

Neuroscan Support in BCI2000

Gerwin Schalk

© 2004 Wadsworth Center, New York State Department of Health

supported by work from Jürgen Mellinger

Eberhard Karls University of Tübingen, Germany

February 3, 2006

Contents

1	Introduction	2
2	Neuroscan Source Module	2
3	The neurogetparams Command Line Tool	2
4	Undocumented Communication	3
4.1	Request Basic Information	3
4.2	Receive 16-bit Data Packet	4

1 Introduction

NuAmps/SynAmps are EEG recording systems from Neuroscan, Inc., that are widely used in clinical settings. The increasing use of BCI2000 in those environments created the need for Neuroscan support in BCI2000. This document describes this support that consists of two components: A BCI2000-compatible Source Module (**Neuroscan.exe**) and a command-line tool (**neurogetparams**). These components are described below and have been tested with *Neuroscan Acquire* version 4.3.1.

2 Neuroscan Source Module

The BCI2000-compatible Source Module **Neuroscan.exe** can be used instead of any other source module. In addition to standard parameters (i.e., *SampleBlockSize*, *SamplingRate*, *SoftwareCh*), it only requests one Neuroscan-specific parameter (*ServerAddress*) that defines the IP address (or host name) and the port number of the *Acquire* server. An example for an appropriate definition is **localhost:3999**. Before starting BCI2000, you need to start *Acquire*, click on the *S* symbol in the top right corner (to enable the server) and start the server on any port. Please note that by default, *Acquire* suggests port 4000, which is a port used by BCI2000. You might use port 3999 instead. Once the server is running, you can start BCI2000. It is imperative that certain parameters (e.g., the number of channels) in BCI2000 match the settings in *Acquire*. You can read these settings from *Acquire* using the command line tool described in the following section. This command line tool can create a parameter file fragment that needs to be loaded on top of an existing parameter file every time a setting in *Acquire* is changed.

3 The neurogetparams Command Line Tool

This command line tool reads system settings from the *Neuroscan Acquire* server, displays them on a screen and creates a BCI2000 parameter file fragment if desired. It can be used as follows: **neurogetparams -address localhost:3999 -paramfile test.prm**. (The *Acquire* server has to be enabled before using this tool.) Once BCI2000 is configured correctly, this parameter file fragment needs to be loaded on top of the existing configuration to make sure that the settings match. You only need to repeat this

procedure if you change settings in *Acquire* (e.g., such as the number of channels or the amplification).

BCI2000 Parameter Tool for Neuroscan Acquire V4.3

(C)2004 Gerwin Schalk and Juergen Mellinger

Wadsworth Center, New York State Dept of Health, Albany, NY

Eberhardt-Karls University of Tuebingen, Germany

Signal Channels: 32

Event Channels: 1

Block Size: 40

Sampling Rate: 1000

Bits/Sample: 16

Resolution: 0.168uV/LSB

Parameter file test.prm successfully written

4 Undocumented Communication

Neuroscan Acquire supports communication options that are not documented in the `NetAcquire.doc` document. BCI2000 needs to use two of them, the request for basic Neuroscan configuration information (which will return information such as the number of channels and the sampling rate) and the receipt of packets of data, to function properly. They are listed below:

Packet ID	Code	Request	Body Size
"CTRL"	3 (Client Ctrl Code)	5 (Req Basic Info)	0
"DATA"	1 (Datatype Info)	3 (InfoType BasicInfo)	bodyLen
"DATA"	2 (EEG Data)	1 (Raw 16-bit)	bodyLen

4.1 Request Basic Information

The client sends a request for basic information ("CTRL", Client Ctrl Code, Req Basic Info). The server responds by sending a data block ("DATA", Datatype Info, InfoType BasicInfo) that contains a data structure describing the current configuration of data acquisition in *Acquire*. This data structure is as follows:

```

struct AcqBasicInfo
{
    DWORD dwSize;          // Size of structure, used for version control
    int   nEegChan;        // Number of EEG channels
    int   nEvtChan;        // Number of event channels
    int   nBlockPnts;      // Samples in block
    int   nRate;           // Sampling rate (in Hz)
    int   nDataSize;       // 2 for "short", 4 for "int" type of data
    float fResolution;     // Resolution for LSB
};

```

The first element, `dwSize`, can and should be cross checked with the length of the structure `AcqBasicInfo` and must match.

4.2 Receive 16-bit Data Packet

The client may send a *Request to Start Sending Data* to the Acquire server (as described in the `NetAcquire.doc` document). This will prompt the server to start sending blocks of raw signal values. In order to interpret these signal values, the client must first have sent a *Request for Basic Information*, as described above. The incoming data packets ("DATA", EEG Data, Raw 16-bit) can then be interpreted using this information and the code below.

```

// make sure the body has the right length
assert((int)(pMsg->m_dwSize) ==
        (int)(m_BasicInfo.nEegChan+m_BasicInfo.nEvtChan)
        *m_BasicInfo.nBlockPnts*m_BasicInfo.nDataSize);
// process raw 16 bit data
for (int samp=0; samp<m_BasicInfo.nBlockPnts; samp++)
{
    for (int ch=0; ch<m_BasicInfo.nEegChan+m_BasicInfo.nEvtChan; ch++)
    {
        short *sample=(short *)pMsg->m_pBody;
        printf("%d ", sample[samp*(m_BasicInfo.nEegChan+m_BasicInfo.nEvtChan)+ch]);
    }
}

```