

## **Plexon Online Client Software Development Kit for Matlab**

This package allows Matlab scripts to read online spike, continuous, and digital event data from Plexon OmniPlex and MAP systems. The package contains the files `mexPlexOnline.mexw32` (for use with 32 bit versions of Matlab) and `mexPlexOnline.mexw64` (for use with 64 bit versions of Matlab), plus Matlab subroutine definition `.m` files and sample scripts. Note that an older version of the SDK, `mexPlexOnline.dll`, is no longer supported and cannot be used with recent versions of Matlab.

`mexPlexOnline.mexw64` (or `mexw32`, as appropriate), the file `PlexClient.dll`, and the `.m` files need to be installed in a path known to Matlab. Your Matlab programs call the functions which are defined in the provided `.m` files, which then in turn call `mexPlexOnline` to interface to the online data stream from an OmniPlex or MAP system.

Note that new releases of OmniPlex may include updated versions of `PlexClient.dll`, but the OmniPlex installer can only update the version that is in the OmniPlex application folder. In order to insure that the versions of `PlexClient.dll` used by OmniPlex and by online Matlab client programs are the same, download the latest version of the Matlab SDK, or after updating OmniPlex, manually copy the latest `PlexClient.dll` from the OmniPlex application folder to the folder containing `mexPlexOnline`.

### **Basic API**

**PL\_InitClient** - you must call this function before calling any other `PL_*` function

**PL\_GetPars** - retrieves the acquisition parameters of the Plexon Server

**PL\_GetTS** - gets the timestamps

**PL\_GetWF** - gets the waveforms

**PL\_GetWFEvs** - gets the waveforms and external events

**PL\_GetAD** - gets the continuous A/D data (single-rate continuous channels only); sample values are in raw A/D units (typically in the range -2048 to +2048)

**PL\_GetADEx** – gets the continuous A/D data for both “slow” (LFP) and “fast” (spike or wideband) channels; sample values are in raw A/D units

**PL\_GetADV** - gets the continuous A/D data (single-rate continuous channels only); sample values are in volts

**PL\_GetADVEx** – gets the continuous A/D data for both “slow” (LFP) and “fast” (spike or wideband) channels; sample values are in volts. This is the recommended function for reading continuous data from OmniPlex systems. Note that the numbering of continuous channels is the same as the “PLX Chan” numbering shown in the Properties Spreadsheet in PlexControl.

**PL\_GetGains** – gets the total per-channel gains for spike channels

**PL\_GetADGains** – gets the total per-channel gains for continuous A/D channels

**PL\_GetNumUnits** - gets the number of defined units for a channel

**PL\_WaitForServer** - wait until the server signals that it has new data. This minimizes the latency of reading new data from the server, as opposed to polling (waiting) for a fixed interval.

**PL\_SendUserEvent** - injects an event into the data stream.

**PL\_Close** - closes the connection to the Plexon Server

**PL\_VTInterpret** - takes matrix returned from PL\_GetTS or PL\_GetWFEvs and produces array of coordinates from CinePlex / Video Tracker data.

See the .m files (both the PL\_\*.m and test\*.m scripts) for usage and additional notes.

Before using any of the PL\_XXX functions you must call PL\_InitClient only **once** when your client starts acquiring data from the Plexon Server, and use the value returned by PL\_InitClient in subsequent PL\_\* calls, e.g.

```
s = PL_InitClient(0);  
pars = PL_GetPars(s);  
[num, ts] = PL_GetTS(s);  
....
```

Likewise, you must call PL\_Close(s) only **once** when your client shuts down, to close the connection with the Plexon server:

```
PL_Close(s);  
s = 0;
```

Note that the functions PL\_WaitForServer and PL\_SendUserEvent will not work over PlexNet; they can only be used when the calling Matlab program is running on the same computer as OmniPlex. Also, the PL\_SendUserEvent function is not currently supported on OmniPlex systems.

## **Trial-Oriented API**

**PL\_TrialDefine** - defines various trial events and parameters

**PL\_TrialStatus** - returns various information on current trial

**PL\_TrialEvents** - returns list of events accumulated during trial

**PL\_TrialSpikes** - returns list of spikes accumulated during trial

**PL\_TrialAnalogSamples** - returns list of analog samples accumulated during trial

The Trial-oriented functions provide a ‘middleware’ layer that buffers the data for an experimental trial and delivers it to Matlab all at once when the trial finishes. This frees the Matlab application from having to buffer the data on its own.

The Trial-oriented functions behave like a separate API. However, you still need to call **PL\_InitClient** before using any **PL\_TrialXXX**. When your client exits you should close the connection with the Plexon server using **PL\_Close**.

While is possible to use trial-related functions and other **PL\_XXX** functions together, it is not advised. By using other functions during trial you risk losing some data because it was retrieved from the Plexon server by some other means, or because it was over-written in the buffer due to delays.

However, it is completely safe to use other **PL\_XXX** functions between trial runs.

## **Test Scripts**

The test scripts

test\_pars.m, test\_ts.m, test\_wf.m, test\_trial.m, test\_ad.m, etc

in the Samples subdirectory provide examples of complete scripts that open a connection to the server, get some data and close the connection.

## **Support**

If you have questions about using the Client Software Development Kit, please contact [support@plexon.com](mailto:support@plexon.com).

## Changes:

### Version 1.3.1, 8/17/15

- Updated PL\_GetADVEx so that the sample data for a range of channels is never “split” across successive calls.
- Fixed bug where PL\_GetADEx and PL\_GetADVEx were not returning per-channel sample counts required for OmniPlex systems.
- Fixed bug in PL\_TrialAnalogSamples where an incorrect number of samples was being returned when used with OmniPlex systems.
- Fixed various OmniPlex-compatibility issues in SoftServer.
- Continuous and triggered LFP sample clients are no longer supported.
- Minor documentation changes (mexw32/64, PlexClient.dll) and cleanup.

### Version 1.3.0, 11/18/05

- Started this document
- Added PL\_SendUserEvent
- Added PL\_VTInterpret
- Added trial-based API

### Version 1.3.1, 2/22/06

- Added advanced handling of strobed events to Trial-based API

### Version 1.4.0, 8/1/07

- Updated PL\_GetPars to return per-channel sampling rates and a list of true channel numbers for enabled continuous A/D channels
- Added PL\_GetADEx to return continuous sample data for systems with both “slow” and “fast” continuous A/D channels
- Added PL\_GetGains to return total per-channel gains for spike channels
- Added PL\_GetADGains to return total per-channel gains for A/D channels
- Added continuous and triggered LFP clients, including real-time spectrogram and power spectral density displays

### Version 1.4.1, 1/23/08

- Added PL\_GetADV and PL\_GetADVEx, which are identical to PL\_GetAD and PL\_GetADEx except that they return sample values in volts, accounting for preamp and NIDAQ gains, i.e. true electrode voltages are returned. Note that the analog channel preamp gains in Sort Client (e.g 1000) must be defined correctly for your preamp in order to get correctly scaled values.