

# ThinkGear Connector Development Guide

---

December 18, 2009



The NeuroSky product families consist of hardware and software components for simple integration of this bio-sensor technology into consumer and industrial end-applications. All products are designed and manufactured to meet exacting consumer specifications for quality, pricing, and feature sets. NeuroSky sets itself apart by providing building-block component solutions that offer friendly synergies with related and complementary technological solutions.

Reproduction in any manner whatsoever without the written permission of NeuroSky Inc. is strictly forbidden. Trademarks used in this text: eSense™, ThinkGear™, MDT™, NeuroBoy™ and NeuroSky™ are trademarks of NeuroSky Inc.

**NO WARRANTIES: THE DOCUMENTATION PROVIDED IS "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT OF INTELLECTUAL PROPERTY, INCLUDING PATENTS, COPYRIGHTS OR OTHERWISE, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT SHALL NEUROSKY OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, COST OF REPLACEMENT GOODS OR LOSS OF OR DAMAGE TO INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE DOCUMENTATION PROVIDED, EVEN IF NEUROSKY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. , SOME OF THE ABOVE LIMITATIONS MAY NOT APPLY TO YOU BECAUSE SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES.**

# Contents

<b>Introduction</b>	<b>4</b>
<b>Getting Started</b>	<b>5</b>
<b>Connecting to TGC</b>	<b>6</b>
<b>Reading Data from TGC</b>	<b>7</b>
<b>Miscellany</b>	<b>8</b>
Policy Files . . . . .	8

# Introduction

---

The ThinkGear Connector (TGC) is an executable that provides a daemon-like service that manages communications with ThinkGear devices, such as the MindSet, that are connected to the computer. The TGC runs continuously in the background, and keeps an open socket on the local user's computer, allowing applications to connect to it and receive information from the connected ThinkGear devices. This means that any application in any language that can open and read from sockets (such as Flash's ActionScript3, and other scripting languages) can connect to and receive data from MindSet headsets.

The TGC is provided as an executable for Microsoft Windows, with Mac OS X support planned in a future release.

**Important:** TGC does *not yet completely implement the ThinkGear Socket Protocol* specification, specifically the authorization components. Any authorization commands sent to TGC will simply be **ignored**; applications designed for TGC should not expect any response from the authorization commands.

# Getting Started

---

The first step is to follow the standard end-user instructions for installing and running the TGC as described in the [ThinkGear Connector User's Guide](#). After following the instructions there, return here for more information about developing.

# Connecting to TGC

---

Once the TGC is up and running on the system, the next step is to open a socket connection to it. Refer to the socket API documentation for your application's platform/programming language for details on how to open a socket connection.

Open a socket connection to the TGC using the following configuration:

<b>Host address</b>	127.0.0.1
<b>Port</b>	13854
<b>Protocol</b>	TCP

## Reading Data from TGC

---

Once your application has opened a socket connection to the TGC, it should be able to read MindSet data from the socket connection. In order to parse and understand the data stream, please refer to the [ThinkGear Socket Protocol](#) document. Once your application is properly parsing the data stream according to the ThinkGear Socket Protocol, it has access to all the data coming from the ThinkGear headset! Congratulations, your application can now respond to brainwave data and is BCI-enabled!

# Miscellany

---

## Policy Files

When SWF files open a socket connection, SWF will typically automatically request a `crossdomain.xml` file from the TGC by sending the following XML to the TGC:

```
<policy-file-request/>
```

In response, TGC will automatically write the following XML to the socket to complete the handshake:

```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM "http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
  <allow-access-from domain="*" to-ports="*" />
</cross-domain-policy>
```

This all usually occurs automatically when the socket is opened in SWF, and requires nothing special on the SWF programmer's part. The SWF program can then communicate over the socket (read and write) as normal.

If no policy file is requested by the client, then no policy file will be sent by TGC; TGC will still function as normal in this case.

By default, the TGC allows *any* remotely-running SWF file to access it, though it will only listen on a specific address and port. For details on Flash's policy file implementation, refer to [this link](#).