この度は、Jungo ドライバ開発ツール「WinDriver」をお買い求めいただき誠にありがとうございます。

ご使用の前にこの README を必ずお読みください。

WinDriver のインストール

Windows 10 / 10 IoT Core / Server 2016 / 8.1 / Server 2012 R2 / 8 / Server 2012 / 7 / Server 2008 R2 / Server 2008 x86 32 ビット、x64 64 ビット用、Windows CE または Linux オペレーティング システム用の WinDriver 製品をダウンロードしてください。

インストール ファイル一覧 (2020年1月21日現在)

WinDriver			
Bus	**オペレーティング システム**	**ダウンロード ファイル名**	**Version**
PCI / PCI-Express	Windows 10 / 10 IoT Core / Server 2016 / 8.1 / Server 2012 R2 / 8 / Server 2012 / 7 / Server 2008 R2 / Server 2008 32bit	WD1421.EXE	14.21
	Windows 10 / 10 IoT Core / Server 2016 / 8.1 / Server 2012 R2 / 8 / Server 2012 / 7 / Server 2008 R2 / Server 2008 x64	WD1421X64.EXE	14.21
	Windows CE (別名 Windows Embedded Compact) 4.x - 7.x (Windows Mobile を含む、x86/MIPS/ARM)	WD1230CE.EXE	12.30
	Windows CE 8.x (別名 Windows Embedded Compact 2013、x86/ARM)	WD1230CE2013.EXE	12.30
	Linux x86 2.6.x またはそれ以降 (32 bit)	WD1421LN.tgz	14.21
	Linux x86 2.6.x またはそれ以降 (64 bit)	WD1421LNX86_64.tgz	14.21
	Linux ARM (32 bit)	WD1421LNARM.tgz	14.21
	Linux ARM (64 bit)	WD1421LNARM64.tgz	14.21
USB	Windows 10 / 10 IoT Core / Server 2016 / 8.1 / Server 2012 R2 / 8 / Server 2012 / 7 / Server 2008 R2 / Server 2008 s32bit	WD1421.EXE	14.21
	Windows 10 / 10 IoT Core / Server 2016 / 8.1 / Server 2012 R2 / 8 / Server 2012 / 7 / Server 2008 R2 / Server 2008 / x64	WD1421X64.EXE	14.21
	Windows CE (別名 Windows Embedded Compact) 4.x - 7.x (Windows Mobile を含む)	WD1230CE.EXE	12.30
	Windows CE 8.x (別名 Windows Embedded Compact 2013、x86/ARM)	WD1230CE2013.EXE	12.30
	Linux x86 2.6.x またはそれ以降 (32 bit)	WD1421LN.tgz	14.21
	Linux x86 2.6.x またはそれ以降 (64 bit)	WD1421LNX86_64.tgz	14.21
	Linux ARM (32 bit)	WD1421LNARM.tgz	14.21
	Linux ARM (64 bit)	WD1421LNARM64.tgz	14.21

メモ:

1. WinDriver for Windows (PCI / PCI-Express / ISA 用) と WinDriver USB (USB 用) のインストール ファイルは共通です。評価版として使用する場合、両方の製品を 30 日間ご試用になれます。製品を購入して登録ユーザーとしてインストールした場合は、開発元より発行されるライセンス コードにより購入された製品が動作するようになります。
2. インストール ファイル名の数字は、バージョンを表します (たとえば、WD1421.EXE はバージョン 14.21 となります)。そのためインストール ファイル名は、最新バージョンがリリースされるたびにバージョンをあらわす数字が変更されます。

WinDriver のインストール手順

● WinDriver for Windows 10 / 10 IoT Core / Server 2016 / 8.1 / Server 2012 R2 / 8 / Server 2012 / 7 / Server 2008 R2 / Server 2008 (PCI / PCI-Express / ISA / USB) をインストールするには

必要なシステム:

1. x86 32 ビットまたは 64 ビット (x64: Intel EM64T または AMD64) プロセッサ、Microsoft がサポートする ARM プロセッサ
2. C または .NET をサポートする開発環境
3. Windows 10 / 10 IoT Core / Server 2016 / 8.1 / Server 2012 R2 / 8 / Server 2012 / 7 / Server 2008 R2 / Server 2008
   注意: 64-Bit Windows OS では、デジタル署名の強制を一時的に無効にする設定が必要です。

● WinDriver for Windows 10 IoT Core をインストールするには

WinDriver は、Windows 10 IoT Core でのドライバ開発をサポートします。Windows 10 IoT エンタープライズは Windows 10 のフルバージョンであるため、通常の WinDriver for Windows Desktop / Server を使用することができます。

必要なシステム:

1. DriverWizard を使用するには、開発コンピュータの Windows デスクトップに WinDriver をインストール
2. 対象のコンピュータとの SSH セッションを許可するネットワーク接続

インストール手順:

注意: 必ず Administrative 権限を持ったユーザーで WinDriver のインストールを行ってください。

1. WinDriver を弊社ダウンロードサイトよりご希望の製品をクリックしダウンロードします。
2. Windows x86 32 ビット OS の場合、**WD1421.EXE** ファイル、Windows x64 64 ビット OS の場合、**WD1421X64.EXE** ファイルがダウンロードされます。[**Next**] ボタンをクリックします。
3. ライセンス同意書に目を通してください。同意する場合は、[**Yes**] をクリックします。
4. WinDriver のインストール先を指定します。
5. [**Setup Type**] ダイアログ ボックスで、次のいずれかを選択します:
   - **Typical** – WinDriver モジュールをインストールします (WinDriver ツールキットと特定チップセット用の API)。
   - **Compact** – WinDriver ツールキットだけをインストールします。
   - **Custom** – インストールする WinDriver のモジュールを選択します。
6. インストールに必要なファイルのコピーが完了すると、クイック スタート ガイドを表示するか選択します。
7. セットアップを完了したら、コンピュータを再起動してください。

注意:
WinDriver のインストールでは、**WD_BASEDIR** 環境変数を定義します (インストール中に選択した WinDriver のディレクトリの場所を示します)。WinDriver の DriverWizard でコードを生成する際には、この変数を使用します - 生成したコードを保存するデフォルトのディレクトリで、生成された project / make ファイルの include パスに使用します。サンプルの Kernel PlugIn プロジェクトおよび makefile でも、この変数を使用します。従って、WinDriver のインストール後、WinDriver のディレクトリの名前 / 場所を変更する場合、WD_BASEDIR 環境変数の値を変更し、新しい WinDriver のディレクトリの場所を指すように設定する必要があります。以下の手順で、WD_BASEDIR の値を変更できます:

1. システムのプロパティ ダイアログを開きます:
   [スタート] – [設定] – [コントロール パネル] – [システム]
2. [詳細設定] タブで、[環境変数] ボタンをクリックします。
3. [システム環境変数] ボックスで、WD_BASEDIR 変数を選択し、[編集] ボタンをクリックするか、変数をダブル クリックします。
4. [システム変数の編集] ダイアログで、[変数値] を新しい WinDriver のディレクトリのフルパスに置き換えて、[OK] をクリックし、[環境変数] ダイアログで [OK] ボタンをクリックし、 更に、[システムのプロパティ] ダイアログで、[OK] ボタンをクリックします。

ERROR_FILE_NOT_FOUND エラーでインストールに失敗する場合、Windows のレジストリの **HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion** に **RunOnce** キーが存在することをご確認ください。Windows Plug-and-Play で INF ファイルを使用してドライバを正しくインストールする際にこのレジストリ キーが必要です。**RunOnce** キーがない場合、作成し、再度 INF ファイルのインストールをお試しください。

注意:
ツールキットを使用するためには、Windows OS 上でデジタル署名の強制を一時的に無効にする設定が必要です。Windows 10 はデジタル署名付きカーネル ドライバが必要です。WinDriver インストール中にインストールされる WinDriver virutal デバイスは既にデジタル署名されています。しかし、DriverWizard ユーティリティによって作成された .inf ファイルはデジタル署名されていないため Windows 10 上でインストールすることができません。.inf ファイルをインストールするには、デジタル署名を行うか、または署名せずにデジタル署名の強制を一時的に無効にします。以下の手順で、デジタル署名の強制を一時的に無効にできます:

1. [スタート] メニューボタンをクリックします。
2. [シフト] ボタンをキーボード上で押したまま、[スタート] メニューの左側にある [電源] アイコンをクリックします。
3. 開いたメニューから [再起動] を選択します。
4. 開いたスクリーンから [トラブルシューティング] を選択します。
5. [詳細設定] を選択します。
6. [スタートアップ設定] を選択します。
7. コンピュータが再起動し、スタートアップ設定スクリーンが開きます。[F7] を押しドライバ署名の強制を無効にします。
8. .inf ファイルがインストールされます。

登録版ユーザーの場合:

