API version 5 to version 6 conversion notes

With UPDD version 6 we have made significant changes. This is to remove and / or improve large amounts of legacy code that; having originally been designed for older architectures and having evolved over many years was becoming difficult to maintain and extend. This approach extends to the API and consequently customers wishing to utilise the UPDD API in version 6 and above must make changes to their code. Whilst the changes in the API are extensive, in most cases the external alterations are merely name changes to clarify the API usage. For most users the changes to upgrade to V6 should in fact be minimal, and for the most part merely require a number of "find / replace" operations.

A suggested conversion procedure is outlined below. After checking your software compiles and links correctly with UPDD V5 API, change your project include and link settings. If in doubt refer to the documentation for your compiler or IDE for further information on this. The UPDD API file names are now.

	Header	Import Library	Dynamic Library
Windows	upddapi.h	upddapi.lib	upddapi.dll
Mac OS/x	upddapi.h	n/a	libupddapi.n.n.n.dylib
Linux	upddapi.h	n/a	upddapi.so

We recommend that you then compile your program and deal with the resulting compiler errors following the guidance below. It is not the intention of this guide to list every possible change as a very large number of items are affected; most of which are obsolete. If you find an item which is missing first check upddapi.h for any likely replacement values. In the case of an item that cannot be identified contact Touch-base support for further information.

References to the _PointerData structure should be change to reflect the new name _PointerEvent. The union within this struct previously named pd is now named pe.

The internal management of settings has changed significantly. For a V5 to V6 conversion please note these specific items:

The APIs names have changed:

V5 names	V6 names
TBApiGetSettingDWORD	TBApiGetSettingFromInt

API version 5 to version 6 conversion notes

TBApiGetSettingSZ	TBApiGetSetting
TBApiSetSettingDWORD	TBApiSetSettingFromInt
TBApiGetSettingSZ	TBApiSetSetting

The "Set" APIs now have an extra argument, aDeviceSpecific; To retain the same behavior as in V5 pass false if aHandle is zero, otherwise true. The purpose of this is described in more detail in the API documentation. The global settings API's have been removed. Setting names that previously had spaces now use an underscore for a separator. e.g. "monitor number" is now "monitor_number". The setting name "logical desktop segment" has been replaced by "display_segment" and "monitor_number".

A number of settings have been removed or renamed, these are mostly obsolete or settings intended for internal use only so it is unlikely your program will be affected; but if this is the case contact Touch-Base support for further information.

Some important setting changes are listed here:

"logical desktop segment" and "MonitorNumberForSegment" have been removed. These are replaced by monitor_number and monitor_segment.

Note that there is not a direct correspondence between the old and new values and you should refer to the specific documentation for these new settings if necessary.

Further note that the new values are delivered in API callbacks where appropriate so it might not be necessary to reference these settings directly.

Callback types previously named _ReadDataTypeXXX are now named _EventTypeXXX

e.g. _ReadDataTypeXY is now _EventTypeXY.

An exception to this is that _ReadDataTypeXYCal is now EventTypeXYNoMask.

Remove calls to TBApiInit and TBApiTerminate, these are no longer required.

Most API functions now require an open connection to the driver, so check you do not make calls before calling TBApiOpen or after calling TBApiClose. The two exceptions to this rule are TBApiGetApiVersion and TBApiRegisterEvent. The latter can only be use to register to receive CONFIG_EVENT_CONNECT / CONFIG_EVENT_DISCONNECT events.

API version 5 to version 6 conversion notes

If you did not previously call TBApiOpen then you will need to do so. Generally it is recommended that you should call TBApiOpen once only at the beginning of your program and TBApiClose once at the end.

TBApiOpen executes asynchronously. A client must wait for a connection either asynchronously or synchronously.

The synchronous approach involves calling TBApiIsDriverConnected in a loop until true is returned.

Example

```
while (!TBApiIsDriverConnected())

{

#ifdef_WIN32

Sleep(10);

#else

usleep(100);

#endif

}
```

For the asynchronous approach

Register as follows to receive configuration notifications before calling TBApiOpen.

Example

```
TBApiRegisterEvent(0, 0, _EventConfiguration, MyConnectCallback);


TBApiOpen ();
```

MyConnectCallback will be called when connected.

API version 5 to version 6 conversion notes

Example

```
staticvoid TBCALL MyConnectCallback(unsignedlong/*context*/, _PointerEvent* ev)

{

if (ev->pe.config.configEventType == CONFIG_EVENT_CONNECT)

{

….

}

}
```

Both these approaches are illustrated in the [cbconsole example](#).

TBApiMousePortInterfaceEnable and TBApiMousePortInterfaceEnableEx have been combined as TBApiMousePortInterfaceEnable with a new set of parameters.

TBApiApply and TBApiApplyNoReload are obsolete and are no longer used. In UPDD v6 calls to tbapisetxxx functions are effective immediately removing the need for these functions.

If you are using an API function that does not have a direct replacement in UPDD API V6, it is likely that it was either considered obsolete or is a function not yet supported in V6. Contact Touch-base support for further information in this event.

V6 new features

This document is aimed at the minimum required to convert a UPDD version 5 based program to work with version 6.

However, whilst it's not necessary for a direct conversion your attention is drawn to the new features in V6. Notably the new event type of [_EventTypeDigitiserEvent](#).

Digitiser events offer a means to gather all information relevant to a pen or touch input in a single event. It also has enhanced co-ordinate information which, for example, calculate screen positions in multi monitor environments; which previously had to be performed by the client via reference to a number of settings and APIs.

API version 5 to version 6 conversion notes

Other improvements are :

TBApiEnableApiTrace, although previously implemented this is improved and now provides debug output from all APIs to assist in diagnostics.

