



---

## TranSpec++

### Programming Library for TranSpec<sup>®</sup> - Spectrometer

---

For developing your own spectroscopy and photometry applications using our TranSpec-DSP spectrometers running on Windows 2000/XP, we provide our powerful and easy-to-use programming library **TranSpec++**.

The TranSpec-DSP high-speed spectrometers are equipped with an integrated processor system, which performs the entire data acquisition and further spectra preprocessing independently of the connected personal computer. Now the TranSpec++ class library is used to communicate with TranSpec-DSP in a client/server scheme, for instance, setting up measurement parameters, request TranSpec to perform certain measurement runs and retrieve the measured spectra back from the instrument. With this, the entire spectra acquisition, i.e. scanning the diode array, a possible raw data average, dark current correction and the spectra normalization is fully encapsulated in TranSpec++ and TranSpec's firmware, so that the programmer can focus the development onto his application.

- Runtime licensed Dynamic Link Library (DLL) providing standard C calls  
Compatible with common C/C++ compilers, Visual Basic and VBA (Excel), LabView, etc.
- Extensive Parameter Checks and Measurement Status Verification  
You hardly can do anything wrong when working with TranSpec++
- Detailed User's Manual as compiled HTML file and printed PDF document
- Demo software as Windows console application, including C/C++ source code
- See next page for a programming example!

Technical specifications on next page ►



### TranSpec++ • Technical Specifications

October 2007, related to version 4.04, without guarantee, subject to changes.

#### Minimum Hardware and Software Requirements

- PC with at least Pentium-4
- Windows 2000 or Windows XP **Note:** TranSpec++ does not run on Windows NT/95/98/ME
- C/C++ development system (MS Visual Studio recommended), Delphi, Visual Basic or VBA, LabView
- TranSpec Spectrometer
- TranSpec 2000 software is recommended, but not required

#### Programming Example

The following shows a small programming example, in order to demonstrate the capabilities and easy usage of the TranSpec++ library. As a simple exercise, we program the fully automatic measurement of a dark current corrected emission spectrum:

```
#include "TranSpec.h" // references the global instance of the class CTranSpec as "theTranSpec"
...
// Setup of general parameters for scanning:
MEASPARA sPara;
sPara.IntegrationTime = 10000; // integration time 10000 microseconds = 10 milliseconds
sPara.nNumberAverage = 1; // no raw data average
sPara.nNumberScans = 1; // only one spectrum to measure
sPara.sTrigger.eTriggerType = NONE; // no trigger checking

// Upload parameters to TranSpec:
theTranSpec.Request_SetMeasParameter( sPara );

// In order to perform a dark current correction, we now measure a dark current spectrum, which will be stored in TranSpec.
// After that, each measured spectrum will be automatically dark current corrected by TranSpec.
theTranSpec.Request_StartMeasurement( DARKCURRENT );

// Now we start the measurement of the emission spectrum, which is going to be stored as a raw spectrum in TranSpec:
theTranSpec.Request_StartMeasurement( SPECTRUM );

// Wait until the measurement is done. In our example this would take 10 milliseconds only:
while ( theTranSpec.GetStatus_IsMeasuring() )
    theTranSpec.Request_GetStatus(); // get current status of TranSpec

// Measurement is done. Now we download the measured raw spectrum from TranSpec....
theTranSpec.Request_GetRawSpectrum( SPECTRUM );

// ...and compute the raw spectrum into Emission units:
theTranSpec.GetData_NormalizeRawSpectrum( SPECTRUM, EMISSION );
```

**Note:** TranSpec is a registered German trademark of Dipl.-Ing. (FH) Th. Fuchs, Engineer's Office for Applied Spectroscopy. All other mentioned product names are or possibly might be trademarks or registered trademarks of their owners.