次の手順で、エクセルソフト株式会社から受け取ったライセンス コードを入力して WinDriver を登録します。

1. [**スタート**] メニューから [**プログラム**] - [**WinDriver**] - [**DriverWizard**] の順に選択して DriverWizard を起動します。
2. [**Help**] メニューから [**Licensing Information**] を選択して [**Licensing Information**] ダイアログボックスを表示します。
3. 以前のバージョンのライセンス コードが登録されている場合、[**Deactivate license**] ボタンをクリックして、以前のバージョンのライセンス コードを解除します。
4. [**Please enter your license string**] 入力ボックスにエクセルソフト株式会社から受け取ったライセンス コードを入力して [**Activate license**] ボタンをクリックし、ライセンス コードを登録します。

注意: カーネル上の現在のライセンスをチェックするには、[WinDriver Wizard] を実行して、[**Help**] メニューから [**Licensing Information**] を選択してください。現在、カーネルに設定されている有効なライセンスが表示されます。
ライセンス コードには、スペースおよびピリオドなども含まれますのでライセンス登録の際には、電子メールで受け取ったこの文字列を「コピー＆貼り付け」し、手入力によるミスを防いでください。

● WinDriver for Windows CE をインストールするには

必要なシステム:

3. Windows CE (別名 Windows Embedded Compact) 4.x – 8.x (Windows Embedded Compact 2013、Windows Mobile を含む) が起動する x86 / ARM ターゲット プラットフォーム
   Windows CE 4.x – 7.x (Windows Mobile を含む) が起動する MIPS ターゲット プラットフォーム
4. Windows Windows 10 / 8.1 / Server 2012 R2 / 8 / Server 2012 / 7 / Server 2008 R2 / Server 2008 ホスト開発プラットフォーム
5. Windows CE 4.x – 5.x (Windows Mobile を含む) の場合: 対応するターゲット SDK と Microsoft eMbedded Visual C++ または Microsoft Visual Studio 2005 / 2008
   またはターゲット プラットフォームに対応する BSP (Board Support Package) と Microsoft Platform Builder
   Windows CE 6.x の場合: 対応するターゲット SDK または Windows CE 6.0 plugin と Microsoft Visual Studio 2005 / 2008
   Windows CE 7.x の場合: Windows CE 7 plugin と Microsoft Visual Studio 2008
   Windows CE 8.x の場合: Application Builder for Windows Embedded Compact 2013 plugin と Microsoft Visual Studio 2012 またはそれ以降

インストール手順:

新規の CE ベースのプラットフォームを開発する際に WinDriver CE をインストールする場合:

注意:
以下の手順は、Windows CE Platform Builder を使用して、または対応する Windows CE plugin と MS Visual Studio 2005 / 2008 を使用して Windows CE カーネル イメージをビルドするプラットフォーム開発者向けです。この手順では、これらのプラットフォームの参照を "**Windows CE IDE**" の表記を使用します。
インストール前に Windows CE とデバイス ドライバの統合について Microsoft のドキュメントをよくお読みください。

1. ターゲット ハードウェアに一致したプロジェクト レジストリ ファイルを編集します。
   WinDriver¥samples¥wince_install ¥project_wd.reg

注意: ID の値を定義する場合、project_wd.reg のコメントで使用しているように、正しいフォーマットを使用してください - PCI デバイスの場合は 16 進数、USB デバイスの場合は 10 進数。

2. 対象の Windows CE プラットフォームをコンパイルします (Sysgen ステージ)。
3. 対象のプラットフォームにドライバを統合します。
   (a) Windows CE IDE を実行してプラットフォームを開きます。
   (b) [**Build**] メニューから [**Open Release Directory**] を選択します。
   (c) WinDriver CE カーネル ファイル **¥WinDriver¥redist¥TARGET_CPU¥windrvr1230.dll** を開発プラットフォーム上の **%_FLATRELEASEDIR%** サブディレクトリにコピーします。
   (d) **¥WinDriver¥samples¥wince_install¥project_wd.reg** ファイルの内容を **%_FLATRELEASEDIR%¥project.reg** レジストリ ファイルに追加します。
   (e) **¥WinDriver¥samples¥wince_install¥project_wd.bib** ファイルの内容をバイナリ イメージ ビルダ ファイルの FILES セクション - **%_FLATRELEASEDIR%¥project.bib** ファイルに追加し、ターゲット プラットフォーム (コピーしたテキストの "TODO" コメントを参照) に一致する行のコメントを外します。

   WinDriver CE カーネル ファイル (**windrvr1230.dll**) を永続的に Windows CE イメージ (**NK.BIN**) の一部とする場合にのみこのステップが必要です。例えば、フロッピーディスクを使用してターゲット プラットフォームにカーネル ファイルを移す場合などがこれに該当します。オン デマンドで CESH/PPSH サービスを通して **windrvr1230.dll** をロードする場合、永続カーネルをビルドするまでこのステップを実行する必要はありません。
4. [**Build**] メニューより [**Make Run-Time Image**] を選択し、新しいイメージ **NK.BIN** の名前をつけます。
5. 新規カーネルをターゲット プラットフォームにダウンロードして初期化します ([**Target**] メニューより [**Attach Device**] を選択します。あるいはブートディスクを使用します)。Windows CE 4.x の場合、[**Target**] メニューより [**Attach Device**] ではなく [**Download/Initialize**] を選択します。
6. ターゲット CE プラットフォームを再起動します。WinDriver CE カーネルは自動的にロードします。
7. サンプル プログラムをコンパイルして起動し、WinDriver CE がロードされ、正常に動作するのを確認してください。

Windows CE コンピュータのアプリケーションを開発する際に WinDriver CE をインストールする場合:

注意:
指定がない限り、このセクションの "Windows CE" の記述は、Windows Mobile を含むサポートするすべての Windows CE プラットフォームを表します。

この手順は Windows CE カーネルをビルドするのではなく、ドライバのダウンロードのみ行うドライバ開発者、または既成の Windows CE プラットフォームに MS eMbedded Visual C++ または MS Visual Studio 2005 / 2008 を使用してビルドするドライバ開発者向けです。

1. WinDriver を弊社ダウンロードサイトよりご希望の製品をクリックしダウンロードします。
2. ライセンス同意書に目を通してください。同意する場合は、[**Yes**] をクリックします。
3. WinDriver のインストール先を指定し、インストールします。
4. Windows ホスト開発環境 PC の **¥WinDriver¥redist¥WINCE¥<TARGET_CPU>** ディレクトリ以下から WinDriver CE カーネル モジュール – windrvr1230.dll – をターゲットの Windows CE コンピュータの **¥WINDOWS** ディレクトリにコピーします。
5. 起動時に Windows CE がロードするデバイス ドライバのリストに WinDriver を追加します:
   - **¥WinDriver¥samples¥wince_install¥project_wd.reg** ファイルに記載されたエントリに従って、レジストリを編集します。ハンドヘルド CE コンピュータの Windows CE Pocket Registry Editor を使用するか、または MS eMbedded Visual C++ / MS Visual Studio 2005 / 2008 で提供される Remote CE Registry Editor Tool を使用して実行します。Remote CE Registry Editor ツールを使用するには、対象の Windows ホスト プラットフォームに Windows CE Services がインストールされている必要があります。
   - Windows CE の多くのバージョンでは、起動時に OS のセキュリティ スキーマが署名されていないドライバのロードを防ぎます。従って、起動後に、WinDriver のカーネル モジュールを再ロードする必要があります。ターゲットの Windows CE プラットフォームで、OS の起動時に毎回、WinDriver をロードするには、**WinDriver¥redist¥Windows_Mobile_5_ARMV4I¥wdreg.exe** ユーティリティをターゲット PC の **Windows¥StartUp¥** ディレクトリにコピーします。
6. ターゲット CE コンピュータを再起動します。WinDriver CE カーネルは自動的にロードします。suspend / resume ではなく、システムの再起動を行ってください。(ターゲット CE コンピュータのリセットまたは電源ボタンを使用します。)
7. サンプル プログラムをコンパイルして起動し、WinDriver CE がロードされ、正常に動作するのを確認してください。

WINDOWS CE のインストールにおける注意事項:

Windows 10 / 8.1 / Server 2012 R2 / 8 / Server 2012 / 7 / Server 2008 R2 / Server 2008 ホスト PC での WinDriver のインストールでは、**WD_BASEDIR** 環境変数を定義します (インストール中に選択した WinDriver のディレクトリの場所を示しま

す)。WinDriver の DriverWizard でコードを生成する際には、この変数を使用します - 生成したコードを保存するデフォルトのディレクトリで、生成された project / make ファイルの include パスに使用します。

注意:
Windows 10 / 8.1 / Server 2012 R2 / 8 / Server 2012 / 7 / Server 2008 R2 / Server 2008 ツールキットを同じホスト PC にインストールする場合、インストール時に、Windows CE のインストールで設定した WD_BASEDIR 変数の値を上書きするのでご注意ください。

