


[Contact](#)

Note: This is an old calibration program written many years ago and is shown here as a temporary introduction to user written calibration. When we officially release 4.1.10 we hope to have a new calibration program available for download at which point this documentation will be updated.

This section describes the process required for an application to handle its own calibration.

Calibration can be performed in 1 of 3 ways:

1. The desktop user interface as described [here](#).
2. Run [TBcalib.exe with command-line arguments](#) as required.
3. Total application control via UPDD's Application Program Interface.

This section covers option 3 above. However, before you embark on this method of calibration you should checkout the full characteristics of TBcalib. The program allows for different calibration images and caters for all the subtle (and somewhat complex) issues that need to be considered when calibrating a device. We **highly recommend** you use TBcalib to handle calibration. Nevertheless, you are able to handle your own calibration if you wish. Please note that example calibration source code is available from Touch-Base. You should make yourself familiar with this code before proceeding.

An overview of the calibration sequence for UPDD is as follows:

1. Call the API's [TBApiInit](#) and [TBApiOpen](#)
2. Register a callback function to read raw touch data using [TBApiRegisterDataCallback](#) and the constant `_ReadDataTypeXY (0x01)`. The callback function should store the co-ordinates received.
3. Put a calibration mark on the screen.
4. Wait for a co-ordinate packet.
5. Save the values of the calibration points.
6. Repeat 3 - 5.
7. Write the raw co-ordinates to the registry (with updd version 4.1.x these are stored in the settings file)

The simplest way to write calibration information to the registry is to use [TBApiSetCalibrationStyle](#). This API call can be used to change the settings for an existing style, or to create a new style.

The example program 'CalibrationSample' handles calibration on a single device with the following characteristics:

- 2 point calibration
- Calibration points are located 12% in from the corners
- 10 second timeout
- Not transparent
- No rotation

Calibration Sample Callback routine

The key processing of this program is located in the callback routine (this is registered to receive co-ordinate data). Check the full source code in module CalibrationSampleDlg.cpp for more details.

```
void CCalibrationSampleDlg::PointerDataCallback(_PointerData* data)
{
    static long theLastRawX = 0x8888;
    static long theLastRawY = 0x8888;

    if (m_bFinished)
        return;

    // Ignore touches too close to first
    if(Delta(data->xy.rawx,theLastRawX) < 5 && Delta(data->xy.rawy,theLastRawY) < 5)
        return;
    else
    {
        if (theLastRawX == 0x8888)
        {
            // First touch received
            theLastRawX = data->xy.rawx;
            theLastRawY = data->xy.rawy;
            DisplayWindow(false); // Display second bitmap
        }
        else
        {
            // Second touch received
            _CalStyle aStyle; // Define storage for calibration style details
            HTBSTYLE hStyle; // Handle to be used for accessing this calibration style
            ((CWnd*)GetDlgItem(IDC_BR))->SetWindowPos(NULL, 0, 0, 0, SWP_NOSIZE | SWP_HIDEWINDOW );

            int aDeviceId = TBApiGetRelativeDevice(0); // get the id for one and only controller
        }
    }
}
```

```

// Given a calibration style name, return the style handle (returns -1 if not found)
hStyle = TBApiGetCalibrationStyleByName(aDeviceId, _T("Normal"));
if (hStyle == -1)
{
    AfxMessageBox("Cannot get calibration handle");
    return;
}

// Set up calibration details in structure.
// NB. An alternative method is simply to set the CALX/Y0/1 values
// in the UPDD registry.
_tcscpy( aStyle.theName, _T("Normal"));
aStyle.theTransparent = 0;
aStyle.theNPoints = 2;
aStyle.theRotation = 0;
aStyle.theStartPct = 12;
aStyle.theTimeout = 10;
aStyle.thePoints[0].x = theLastRawX; // raw co-ordinates received earlier
aStyle.thePoints[0].y = theLastRawY; // ditto
aStyle.thePoints[1].x = (int)data->xy.rawx; // this packet
aStyle.thePoints[1].y = (int)data->xy.rawy; // ditto

if (!TBApiSetCalibrationStyle(aDeviceId, hStyle, &aStyle))
    AfxMessageBox("Unable to set calibration style");
else
{
    if (TBApiActivateCalibrationStyle(aDeviceId, _T("Normal"), true))
    {
        TBApiApply();
        AfxMessageBox("Calibration successful!");
    }
    else
        AfxMessageBox("Failed to activate calibration");
}
m_bFinished = true;
}
}
}

```

The calibration style may not exist, i.e. it need not have been previously defined by the UPDD Console, Calibration, Style. To define a new style, remove the call to **TBApiGetCalibrationStyleByName** and simply pass **TBNEW_STYLE** as the style handle (i.e. the second) argument to **TBApiSetCalibrationStyle**.

Values supplied in thePoints are the raw co-ordinates recorded. Note that the calibration **reference** points are re-calculated by this function.

Once calibration is complete ensure that [TBApiUnregisterDataCallback](#), [TBApiClose](#) and [TBApiTerminate](#) are called before exiting the application.

Calibration API Calls

A number of API calls support calibration functionality:

TBApiGetCalibrationStyle	Get the calibration style details for the specified device and style.
TBApiSetCalibrationStyle	Modifies an existing calibration style or adds a new style.
TBApiRemoveCalibrationStyle	Delete a calibration style.
TBApiActivateCalibrationStyle	Activate the named calibration style for the specified device.
TBApiGetCalibrationAsStyle	Get the calibration settings for the specified device.
TBApiGetCalibrationStyleByName	Given a calibration style name, return the style handle.
TBApiGetSettingSZEx	Can be used to get the names of all calibration styles for the specified driver. Replaces the UPDD V2 call TBApiGetCalibrationStyleNames.
TBApiAutoSetSwapXY	Examines the recorded calibration points for a device to see if "SwapXY" is incorrectly set.

Calibration Style Structure

This structure is used in many of the calibration API calls and can be found in the header file **TBapi.h** (struct **_CalStyle**) or **Tbapi.bas** (Co-ordinateData Type).

EEPROM consideration

There are currently no API's to write the calibration data to eeprom although these may be implemented in a future release. Where UPDD supports writing calibration data to a controller's eeprom (as documented [here](#)) you can call the **tbcalib** command line interface with the parameter **eepromwrite**, e.g. **tbcalib eepromwrite**.

Contact

For further information or technical assistance please email the technical support team at technical@touch-base.com.

