ADVANTECH
*Enabling an Intelligent Planet*

# Advantech AE Technical Share Document

| | | | |
|---|---|---|---|
| **Date** | 2016/8/15 | **SR#** | 1-2533309631 |
| **Category** | ■ FAQ □ SOP | **Related OS** | Windows / Linux |
| **Abstract** | Necessary Programming Changes as Advancing to Advantech DAQNavi 4.0 | | |
| **Keyword** | Software upgrade, DAQNavi 4.0, DAQNavi SDK | | |
| **Related Product** | PCI-1706, PCI-1710U/HGU/UL, PCI-1711U/UL, PCI-1712, PCI-1712L, PCI-1713U, PCI-1714U/UL, PCI-1715U, PCI-1716, PCI-1718HDU/GU, PCI-1720U, PCI-1721, PCI-1723, PCI-1724U, PCI-1727U, PCI-1730U,PCI-1733, PCI-1734, PCI-1735U, PCI-1737U, PCI-1739U, PCI-1741U, PCI-1742U, PCI-1747U, PCI-1750, PCI-1751, PCI-1752U, PCI-1753, PCI-1754, PCI-1756, PCI-1758UDI, PCI-1758UDIO, PCI-1758UDO, PCI-1760U, PCI-1761, PCI-1762, PCI-1780U, PCI-1784U, PCIE-1730, PCIE-1751, PCIE-1752, PCIE-1753, PCIE-1754, PCIE-1756, PCIE-1760, PCIE-1802, PCIE-1810, PCIE-1816/H, PCIE-1840, PCM-3718H, PCM-3718HO, PCM-3724, PCM-3725, PCM-3730, PCM-3730I, PCM-3753I, PCM-3761I, PCM-3810I, PCM-3813I, PCM-27D24DI, USB-4702, USB-4704, USB-4711A, USB-4716, USB-4718, USB-4750, USB-4751, USB-4751L, USB-4761 | | |

The changes on SW structure had been made in DAQNavi 4.0. We'd strived as much as possible to minimize the impact to existing users, however inevitably some of the functions will no longer be available in the original calling way, mostly happens on the functions relating to the register level work on the boards for example BoardID (read only), read private region…etc. The programming must be changed if those kinds of functions had already been applied to your program. Otherwise, you'll get errors indicating that the function isn't available as you run the application. This document will demonstrate what should be changed in your programming as advancing to DAQNavi 4.0.

## ■ <u>Brief Solution - Step by Step</u>:

Here we separate the situation into 3 types and list the correspondent solutions:

### 1. <u>Functions in DeviceCtrlExt</u>

Those methods and properties which are listed in class DeviceCtrlBaseExt, (made to public after SDK revision 3.2.6.0) is direct a class under namespace Automation::BDaq and won't be available in the original function calling style, for example to get BoardID in DAQNavi 3.0: (we will demonstrate C++ and C# in this document)

*<u>C++:</u>*

```
int boardID = instantDiCtrl->getBoardId();
```

*<u>C#:</u>*

```
Int boardID = instantDiCtrl.BoardId();
```

The followings are the functions of class DeviceCtrlBaseExt we list in bdaqctrl.h of DAQNavi 3.0:

```
// method
virtual ErrorCode BDAQCALL ReadRegister(int32 startAddress, int32 length, void *data) = 0;
virtual ErrorCode BDAQCALL WriteRegister(int32 startAddress, int32 length, void *data) = 0;
virtual ErrorCode BDAQCALL ReadPrivateRegion(int32 signature, int32 length, uint8 *data) = 0;
virtual ErrorCode BDAQCALL WritePrivateRegion(int32 signature, int32 length, uint8 *data) = 0;

// property
virtual ProductId                    BDAQCALL getProductId() = 0;
virtual int32                        BDAQCALL getBoardId() = 0;
virtual void                         BDAQCALL getBoardVersion(int32 length, wchar_t *ver) = 0;
virtual void                         BDAQCALL getDriverVersion(int32 length, wchar_t *ver) = 0;
virtual void                         BDAQCALL getDllVersion(int32 length, wchar_t *ver) = 0;
virtual void                         BDAQCALL getLocation(int32 length, wchar_t *loc) = 0;
virtual int32                        BDAQCALL getPrivateRegionLength() = 0;
virtual int32                        BDAQCALL getHotResetPreventable() = 0;
virtual ICollection<int32>*          BDAQCALL getBaseAddresses() = 0;
virtual ICollection<int32>*          BDAQCALL getInterrupts() = 0;
virtual ICollection<TerminalBoard>*  BDAQCALL getSupportedTerminalBoard() = 0;
virtual ICollection<EventId>*        BDAQCALL getSupportedEvents() = 0;

virtual TerminalBoard                BDAQCALL getTerminalBoard() = 0;
virtual ErrorCode                    BDAQCALL setTerminalBoard(TerminalBoard board) = 0;
virtual int32                        BDAQCALL getLoadingTimeInit() = 0;
virtual ErrorCode                    BDAQCALL setLoadingTimeInit(int32 init) = 0;
```

The correspondent new programming style is listed below: (get BoardID)

*C++:*

```
    DeviceCtrl * devCtrl = instantDiCtrl->getDevice();
    Int boardID = devCtrl->getBoardId();
```

*C#:*

```
    int boardID = instantDiCtrl.Device.BoardId;
```

## 2. Undocumented functions

Normally this is a customized solution. It's not recommended upgrading the SDK in this situation. Please contact your local support to find out the solution if you'd like to upgrade the SDK.

## 3. The program without the functions above (DeviceCtrlExt & Undocumented)

You can still use the built program without rebuilding or code revision. The program should get used to it automatically. The old programming style would still be available but hidden so that you won't see the 3.0 style functions be listed in the IDEs when you program your application.

■ **Reference:**