● WinDriver for Linux をインストールするには

必要なシステム:

1. Linux カーネル 2.6.x またはそれ以降と以下のプロセッサ アーキテクチャ:
   x86 32 ビット プロセッサ
   または
   x86 64 ビット AMD64 または Intel EM64T (x86_64) プロセッサ、ARM Cortex-A7、A9、A15、A53、A57
2. GCC コンパイラ
   注意: GCC コンパイラは起動中の Linux カーネルと同じバージョンをご使用ください。
3. C をサポートする 32 ビットまたは 64 ビット開発環境
4. 開発環境 PC: glibc2.14.x (またはそれ以降)
5. GUI WinDriver アプリケーション (DriverWizard、Debug Monitor など) の起動用に以下のライブラリが必要
   libstdc++.so.6
   libpng12.so.0
   libQtGui.so.4
   libQtCore.so.4
   libQtNetwork.so.4

インストール手順:

注意: Linux では、カーネル自身をコンパイルしたのと同じヘッダーファイルでカーネル モジュールをコンパイルする必要があります。WinDriver はカーネル モジュールをインストールするため、インストールj時に Linux カーネルのヘッダーファイルでコンパイルする必要があります。そのため、WinDriver をインストールする前に、Linux ソースコードおよび versions.h ファイルがご使用のマシンにインストールされていることを確認してください。

1. WinDriver を弊社ダウンロードサイトよりご希望の製品をクリックし**WD1421LN.tgz** ファイルをダウンロードします。
2. インストール用の任意のディレクトリに移動します。(例 /home/username/tmp)
   /$ **cd /home/username/tmp**
   注意: インストール ディレクトリへのパスにはスペースを含めないでください。
3. **WD1421LN.tgz** ファイルを解凍します。
   $ **tar -xvzf /mnt/cdrom/LINUX/WD1421LN.tgz**
4. WinDriver の **redist/** ディレクトリに移動します。(tar によって自動的に **WinDriver/** ディレクトリが作成されます)
   $ **cd WinDriver/redist/**
5. WinDriver をインストールします。
   - ~/WinDriver/redist$ **./configure –disable-usb-support**

     注意: configure スクリプトで、起動中のカーネル ベースの **makefile** を作成します。インストールした他のカーネル ソース ベースでも **configure** スクリプトにフラグ **--with-kernel-source=<path>** を付けて、configure スクリプトを起動できます。<path> はカーネル ソース ディレクトリへのフルパスです。デフォルトのカーネル ソース ディレクトリは **/usr/src/linux** です。
     Linux カーネル バージョンが 2.6.26 またはそれ以降の場合、**configure** は、**kbuild** を使用してカーネル モジュールをコンパイルする **makefile** を生成します。以前のバージョンの Linux で、**kbuild** を強制的に使用するには、**configure** に **--enable-kbuild** フラグを渡します。

     同じ VID/PID を持つ複数の USB デバイスに WinDriver を使用する場合は、**makefile.usb.kbuild** の **EXTRA_CFLAGS** 変数に **-DMULTIPLE_SAME_DEVICES** を追加します。
   - ~/WinDriver/redist$ **make**
   - スーパーユーザーになります。
     ~/WinDriver/redist$ **su**
   - ドライバをインストールします
     ~/WinDriver/redist$ **make install**
6. シンボリック リンクを作成し、DriverWizard GUI を簡単に起動できるようにします。
   ~/WinDriver$ **ln -s ~/WinDriver/wizard/wdwizard/ usr/bin/wdwizard**
   Ubuntu 18.04 などの新しい Linux 環境では、wdwizard の代わりに wdwizard_1804 を実行します。

7. wdwizard ファイルの read (読み取り) / execute (実行) の権限を設定し、他のユーザーがプログラムにアクセスできるようにします。
8. ユーザーおよびグループ ID を変更します。必要に応じて read (読み取り) / write (書き込み) の権限をデバイスファイル **/dev/windrvr1421** に設定し、ユーザーにデバイスを介してハードウェアにアクセスできるようにします。
   udev ファイル システムの Linux 2.6.x カーネルを使用している場合、**/etc/udev/permissions.d/50-udev.permissions** ファイルを編集して権限を変更します。たとえば、以下の行を追加して、read (読み取り) / write (書き込み) の権限を設定します。
   windrvr**1421**:root:root:0666
   または、**chmod** コマンドを使用します。**chmod /dev/windrvr1421 666**
9. **WD_BASEDIR** 環境変数を定義し、WinDriver ディレクトリの場所を示すように設定します (インストール中に選択した WinDriver のディレクトリの場所を示します)。この変数を WinDriver のサンプルおよび DriverWizard で生成されたコードの make およびソース ファイルで使用し、DriverWizard で生成されたプロジェクトを保存するデフォルトのディレクトリにも使用します。この変数を定義しない場合、WinDriver の makefile を使用して、サンプルおよび DriverWizard で生成されたコードをビルドする際に、定義するように要求されます。
   注意: WinDriver のインストール後、WinDriver のディレクトリの名前 / 場所を変更する場合、WD_BASEDIR 環境変数の値を変更し、新しい WinDriver のディレクトリの場所を指すように設定する必要があります。
10. スーパーユーザーモードを終了します。
    # exit
11. ハードウェアにアクセスを開始し、ドライバ コードを生成します。

ヒント: **WinDriver/util/wdreg** スクリプトを使用して、WinDriver のカーネル モジュールをロードします。システムの起動時に、自動的に WinDriver をロードするには、ターゲットの Linux のブートファイル (**/etc/rc.d/rc.local**) に以下の行を追加して wdreg を起動します:
<path to wdreg>/wdreg windrvr**1421**

登録版ユーザーの場合:

次の手順で、エクセルソフト株式会社から受け取ったライセンス コードを入力して WinDriver を登録します。

1. DriverWizard GUI を起動します。
   ~/WinDriver/wizard$ **./wdwizard**
2. [**Help**] メニューから [**Licensing Information**] を選択して [**Licensing Information**] ダイアログボックスを表示します。
3. 以前のバージョンのライセンス コードが登録されている場合、[**Deactivate license**] ボタンをクリックして、以前のバージョンのライセンス コードを解除します。
4. [**Please enter your license string**] 入力ボックスにエクセルソフト株式会社から受け取ったライセンス コードを入力して [**Activate license**] ボタンをクリックし、ライセンス コードを登録します。

ドライバの配布方法について

ドライバの配布方法について、以下を必ずお読みください。

注意:

- 以下の説明の **wdreg** と記述している個所を **wdreg_gui** に置き換えることができます。**wdreg_gui** は同じ機能ですが、コンソールモード メッセージの代わりに GUI メッセージが表示されます。
- WinDriver のカーネル モジュール (**windrvr1421.sys**) の名前を変更する場合、**windrvr1421** に関連する参照を対象のドライバ名に置き換え、**WinDriver¥redist** ディレクトリへの参照を変更したインストール ファイルを含むディレクトリのパスに置き換えてください。 たとえば、DriverWizard で生成したドライバ プロジェクトに名前を変更したドライバ ファイルを使用する場合、**WinDriver¥redist** への参照を生成された **xxx_installation¥redist** ディレクトリに置き換えてください (xxx は生成されたドライバ プロジェクトの名前です)。
- 新しい INF および / またはカタログ ファイルを作成した場合、オリジナルの WinDriver の INF ファイルおよび / または **wd1421.cat** ファイルへの参照を対象の新しいファイル名に置き換えてください。

作成したドライバを配布するには、いくつかのステップを行う必要があります。まずドライバをターゲットシステムにインストールする配布パッケージを作成します。ターゲットマシンにドライバをインストールします。このプロセスは **windrvr1421.sys** と **windrvr1421.inf** のインストール、デバイス用の INF ファイルのインストール (プラグ アンド プレイ ハードウェア用)、Kernel PlugIn ドライバ (作成した場合) をインストールします。最後に、WinDriver を使用して開発したハードウェア のコントロール アプリケーションをインストールして実行します。これらすべての手順は、**wdreg** ユーティリティを使用して実行できます。

ドライバの配布:

注意: 32 ビットと 64 ビット プラットフォームの両方をターゲットとしてドライバを配布する場合、各プラットフォームに対して、別々にドライバパッケージを用意する必要があります。各パッケージに必要なファイルは、それぞれのプラットフォームの WinDriver のインストール ディレクトリにあります。