TBApiGetLastError, most apis return a Boolean value (ok / fail) this api provides a text based explanation of the error.

TBApiIsDeviceConnected, provides a supported means to see if a PnP device is connected.

Notes

1. UPDD v6 offers improved support for checking that a connection to the driver is available. These are not required to port an application to V6, but you might wish to consider using the new _EventConfiguration callback in conjunction with CONFIG_EVENT_CONNECT / CONFIG_EVENT_DISCONNECT to check for a driver connection before making further API calls. Refer to UPDD api documentation for further information.
2. TBApiMousePortInterfaceEnable is obsolescent; you should not use it in new code and should consider replacing it in existing code. Refer to V6 API documentation for further information.
3. The V6 API is fully re-entrant. In V5 each API call locked a common mutex essentially serialising calls to the API. In the event that this change in behavior causes problems with an existing program then the calling program will need to be redesigned to properly manage its thread synchronization. A simple solution would be to use a single common mutex and acquire it for the duration of each API call.
4. When working with multi touch devices the stylus information in the callback packet is exactly that sent by the controller. In v5 there was in some cases "round robin" pre-processing applied. In cases where client code relies on the primary touch being stylus zero; some changes may be required.

    A simple example of this is illustrated in the [cbconsole example](#).

.telerik-reTable-2 { border-collapse: collapse; border: solid 0px; font-family: Tahoma; } .telerik-reTable-2 tr.telerik-reTableHeaderRow-2 { border-width: 1.0pt 1.0pt 3.0pt 1.0pt; margin-top: 0in; margin-right: 0in; margin-bottom: 10.0pt; margin-left: 0in; line-height: 115%; font-size: 11.0pt; font-family: "Calibri" , "sans-serif"; width: 119.7pt; border: solid white 1.0pt; border-bottom: solid white 3.0pt; background: #4F81BD; padding: 0in 5.4pt 0in 5.4pt; color: #FFFFFF; } .telerik-reTable-2 td.telerik-reTableHeaderFirstCol-2 { border-width: 1.0pt 1.0pt 3.0pt 1.0pt; border: solid white 1.0pt; border-bottom: solid white

API version 5 to version 6 conversion notes

3.0pt; padding: 0in 5.4pt 0in 5.4pt; } .telerik-reTable-2 td.telerik-reTableHeaderLastCol-2 { border-width: 1.0pt 1.0pt 3.0pt 1.0pt; border: solid white 1.0pt; border-bottom: solid white 3.0pt; padding: 0in 5.4pt 0in 5.4pt; } .telerik-reTable-2 td.telerik-reTableHeaderOddCol-2 { border-width: 1.0pt 1.0pt 3.0pt 1.0pt; border: solid white 1.0pt; border-bottom: solid white 3.0pt; padding: 0in 5.4pt 0in 5.4pt; } .telerik-reTable-2 td.telerik-reTableHeaderEvenCol-2 { border-width: 1.0pt 1.0pt 3.0pt 1.0pt; border: solid white 1.0pt; border-bottom: solid white 3.0pt; padding: 0in 5.4pt 0in 5.4pt; } .telerik-reTable-2 tr.telerik-reTableOddRow-2 { color: #666666; background-color: #F2F3F4; vertical-align: top; } .telerik-reTable-2 tr.telerik-reTableEvenRow-2 { color: #666666; background-color: #E7EBF7; vertical-align: top; } .telerik-reTable-2 td.telerik-reTableFirstCol-2 { margin-top: 0in; margin-right: 0in; margin-bottom: 10.0pt; margin-left: 0in; line-height: 115%; font-size: 11.0pt; font-family: "Calibri" , "sans-serif"; width: 119.7pt; border-top: none; border-left: solid white 1.0pt; border-bottom: none; border-right: solid white 3.0pt; background: #4F81BD; padding: 0in 5.4pt 0in 5.4pt; color: #FFFFFF; } .telerik-reTable-2 td.telerik-reTableLastCol-2 { padding: 0in 5.4pt 0in 5.4pt; } .telerik-reTable-2 td.telerik-reTableOddCol-2 { padding: 0in 5.4pt 0in 5.4pt; } .telerik-reTable-2 td.telerik-reTableEvenCol-2 { padding: 0in 5.4pt 0in 5.4pt; } .telerik-reTable-2 tr.telerik-reTableFooterRow-2 { color: #666666; background-color: #FFFFFF; vertical-align: top; padding: 0in 5.4pt 0in 5.4pt; } .telerik-reTable-2 td.telerik-reTableFooterFirstCol-2 { margin-top: 0in; margin-right: 0in; margin-bottom: 10.0pt; margin-left: 0in; line-height: 115%; font-size: 11.0pt; font-family: "Calibri" , "sans-serif"; width: 119.7pt; border-top: none; border-left: solid white 1.0pt; border-bottom: none; border-right: solid white 3.0pt; background: #4F81BD; padding: 0in 5.4pt 0in 5.4pt; color: #FFFFFF; } .telerik-reTable-2 td.telerik-reTableFooterLastCol-2 { padding: 0in 5.4pt 0in 5.4pt; } .telerik-reTable-2 td.telerik-reTableFooterOddCol-2 { padding: 0in 5.4pt 0in 5.4pt; } .telerik-reTable-2 td.telerik-reTableFooterEvenCol-2 { padding: 0in 5.4pt 0in 5.4pt; } Other important changes in V6:

The updd setting file tbupdd.ini is now an SQL database file updd.db so no text based access is possible.

V6 comprises 32-bit components for 64-bit and 32-bit Windows OS

Complete component listings are here.

Touch-Base Support
http://support.touch-base.com/Documentation/50119/API-version-5-to-version-...