

XML File

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<System>
```

```
  <Vector name="FixData">
```

```
    <Element name="Time" />
```

```
    <Element name="Latitude" />
```

```
    <Element name="Longitude" />
```

```
    <Element name="Quality" />
```

```
    <Element name="NumberSatellites" />
```

```
    <Element name="HDOP" />
```

```
    <Element name="Altitude" />
```

```
    <Element name="GeoidHeight" />
```

```
  </Vector>
```

```
  <Module name="input" type="ParseGPSMessages">
```

```
    <InputFile value="gps_data.txt" />
```

```
    <OutputVector value="FixData" />
```

```
  </Module>
```

```
  <Module name="logger" type="LogData">
```

```
    <InputVector value="FixData" />
```

```
    <OutputFile value="fix_data_output.txt" />
```

```
  </Module>
```

```
</System>
```

Class Definition

```
#include "Module.h"
#include <string>
#include <iostream>

class System;
class ModuleData;
class GlobalVector;

class ParseGPSMessages : public Module
{
public:
    //! Standard module constructor
    ParseGPSMessages (System* system, std::string module_name, ModuleData* md);
    ~ParseGPSMessages ();

    void initialize();
    bool run();
    void terminate();

private:
    void closeInput();                //!< Closes the input file

    GlobalVector* output;             //!< Output vector into which to place the data
    std::ifstream input;              //!< File from which to read the data

    int idx_time;
    int idx_latitude;
    int idx_longitude;

    int idx_quality;
    int idx_num_sats;
    int idx_hdop;
    int idx_alt;
    int idx_geoid;
};
```

Initialization

```
ParseGPSMessages::ParseGPSMessages(System* system,
                                     std::string module_name,
                                     ModuleData* md) : Module(system)
{
    string input_file_name = md->getParameter("InputFile",
                                              "gps_data.txt");
    input.open(input_file_name.c_str());

    string output_vec = md->getParameter("OutputVector",
                                         "GPSData");

    output = system->findVector(output_vec);

    if (output == 0)
    {
        cerr << "Error in configuring ParseGPSMessages, "
              << "couldn't find global vector '" << output_vec << "'" << endl;
        throw "Unable to find global vector for ParseGPSMessages";
    }

    idx_time = output->findElement("Time");
    idx_latitude = output->findElement("Latitude");
    idx_longitude = output->findElement("Longitude");

    idx_quality = output->findElement("Quality");
    idx_num_sats = output->findElement("NumberSatellites");
    idx_hdop = output->findElement("HDOP");
    idx_alt = output->findElement("Altitude");
    idx_geoid = output->findElement("GeoidHeight");
}
```

Run

```
bool
ParseGPSMessages::run()
{
    if (input.good())
    {
        string line;
        getline(input, line);

        double time = 1;
        double lat = 2;
        double lon = 3;
        double quality = 4;
        double num_sats = 5;
        double hdop = 6;
        double alt = 7;
        double geoid = 8;

        // Parse the line into fields

        (*output)[idx_time] = time;
        (*output)[idx_latitude] = lat;
        (*output)[idx_longitude] = lon;
        (*output)[idx_quality] = quality;
        (*output)[idx_num_sats] = num_sats;
        (*output)[idx_hdop] = hdop;
        (*output)[idx_alt] = alt;
        (*output)[idx_geoid] = geoid;

        return false;
    }
    else
    {
        return true;
    }
}
```