ドライバの配布には以下のモジュールが必要です。

- ハードウェア コントロール アプリケーション / DLL
- **Windrvr1421.sys**
  ¥WinDriver¥redist ディレクトリ以下にあります
- **Windrvr1421.inf**
  ¥WinDriver¥redist ディレクトリ以下にあります
- **wd1421.cat**
  ¥WinDriver¥redist ディレクトリ以下にあります
- **wdapi1421.dll** (32 ビット バイナリを 32 ビット ターゲット プラットフォームへ配布する場合、または 64 ビット バイナリを 64 ビット プラットフォームへ配布する場合) または **wdapi1421_32.dll** (32 ビット バイナリを 64 ビット プラットフォームへ配布する場合)
  ¥WinDriver¥redist ディレクトリ以下にあります
- ユーザ アプリケーションが C#.NET/VB.NET アプリケーション/DLL の場合は、(32 ビット バイナリを 32 ビット ターゲット プラットフォームに配布する場合、または 64 ビット バイナリを 64 ビットに配布する場合) wdapi_dotnet1421.dll、または (32 ビット バイナリを 64 ビット プラットフォームに配布する場合) wdapi_dotnet1421_32.dll も追加する必要があります。
  ¥WinDriver¥lib ディレクトリ、または Visual Studio プロジェクトの Release ディレクトリ以下にあります。
- ユーザ アプリケーションが Java アプリケーション/DLL の場合は、(32 ビット バイナリを 32 ビット ターゲット プラットフォームに配布する場合、または 64 ビット バイナリを 64 ビット プラットフォームに配布する場合) wdapi_java1421.dll も追加する必要があります。パッケージに wdapi_java1421.jar も追加することをお勧めします。
  ¥WinDriver¥lib ディレクトリ (該当するプラットフォームに関連するサブディレクトリ) 以下にあります。
- **difxapi.dll** (wdreg.exe ユーティリティで必要です)
  ¥WinDriver¥util ディレクトリ以下にあります
- Driver Wizard を使用して作成した対象のデバイスの INF ファイル (USB デバイスや PCI および PCI-Express などのプラグ アンド プレイ デバイスで必要です)
- Kernel PlugIn ドライバ - **<KP ドライバ名>.sys** - (作成した場合、PCI 版で Kernel PlugIn 機能を使用した場合のみ)

注意: ドライバのインストールの際には、管理者権限を持ったユーザーで実行してください。

WinDriver のカーネル モジュールのインストール:

1. **Windrvr1421.sys**、**windrvr1421.inf** と **wd1421.cat** ファイルを同じディレクトリにコピーします。
   注意: wd1421.cat には、ドライバ認証のデジタル署名が含まれます。署名の妥当性を維持するには、**wd1421.cat** ファイルを **windrvr1421.inf** ファイルと同じインストール ディレクトリに配置する必要があります。カタログ ファイルと INF ファイルを異なるディレクトリに配置する場合、またはこれらのファイルを編集、またはカタログ ファイルの参照ファイル (**windrvr1421.sys** など) を編集する場合、以下のいずれかの処理を行います:
   - 新しいカタログ ファイルを作成し、このファイルを使用して再度ドライバの署名をします。
   - **windrvr1421.inf** ファイルの以下の行をコメント アウトまたは削除し、カタログ ファイルを配布しません:
     CatalogFile = wd1421.cat

   ただし、この場合、インストールではドライバのデジタル署名を使用しないので推奨いたしません。
2. **wdreg** ユーティリティを使用して、ターゲット コンピュータに WinDriver のカーネル モジュールをインストールします:
   ¥> **wdreg –inf <windrvr1421.inf のパス> install**

   たとえば、**windrvr1421.inf** と **windrvr1421.sys** がターゲット コンピュータの **d:¥MyDevice¥** ディレクトリにある場合、以下のようになります:
   ¥> **wdreg –inf d:¥MyDevice¥windrvr1421.inf install**

   wdreg の実行可能ファイルは、**¥WinDriver¥util** ディレクトリ以下にあります。

   注意:
   - **wdreg** は **difxapi.dll** DLL に依存します。
   - **wdreg** はインタラクティブなユーティリティです。失敗した場合には、問題を解決するためのメッセージが表示されます。コンピュータを再起動するようにメッセージが表示される場合もあります。

注意: ドライバの配布時に、新しいバージョンの **windrvr1421.sys** を Windows ドライバディレクトリ (**%windir%¥system32¥driver**) の古いバージョンのファイルで上書きしないようにご注意ください。インストール プログラムまたは INF ファイルでインストーラが自動的にタイムスタンプを比較して新しいバージョンを古いバージョンで上書きしないように設定することを推奨いたします。**windrvr1421.sys** ファイルは COPYFLG_NO_VERSION_DIALOG INF ディレクティブを使用します。これは、コピー先の既存のファイルがコピー元のファイルよりも新しい場合、コピー元のファイルでコピー先のファイルを上書きしないようにデザインされています。同様にCOPYFLG_OVERWRITE_OLDER_ONLY INF ディレクティブがあります。これは、コピー先のファイルがより新しいバージョンに取って代えられる場合のみ、コピー元のファイルをコピー先のディレクトリへコピーするようにデザインされています。ただし、これらの INF ディレクティブの両方は、デジタル署名されたドライバには適用されませんので、ご注意ください。Microsoft の INF CopyFiles Directive ドキュメントの説明のとおり (https://msdn.microsoft.com/en-us/library/ff546346%28v=vs.85%29.aspx)、ドライバパッケージがデジタル署名されている場合、Windows はパッケージ全体をインストールし、対象のコンピュータに既に他のバージョンが存在する場合でもパッケージのファイルを選択して除外しません。Jungo から提供する **windrvr1421.sys** ドライバはデジタル署名されています。

対象のデバイスの INF ファイルのインストール (プラグ アンド プレイ デバイスを windrvr1421.sys と動作するように登録):

対象の INF ファイルを自動的にインストールし Windows デバイス マネージャーを更新するには、以下のように **wdreg** を起動して、**install** コマンドを実行します:

¥> **wdreg -inf <対象のINF ファイルのパス> install**

preinstall コマンドを使用して、PC に接続していないデバイスの INF ファイルをプリインストールすることも可能です:

¥> **wdreg -inf <対象のINF ファイルのパス> preinstall**

注意: ERROR_FILE_NOT_FOUND エラーでインストールに失敗する場合、Windows のレジストリに RunOnce キーが HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Windows¥CurrentVersion にあることを確認してください。INF ファイルを使用してドライバを正してくインストールするために、Windows Plug-and-Play で必要なレジストリ キーです。

ターゲット コンピュータに Kernel PlugIn ドライバをインストール:

Kernel PlugIn ドライバを作成した場合、以下の手順でドライバをインストールします:

1. Kernel PlugIn ドライバ (**<KP ドライバ名>.SYS**) をターゲット コンピュータの Windows ドライバ ディレクトリにコピーします (**%windir%¥system32¥drivers**)。
2. **wdreg** ユーティリティを使用して Windows の起動時にロードするデバイス ドライバ リストに Kernel PlugIn ドライバを追加します。
   以下のようにコマンドを実行します:
   ¥> **wdreg –name <ドライバ名、.sys 拡張子は必要ありません> install**

wdapi1421.dll をインストール:

作成した対象のハードウェアのコントロール アプリケーション / DLL で **wdapi1421.dll** を使用する場合 (WinDriver のサンプル プロジェクトおよび DriverWizard で生成されたプロジェクトは使用します)、**wdapi1421.dll** をターゲット コンピュータの **%windir%¥system32** ディレクトリにコピーします。
32 ビット アプリケーション / DLL をターゲットの 64 ビット プラットフォームに配布する場合、**wdapi1421_32.dll** を **wdapi1421.dll** に名前を変更し、このファイルをターゲットの **%windir%¥sysWOW64** ディレクトリにコピーしてください。

注意: 64 ビットのプログラムをインストールする 32 ビットのインストール プログラムを作成し、64 ビットの **wdapi1421.dll** を **%windir%¥system32** ディレクトリにコピーすると、ファイルは実際には 32 ビットの **%windir%¥sysWOW64** ディレクトリにコピーされます。これは、Windows x64 プラットフォームでは 64 ビットのディレクトリを参照する 32 ビットのコマンドを、32 ビットのディレクトリを参照するように変換するためです。これを回避するには、**WinDriver¥redist** ディレクトリにある **system64.exe** プログラムを使用し、64 ビットのコマンドを使用してインストールを実行します。

作成した対象のハードウェアのコントロール アプリケーション / DLL をインストール:

作成した対象のハードウェアのコントロール アプリケーション / DLL をターゲット コンピュータにコピーし、起動します。

s

ライセンス コードの申請方法

ライセンス コードの申請方法に関しては、別紙の「ユーザー登録のご案内とライセンス コードの申請方法について」を参照して

ください。

テクニカル ドキュメント

下記 Web サイトから、WinDriver のテクニカル ドキュメント データベースも利用可能です:
https://www.xlsoft.com/jp/products/windriver/support/tech_docs_indexes/main_index.html
テクニカル ドキュメント データベースには、WinDriver の機能、ユーティリティ、API とその他の使用方法についての詳細な説明、一般的なトラブルシューティング、役立つヒント、およびよくある質問が含まれます。

更新履歴

V14.21 の変更点

新機能:

