, you got it! Constant DIM_A1 specifies the number of items of the "**`s0_anyType.item`**". The last step to do is to change that constant to match your actual needs:



Sample 2: RNX0105 – A character representation of a numeric value is in error

Beside the job log you should also look at the HTTPAPI debug log. Starting with v1.15 WSDL2RPG places additional debug information into that log for some errors.

For a RNX0105 error message the job log may look like this:



The messages enclosed with a **brown frame** are what you get back to the command line, if you used the generated test program to call the web service.

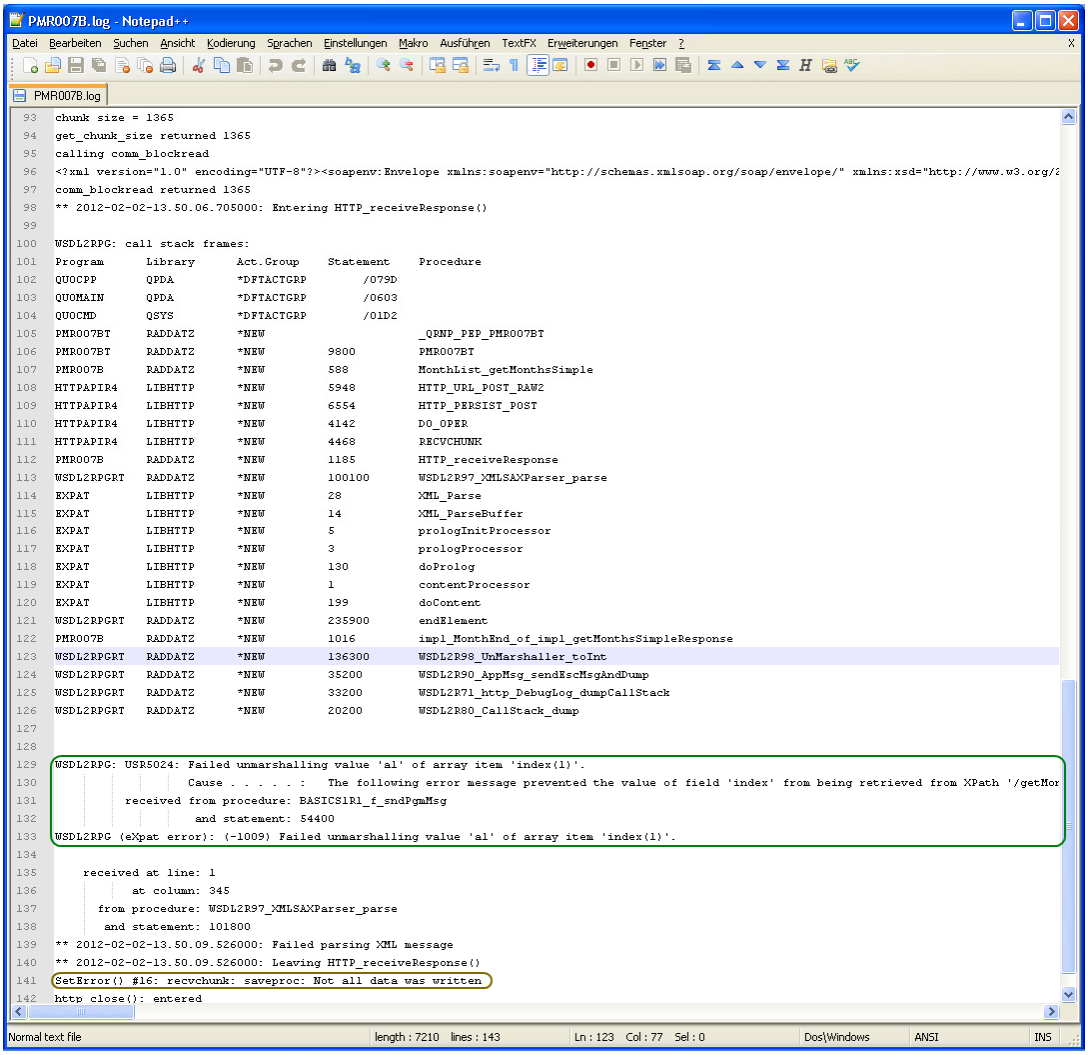
The message with the **green frame** is the message which was captured by WSDL2RPG and appended to the HTTPAPI debug log.

The message with the **red frame** is the original error message that was send by the RPG runtime.

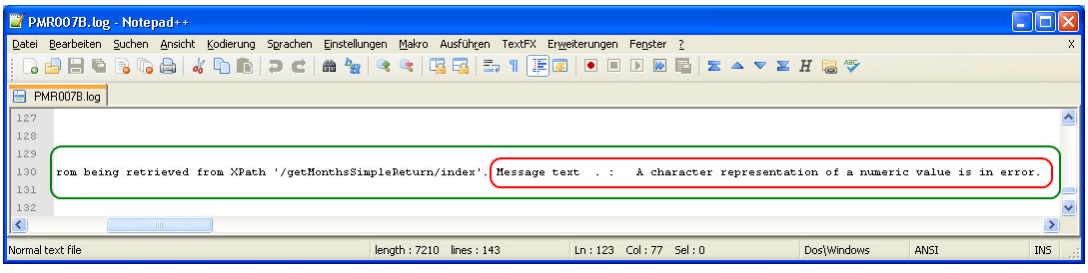
Hopefully you had enabled the debug log before you called the web service. If you did not do that, please enable the debug log and call the web service again:

```
MonthList_Port_setHttpDebug(
   *ON: '/home/raddatz/wsdl2rpg/PMR007B.log');
```

With the debug log enabled you can take advantage from the additional debug information that has been appended to the HTTPAPI debug log as shown below:



Do you miss the original error message that was sent by the RPG runtime? Well, it is there. Just scroll to the right to see it:



Your comments are important to me! Please send me your comments about this FAQ. I will greatly appreciate it.

thomas.raddatz@tools400.de