- コード生成:C 言語プロジェクトの CMake プロジェクト生成のサポートを追加。 (現時点では、Kernel PlugIn サポートなし)

修正:

- Windows:名前を変更したドライバーを使用する際のバグを修正。 (v14.2.0 以降の問題)
- C サンプル:不要なインクルードを削除。
- WD_PciConfigDump:Kernel PlugIn から呼び出す際のバグを修正。 (v12.8.0 以降の問題)
- さまざまなバグを修正。

V14.20 の変更点

新機能:

- コード生成:Python 3 用のコード生成のサポートを追加。
- usb_diag (C、Python、Java): USB 3 および USB 3.1 のサポートをテストおよび検証済み。速度テストの追加、USB 3 のサポート用にデフォルトの転送サイズの変更。
- Windows:WinDriver カーネルモジュールで WDF フレームワークを使用。

強化:

- Linux:最新の Linux カーネル バージョン v5.3.6 のサポートを追加。
- Linux、USB:windrvr_usb モジュールにコンパイル フラグを追加し、WinDriver で同じ VID/PID を持つ複数の USB デバイスのオープンのサポートを追加。

修正:

- 32 on 64 bit Interrupts:dwEnableIntType の誤った値を修正。
- ウィザード:Kernel PlugIn のコード生成、License ウィンドウ、Check for updates window などさまざまな Driver Wizard のバグを修正 。
- Windows、WDC_DMAContigBufLock:ビット マスク メカニズム の修正。 (v14.1.0 以降の問題)
- C#/ VB.NET API:関数 WDC_Version()、WDC_GetWDHandle()、WDC_GetDevContext()、WDC_PciGetHeaderType() を修正。(v12.7.0 以降の問題)
- Linux ARM:ドライバーのコンパイル エラーを修正。(v14.1.0 以降の問題)

終息:

- ISAPnP:ISAPnP のサポートを削除。
- 高レベル API:終息した関数 GetPageCount()、InterruptThreadEnable()、InterruptThreadDisable()、event_register()、event_unregister() を削除。
- 低レベル API:終息したレガシーバージョンの WD_DMALock()、WD_IntEnable() IOCTL を削除。

V14.11 の変更点

修正:

- ライセンスコードの発行に関する問題を修正。(v14.10 以降)
- 古いバージョンの WinDriver から提供されていた IOCTLS を削除。

V14.10 の変更点

新機能:

- Windows: Visual Studio 2019 のサポートを追加。
- Linux ARM および ARM64: 自動インストール スクリプトを追加。

強化:

- Linux: 最新の Linux カーネル バージョン v5.1.13 のサポートを追加。
- Windows、WDC_DMAContigBufLock: バッファを 4GB までロック可能。ビットマスクを追加可能 (v14.0 Linux 版で追加した機能と同じ)。
- usb_diag (C、Python、Java): デバイスのプロパティを表示するメニュー オプションを追加。
- usb_diag (C、Python、Java): プロセスあたり最大 1 つのデバイスを許可するサンプルコードを追加。
- Linux、WDU_GetDeviceRegistryProperty: Linux で取得できるデバイス プロパティを追加。

修正:

- まざまなセキュリティを修正。

V14.00 の変更点

新機能:

- PCI および USB 向けに JNI DLL および Java プログラミング言語のサンプルを追加。
- Java プログラミング言語でのコード生成を追加。
- Linux ARM および ARM64: 以下の開発ボードのサポートを追加。

  NVidia Jetson TK1, NVidia Drive PX2, Raspberry Pi 2, 3, 3b+, Boundary Sli.imx6 (sabrelite)

強化:

- Linux: 最新の Linux カーネル バージョン v4.19.32 のサポートを追加。
- Linux: 複数の MSI 割り込みをマップする機能を追加。
- Linux、WDC_DMAContigBufLock: アドレス マスクを入力する機能を追加。
- get_os_type() API 関数のリファクタリング

修正:

- Kernel Plugin: コンパイルの問題を修正。
- C#, Python: 生成コードの細かいバグ修正。

- C#.NET/VB.NET: wdapi_dotnet1290_32.dll の API のバグを修正 (v12.8.1 以降の問題)。

V12.90 の変更点

新機能:

- IPC: プロセス間の割り込み共有を追加。
- Linux: SR-IOV サポートの追加。

強化:

- DriverWizard: CSV ファイルからレジスタのインポート機能を追加。
- DriverWizard、Windows: vcxproj ファイルを作成しカスタマイズされたバージョン情報とともにドライバをビルド (古い ddk_make が生成したメイクファイルの代わりに)。
- Linux: カーネル v4.19.1 のサポートを追加。

修正:

- Windows: PCI スキャンと MSI 割り込みで発生するクラッシュの問題を修正。
- Windows 10 IoT Core: INF をインストールしないで PCI デバイスをスキャンする機能を追加。

V12.81 の変更点

強化:

- Qt: Qt5 による WinDriver GUI。Linux では、DriverWizard と Debug Monitor で Qt 4 および Qt5 の両方のバージョンで後方互換性があります。

修正:

- 複数のセキュリティに関する問題を修正。

V12.80 の変更点

新機能:

- PCI と USB 向けに Python プログラミング言語のサンプルを追加。
- Python プログラミング言語でのコード生成を追加。

強化:

- Linux: 最新の Linux カーネル バージョン v4.17.3 のサポートを追加。
- C#.NET PCI サンプル: S/G と連続 DMA バッファをマップする機能を追加。
- Windows: Windows API ライブラリに新しい関数を追加: check_secureBoot_enabled()
- Windows、DriverWizard: INF ファイルのないデバイスを開こうとした際に警告を表示する機能を追加。

修正:

- Windows 10 IoT Core: ARM アーキテクチャのサンプルを修正 (v12.7.0 以降の問題)。
- Windows 32 ビット: 事前にコンパイルされたサンプルをインストレーション パッケージに再度追加 (v12.7.0 以降の問題)。
- Windows 32 ビット: 名前を変更したドライバのコンパイル時に発生するエラーを修正 (v12.5.0 以降の問題)
- C#.NET/VB.NET: wdapi_dotnet1280.dll の API のバグを修正 (v12.7.0 以降の問題)。

V12.70 の変更点

新機能:

- Windows: 開発時の INF ファイルおよびカーネルプラグインのアンインストールおよびインストールを容易化する wdreg_frontend GUI を追加。
- Windows: BSOD が発生した際に、Debug Monitor ユーティリティでカーネルデバッグメッセージの保存および検索ができる機能を追加。
- Windows 10 IoT Core: SSH ターミナル を使用して動作するサンプルを追加 (ベータ版)。
  注意: Windows 10 IoT Core に配布する場合は、別途ライセンスの購入が必要になります。

強化:

- Linux: 最新の Linux カーネル バージョン v4.15.3 のサポートを追加。
- Debug Monitor: アプリケーションを再起動することなく、デバッグしたドライバを変更できるように改良。
- Windows: Visual Studio 2015 および 2017 で C# / VB.NET をサポート。wdapi_dotnet DLL のソースコードをマネージド C++ シンタックスから新しい C++ / CLI シンタックスへ変更。
- Windows: サンプルディレクトリ の (PROJECT_DIR)/(PLATFORM)/(IDE) サブディレクトリに C# / VB.NET ソリューション ファイルを追加 (C のプロジェクトと同様)。

修正:

- Windows: .NET のサンプルをコンパイル時に発生する問題を修正 (v12.2.0 以降の問題)。
- DriverWizard: レジスタにバグがある場合に "Listen to Interrupts" で発生する問題を修正 (v12.2.0 以降の問題)。
- Windows 32 ビット: 名前を変更したドライバのコンパイル時に発生するエラーを修正 (v12.5.0 以降の問題)
- Windows: 複数のセキュリティに関する問題を修正。

V12.60 の変更点

新機能:

- PCI: 新しい関数を WinDriver API ライブラリへ追加:
  WDC_PciGetExpressOffset, WDC_PciGetHeaderType, PciConfRegData2Str, PciExpressConfRegData2Str.

強化:

- DriverWizard: DriverWizrd で生成されたコードのディレクトリ構成を強化。
- DriverWizard: ライセンス レジストレーション ウィンドウおよびライセンス リクエスト フォームの修正と強化。
- PCI: Microsoft Visual Studio 2017 用に Kernel Plugin のコード生成をサポート。
- Linux: 最新の Linux カーネル バージョン v4.14.5 のサポートを追加。

修正:

- PCI: XDMA GUI ユーティリティ上で登録したライセンスを認識。
- Windows: セキュリティに関する問題を修正。

V1251 の変更点

修正:

- Windows: IOCTL に関するセキュリティの脆弱性を修正。

V1250 の変更点

新機能:

- Windows 10 IoT Core x86, x64 および ARM のサポートを追加。

注意: WinDriver for Windows 10 IoT Core はベータバージョンとして提供されます。

- PCI: Xilinx XDMA デバイス向けに実装した GUI ユーティリティを追加。

強化:

- Linux: カーネル v4.12.8 のサポートを追加。
- PCI: DriverWizard で Xilinx XDMA デザイン用にカスタマイズしたコードを生成するオプションを追加。
- DriverWizard のライセンス情報の画面 (Licensing Information) を強化。
- PCI: DriverWizard で "Power Management" と "Plug and Play" の選択セクションを削除してコード生成を簡素化。"Power Management" と "Plug and Play" は DriverWizard で生成したコードに常に含まれます。
- DriverWizard で DMA バッファを事前割り当てするオプションを INF 生成ダイアログに追加。
- WinDriver の Degug MOnitor GUI ユーティリティに 'Find' (検索) 機能を追加。
- 'wddebug' コマンドライン ユーティリティにログをファイルへ保存するオプションを追加。
- PCI: WDC_ScanCapsBySlot()、WDC_GetPciExpressGen() と WDC_GetPciExpressGenBySlot() 関数を WinDriver API Library へ追加。
- DriverWizard の多くの強化と修正。

修正:

- Windows PCI: Windows 64 ビット OS で 32 ビット .NET アプリケーションを実行する際に発生する out of bounds exception (範囲外例外) の問題を修正。
- Linux PCI: 32 ビット ユーザーモード アプリケーションから 64 ビット カーネル プラグインを開く際に発生する OS クラッシュの問題を修正。
- PCI: DriverWizard で 'Add access register' を選択した際に 'Offset' に表示されるゴミを削除。
- USB: DriverWizard で USB デバイスの検出の問題を修正。

終息:

- Windows USB: DriverWizard の Ellisys USB アナライザーへの参照を削除。
- 'speaker' および 'speaker_gui' サンプルの削除。

V12.40 の変更点

新機能:

- Windows: Microsoft Visual Studio 2017 のサポートを追加。

強化:

- PCI: 複数の DMA バッファを事前に割り当てる可能性を追加。
- PCI: DriverWizard で生成されるコードに DMA の API の使用例を追加。
- PCI: "WinDriver for Server" ライセンスを使用する場合、スキャンする PCI デバイスの最大数を 100 から 256 に増加。通常の WinDriver のライセンスでは、スキャンする PCI デバイスの最大数は 100。
- Linux: カーネル v4.10.8 のサポートを追加。
- Linux: WinDriver と Kernel PlugIn のビルドで使用するコンフィグレーション スクリプトを簡素化。

修正:

- Windows: 独自の詳細でリビルドしたドライバ用に windrvrXXXX.lib を修正。(v12.30 からの問題)
- PCI Express: XDMA のサンプルの問題を修正。(v12.30 からの問題)
- Linux: カーネル 4.8 およびそれ以降で、CONFIG_HARDENED_USERCOPY を定義した場合に発生するカーネルのバグを修正。
- Linux: __stack_chk_fail シンボルを定義していない場合に発生する Kernel PlugIn モジュールのロードの問題を修正。

終息:

- Windows: Visual Studio 2005 および 2008 のサポートを終了。
- Windows: .NET 1.1 および 2.0 のサポートを終了。
- PCI: WDC_ID_U and WDC_SLOT_U union (共用体) の定義を削除。

V12.30 の変更点

新機能:

- PCI: Xilinx® DMA Subsystem for PCI Express ® (XDMA) のサンプルを追加。
- Linux: カーネル v4.8.2 のサポートを追加。

強化:

- PCI: WDC_IntType2Str() 関数をハイレベル API へ追加。

- PCI: WDC_GET_ADDR_SPACE_SIZE ヘルパーマクロをハイレベル API へ を追加。
- PCI: Kernel Plugin のサンプルをリファクタリング
- Linux PCI: MSI-X を有効にする場合、あらかじめ定義された値ではなく、デバイスがサポートする割り込みの最大数を有効にする。
- Windows .NET: .NET API に WDC_DMAReservedBufLock を追加。
- Linux: カーネルソースを明示的に指定せずに Ubuntu に自動インストールできる configure スクリプトを改良。
- レジストレーション コードを取得するコマンドライン ユーティリティを追加。
- Linux 32bit: WinDriver パッケージのフットプリントを小さくして、64 ビットバージョンと同様に改良。

修正:

- Windows: 管理者として実行するときに生成された xxx_install.bat が wdreg.exe を検出していることを確認するように修正。
- Windows .NET: wdapi_dotnet 32ビット DLL をビルドする Visual Studio 2013 ソリューションのプロジェクトのプロパティを修正。(v12.10 からの問題)
- Windows .NET: wdapi_dotnetXXX.dll のプロジェクト名を DLL 名と同様に設定するように修正。(wdapi_dotnetXXX.dll 自体をビルドしているユーザーの場合)
- Windows PCI: PCI デバイスを開いたときに .NET の例外エラーが発生する問題を修正。
- Linux: インストールスクリプトの代替モジュールパスを修正。
- Linux: ドライバの名前を変更せずに、2 種類の WinDriver バージョンのインストールを有効化。(v11.9.0 からの問題)
- Linux: IORESOURCE_MEM_64. を定義していない古いバージョンの Linux カーネルで発生するコンパイル エラーを修正。(v12.20 からの問題)
- Linux PCI: DMA バッファを割り当てる際のカーネル クラッシュの問題を修正。(v12.10 からの問題)
- Linux PCI: DriverWizard で生成した xxx_lib.c をコンパイルしない Makefile を生成する可能性のあるDriverWizard のバグを修正。(v11.5.0 からの問題)
- Linux: redist フォルダで 'make clean' を実行すると、LINUX.x.x.x ディレクトリが削除される問題を修正。

終息:

- PCMCIA: PCMCIA デバイスのサポートを終了
- Windows: Windows XP と Windows Vista の OS 固有サポートを終了 (Microsoft によるサポートの提供が終了のため)。

V12.21 の変更点

新機能:

- Linux: カーネル v4.6.2 のサポートを追加。

修正:

- PCI on Windows: レベル センシティブな割り込みを使用した際に発生する BSOD を修正 (v12.20 からの問題)。
- Linux: 生成されるモジュールのコンパイルの問題を修正 (v12.20 からの問題)。
- Windows 32-bit: 長いライセンス コードを入力する際に発生する BSOD を修正 (v12.20 からの問題)。
- Windows XP 32-bit: ドライバーのロードの問題を修正 (v12.20 からの問題)。
- Windows .NET: .NET API の WD_LICENSE_LENGTH の正しい値を設定 (v12.20 からの問題)。
- Windows .NET: .NET で RegistersForm.IsLegalDirection() 関数の生成されるコードの不正なロジックを修正。
- Linux: linux_wrappers.c で発生する可能性のあるコンパイルの警告の問題を修正。
- Windows: Window 7 およびそれ以前でスタート メニューのショートカットをリファクタリング。
- Windows CE: 冗長なログ メッセージを表示するデバッグ モードで WDAPI ライブラリのビルドの問題を修正 (v11.90 からの問題)。

V12.20 の変更点

新機能:

- 予約した DMA 連続メモリーを要求する API を追加 (終息予定の "WDC_IsaDeviceOpen" のアプローチと置き換え)。

強化:

- Pci-diag Kernel Plugin のサンプルをリファクタリング。

修正:

- Linux: "__stack_chk_fail_" ルーティンに関するコンパイル エラーが発生する可能性がある問題を修正 (v12.00 からの問題)。

- PCI on Windows: (INF ファイルで) インストールしたドライバが 32-bit 以上の PCI BAR アドレスを構文解析する際に発生する問題を修正 (v6.00 からの問題)。

V12.10 の変更点

新機能:

- IPC - WinDriver のインフラを使用したユーザー プロセス間の IPC (Inter Process Communication) を追加。
- バッファ共有 - 複数のユーザーモード プロセス間の DMA とカーネル バッファ共有を追加。
- Linux: v4.3.3 まで最新のカーネル バージョンのサポートを追加。

強化:

- Linux & Windows: 共有バッファ (ユーザーモードのドライバと Kernel PlugIn 間) 機能を再記述。これによってよりリソースを使いやすく (非連続な割り当てが可能)、より成功しやすくなりました。API も同様に以下のように変更しました: WDC_SharedBufferAlloc/WDC_SharedBufferFree の名前を WDS_SharedBufferAlloc/WDS_SharedBufferFree へ変更しました。
- PCI: DMA メモリ割り当てを pci_diag サンプル コードへ追加。
- Windows: MS Visual Studio 2013 & 2015 用のサンプルとコード生成のサポートを追加。

修正:

- Windows: WinDriver Kernel Plugin の WDC_ADDR_DESC のアライメントの問題を修正 (v11.80 からの問題)。
- Linux: Linux バージョン >= 4 で Kernel PlugIn のコンパイラの問題を修正 (v12.00 からの問題)。
- PCI on Windows: 連続 DMA ロックの後で事前割り当てしたバッファのカーネル アドレスが NULL になる問題を修正 (v6.00 からの問題)。
- Linux: glibc の以前のバージョンでサンプルとドライバを実行可能 (DriverWizard はまだ実行できません) (v11.80 からの問題)。
- Linux 64-bit: DriverWizard が起動時に発生する可能性があるクラッシュの問題を修正 (v11.80 からの問題)。
- Linux 32-bit: Kernel PlugIn のコンパイル エラーを修正。
- WinCE2013 USB: アイソクロナス パイプからの不正なデータ読み込みの問題を修正 (v11.70 からの問題)。

V12.00 の変更点

新機能:

- Windows: Windows 10 (Public Release) のサポートを追加。

修正:

- USB on Windows: Microsoft の Driver Verifier ユーティリティで不正に認識される可能性がある WDU_ResetPipe() の問題を修正。

終息:

- Windows: Windows Server 2003 の OS 固有のサポートを終了 (Microsoft によるサポートの提供が終了のため)。

V11.90 の変更点

新機能:

- Windows: Windows 10 Technical Preview build 10162 のサポートを追加。

強化:

- WinDriver のドライバ モジュールの名前を変更し、WinDriver のバージョン番号を追加 - **windrvr.sys/ko/dll** (**windrvr1190.sys** など)。
- Linux: 最新のカーネル バージョン (v4.1.1 まで) のサポートを追加。
- Windows: WinDriver のドライバ モジュール (**windrvr<version>.sys**、v11.90 では **windrvr1190.sys**) とサンプルの Kernel PlugIn ドライバ (**kp_pci.sys**) のドライバの署名で新しい SHA-2 ハッシュ アルゴリズムを使用するように変更。**注意:** Windows 7 および Windows Server 2008 では新しいドライバのインストールに Microsoft Security Advisory *2949927* へアップデートが必要です。ただし、最新の Microsoft Security Advisory は *2949927* から *3033929* となっていますので、*3033929* へアップデートしてください。
- DriverWizard: ライセンス リクエストのデータ申請フォームを追加し、WinDriver のオーダー処理の完了を簡素化。
- DriverWizard: Help メニューのオプションを追加し、WinDriver のバージョン アップデートを確認。

修正:

- Windows CE 2013: OS のカーネル デバッガへの Debug Moinitor のデバッグ メッセージのリダイレクトを修正 (v11.70 からの問題)。
- Windows .NET: .NET Windows API ヘッダ ファイルの WinDriver のライセンス文字列の長さを修正。
- Windows CE: set-configuration の呼び出しの後で、alternate setting 0 で 冗長な redundant set-interface の呼び出しを削除。

V11.80 の変更点

新機能:

- PCI: PCI の機能 (特定の PCI Express の機能を含む) をスキャンする API を追加。
  wdc_diag_lib と pci_diag サンプル コードに関連する機能を追加。

強化:

- PCI: DriverWizard の PCI コンフィグレーション スペースのダイアログを改良:
  - 実アドレスでレジスタを並び替え。
  - CardBus (PCI Header type 2) レジスタ、PCI Express レジスタ、ベーシックと拡張 (PCI Express) PCI の機能レジスタを追加。
- PCI: WinDriver API を変更し 64-bit アドレスのサポートを改良 (WD_ITEMS、WD_TRANSFER、 WDC_ADDR_DESC 構造体の変更を含む)。
- DriverWizard: その他、ウィザードの GUI を改良 (新しい License Request フォームやツールバーの変更を含む)。

修正:

- Linux PowerPC (PPC) 64-bit: WinDriver のライセンス ストリングの登録を修正 (v11.60 からの問題)。

V11.70 の変更点

新機能:

- Windows CE: x86 と ARM プラットフォームで Windows Embedded Compact 2013 (WEC2013) のサポートを追加。

強化:

- Linux: カーネル v3.17.0 のサポートを追加。
- USB: C の USB 診断サンプル コードにデバイスのシリアル番号のクエリーのサポートを追加:
  - USB 診断ライブラリ (**usb_diag_lib**) – 新しい PrintDeviceSerialNumberByHandle() 関数を追加。
  - USB 診断アプリケーション (**usb_diag.c**) – 新しい *Display device serial number menu* オプションを追加。
- Debug Monitor: ログ メッセージにタイムスタンプを追加するオプションを **wddebug** と **wddebug_gui** に追加、ログ ファイルを自動保存しカスタム メッセージを追加できる GUI オプションを **wddebug_gui** に追加。

修正:

- USB on Windows CE: 対象のショート パケットによって内部バッファがフルになりストリーム処理が停止する可能性がある問題を修正 (v10.01 からの問題)。
- USB on Windows: 短いタイムアウトで転送した際に発生する可能性がある OS クラッシュの問題を修正 (v9.00 からの問題)。

V11.60 の変更点

強化:

- Linux: カーネル 3.14.4 のサポートを追加。
- WinDriver 64-bit (.NET): 64-bit プラットフォームで、DriverWizard の 32-bit .NET アプリケーションのコード生成のサポートを追加。
- ISA on Windows (.NET): 物理 64-bit メモリ アドレスのアクセス用に、WDC_SLEEP_NON_BUSY item type を .NET API へ追加。

修正:

- PCI on Linux 64-bit: 64-bit Kernel PlugIn ドライバで 32-bit アプリケーションを起動した際に発生するメモリ破損の問題を修正 (v11.30 からの問題)。
- Windows: 稀に発生するメモリ リークの問題を修正。
- PCI on Windows 64-bit: 64-bit 物理アドレスのマッピングの問題を修正 (v10.40 からの問題)
- PCI: WD_INTERRUPT dwLost フィールドの lost-interrupts (取りこぼしの割り込み) カウントの戻り値の同期の問題を修正。

- USB on Windows: single-blocking 転送用に短いタイムアウト (通常、1秒以内) を使用した際に発生する OS クラッシュの問題を修正。
- Kernel PlugIn PCI on Linux: kp_interlocked_exchange() の戻り値を修正し、新しい値ではなく古い Kernel PlugIn の interlocked カウンターの値を返す。

V11.50 の変更点

強化:

- Linux: Linux: カーネル 3.13 のサポートを追加。
- PCI on Windows: DriverWizard で生成したライブラリ コードを MS Visual Studio を使用して DLL へのビルドのサポートを追加。
- Windows: MS Visual Studio 2013 で DriverWizard で生成したコードをサポート。
- Busy Sleep (WDC_Sleep() with the WDC_SLEEP_NON_BUSY option, or WD_Sleep() with the SLEEP_NON_BUSY option) 用の SW の制限である最小値 17,000 マイクロ秒の Sleep-Period を解除。
- Windows: Windows 8.1、Server 2012 のサポートを追加。

修正:

- PCI on Linux: PCI デバイスを閉じる際の DMA バッファの解放の問題を修正 (v11.40 からの問題)。
- Linux PowerPC (PPC): 連続 DMA バッファの割り当ての問題を修正 (v10.30 からの問題)。
- Linux: LINUX_user_page_list_get() が失敗した際に、OS がクラッシュする問題を修正。
- PCI on Linux: Ctrl+C を使用してドライバ アプリケーションを終了する際に、OS がクラッシュする問題を修正 (v11.10 からの問題)。
- USB on Windows: DeviceAttach の hMutex で NullPointerException をハンドルするように DriverWizard で生成した C# のコードを修正 (v8.00 からの問題)。
- Linux: Linux 2.6 またはそれ以降で、_CONFIG_SWIOTLB で設定した DMA バッファを案ロックする際に、OS がクラッシュする問題を修正 (v9.10 からの問題)。

終息:

- Windows CE 4.x/5.x: Microsoft Platform Builder 用の WinDriver のコンポーネントの提供を終了。
- Linux: Debian および RPM (Red Hat Package Manager) 用の WinDriver のインストール パッケージの提供を終了。
- USB: Cypress EZ-USB 用のサンプル コードの提供を終了。

V11.40 の変更点

強化:

- Linux: カーネル 3.10.5 のサポートを追加。
- Windows: Windows 8.1 Preview 版のサポートを追加。
- Windows: パフォーマンスの改良を含む USB 3.0 サポートの動作検証。
  注意: 本バージョンでは、USB 3.0 の機能をフル サポートしておりません。USB 3.0 デバイスを USB 2.0 互換での動作を検証いたしました。
- PCI on Windows: INF ファイルの設定を使用して、事前に割り当てた双方向の連続 DMA バッファ (DMA_TO_FROM_DEVICE) のサポートを追加。
- PCI 割り込み: 同じ非共有 (fNotSharable=1) のレベル センシティブな割り込み (同じ IRQ 番号) で、複数の PCI デバイスの同時オープンのサポートを追加。

修正:

- PCI: Kernel PlugIn のハンドルで、複数の PCI デバイスをオープン/クローズする際に発生する可能性があるクラッシュの問題を修正。
- Windows: WinDriver のドライバー モジュール (**windrvr6.sys**) の名前を変更したバージョンをビルドする際に使用する DriverWizard で生成された sources ファイルの問題を修正 (v9.00 からの問題)。
- Windows 64-bit (PCI): 32-bit (x86) 版の DriverWizard で生成した Kernel PlugIn ドライバのプロジェクトを 64-bit Windows プラットフォームでコンパイルする際の問題を修正 (v10.30 からの問題)。
- PCI: Kernel PlugIn のハンドルで、複数の PCI デバイスをオープン/クローズする際に発生する可能性があるクラッシュの問題を修正。
- Linux: CTRL+C を使用してアプリケーションを終了する際に発生する OS クラッシュの問題を修正。
- PCI interrupts on Linux: デバイスがレベル センシティブな割り込みをサポートしない場合、MSIs (Message-Signaled Interrupts) または MSI-Xs (Extended Message-Signaled Interrupts) のデバイスのサポートを確認 (v9.10 からの問題)。
- PCI on Linux: MSIs (Message-Signaled Interrupts) をサポートする PCI デバイスの MSIs の最大数を有効にする際の問題を修正 (v10.21 からの問題)。
- Linux PCI: DMA ハンドルをクローズせずにアプリケーションを終了する際に発生する OS クラッシュの問題を修正。
- Linux: WinDriver のカーネル モジュールをコンパイルする際に発生する OS_unlock_user_buff() Warning の問題を修正 (v11.10 からの問題)。

V11.30 の変更点

新機能:

- PCI Express: DriverWizard で Altera Qsys と Xilinx BMD design 用にカスタマイズしたコードを生成する機能を追加。

強化:

- Linux カーネル 3.8.5 のサポートを追加。
- PCI: **bits.h** と **pci_regs.h** ヘッダ ファイルを WinDriver の **samples¥shared¥** ディレクトリから **include¥** ディレクトリへ移動。

修正:

- .NET (Windows): WinDriver のサンプルおよび DriverWizard で生成された C# と Visual Basic .NET のコードで発生し得るビルド エラーの問題を修正。
- USB: ブロッキング インプット ストリームで要求したよりも稀に少ないデータが返ってくる問題を修正。
- ISA on Windows: DriverWizard で生成した **<driver_name>_install.bat** のインストール スクリプトの問題を修正。

V11.20 の変更点

新機能:

- Windows: MS Visual Studio 2012 および Windows Driver Kit (WDK) 8 のサポートを追加。

強化:

- Linux カーネル 3.7.1 のサポートを追加。
- PCI on Linux: デバイスでサポートするタイプの中から、特定のタイプ (レベル センシティブ/MSI/MSIx) の割り込みの有効のサポートを追加。
- PCI Express: 成功または失敗の戻り値を返すように、Xilinx BMD design (**bmd_lib**) と Altera Qsys design (**qsys_lib**) のサンプル ライブラリの関数を編集。

修正:

- Windows: Microsoft 社の Windows Hardware Certification Kit (HCK) 2.0 (part of the latest WHQL certification) で WinDriver をテスト時に発生する OS クラッシュ (BSOD) の問題を修正。
- USB: Power-Change Callback (**WDU_POWER_CHANGED_CALLBACK**) の実行を修正。
- Windows: WinDriver カーネル ドライバ (**windrvr6.sys**) のデジタル署名を更新して、Windows 8 をサポート (v11.00 からの問題)。
- PCI on Linux: MSI (Message-Signaled Interrupts) を有効にする際に IRQ 番号を更新する問題を修正 (v9.10 からの問題)。
- Windows: release mode で WinDriver .NET API DLL (**wdapi_dotnet<version>.dll**) をビルドする際に発生するコンパイル エラーを修正 (v7.01 からの問題)。

終息:

- MS Visual Studio 5/6/2003、Borland C++ Builder、Visual Basic 6.0、および Delphi のサポートを削除。

V11.10 の変更点

強化:

- PCI on Windows CE: 新しい WD_ITEM_CE_MAP_VIRTUAL フラグを WD_ITEM_OPTIONS enumeration へ追加 (デフォルトの静的なカーネル アドレスのマッピングとは対照的に、非静的な仮想システムのアドレスへの物理メモリのカーネル マッピングの実行用)。
- PCI Express: Altera Qsys デザインの PCI Express カードに対する拡張サポートを追加 – **WinDriver/altera/qsys_design**。
  新しいサンプルは Altera Stratix IV GX development kit で動作検証しました。
- Linux: IOCTLs のパフォーマンスを改良。
- PCI on Windows: メモリー フラグメンテーションによって生じる割り当てエラーを防ぐために、(Windows の起動時に) ドライバ ロード処理の一部として事前に割り当てた連続 DMA バッファへオプションを追加。

修正:

- Linux: Ubuntu 11.10 など **/usr/lib64** ディレクトリのない 64-bit ディストリビューション用に **/usr/lib** へ **/usr/lib64** のシンボリック リンクを追加 (v11.00 からの問題)。
- Linux: 以前の WinDriver で発生した **conftest.s1** ファイルに関連する **sed** のインストール エラーを解消するために、複数の **modpost** の事象に対する処理を追加。
- Linux: Linux 3.x カーネルで発生した **wdreg** インストール スクリプトの **auto** オプションの問題を修正 (v11.00 からの問題)。

- Linux PowerPC (PPC): multi-domain PCI のスキャン機能を修正して、すべてのドメインを表示 (v11.00 からの問題)。
- PCI: 100 以上の PCI カードが登録されている場合、登録されているデバイスの PCI スキャン (**WDC_PciScanRegisteredDevices() / WD_PciScanCards() with the WD_PCI_SCAN_REGISTERED** オプション) に対するバッファー オーバーランの問題を修正 (v6.02 からの問題)。
- PCI: 64-bit ホストから、4GB アドレス範囲より上位の物理メモリへマップされたページを含むバッファを持つ 32-bit デバイスへ Scatter/Gather DMA を実行した際に発生する可能性のあるデータ破損の問題を修正。
- PCI: 稀に発生する **WDC_xxxDeviceOpen()** 失敗の問題を修正。
- PCI on Windows: Scatter/Gather DMA の割り当て中に起こり得る競合状態の問題を修正 (v9.21 からの問題)。
- Windows: USB アプリケーションが存在する場合に起こり得る OS クラッシュ (BSOD) の問題を修正。
- PCI on 64-bit Windows: DriverWizard で生成したユーザーモードの 64-bit Visual Studio のプロジェクトから **-DWINNT** フラグを削除。ユーザーモードのアプリケーションから直接 **WDC_xxxDeviceOpen()** 関数 (**(PWDC_DEVICE)(*phDev)->pCtx**) によって返されたデバイス コンテキストにアクセスする際に、以前は、このフラグを使用して不正なデータを返していました (v10.01 からの問題)。
- .NET API (Windows): 不明な MSI フィールドを **WD_INTERRUPT** 構造体 – **dwLastMessage** および **dwEnabledIntType** – の定義へ追加し、**WDC_GET_ENABLED_INT_TYPE**、**WDC_GET_ENABLED_INT_LAST_MSG**、および **WDC_INT_IS_MSI** マクロを実装 (v9.10 からの問題)。

終息:

- サポート対象外のサンプルを削除

V11.00 の変更点

新機能:

- Windows 8: Windows 8 Developer Preview のサポートを追加
- DriverWizard: Windows GCC (MinGW/Cygwin) 用のコード生成を追加

強化:

- Linux: Debian と Red Hat Package Manager (RPM) 用の WinDriver インストール パッケージを追加
- Linux カーネル 3.2.4 のサポートを追加
- PCI: 64-bit BAR アドレスのサポートを追加

修正:

- 64-bit カーネルでの Windows PCI 32-bit アプリケーションの実行: WinDriver のデバイス情報を取得する関数 – **WDC_PciGetDeviceInfo() / WDC_PcmciaGetDeviceInfo()** または **low-level WD_PciGetCardInfo() / WD_PcmciaGetCardInfo()** への呼び出しの **dwOptions WD_ITEMS** フィールドの値を無視する問題を修正
- DriverWizard の Kernel PlugIn のコード生成: DriverWizard で生成された **KP_XXX_Open_32_64()** の関数を修正して、以前に Kernel PlugIn のコンパイルで発生していた Warning の問題を解決 (v10.40 からの問題)
- Windows 64-bit: WinDriver のインストール時にスペースを含むパスを指定した場合に発生する問題を修正 (例、**Program Files** ディレクトリへのインストールなど) (v8.00 からの問題)
- DriverWizard on Windows: デフォルトで DriverWizard で生成したプロジェクトを WinDriver の **wizard/my_projects** ディレクトリに保存 (v10.20 からの問題)
- USB on Windows: USB デバイスが無効なインターフェースを持つ **Configuration Descriptor** を提供する場合に、OS のクラッシュが発生する問題を修正 (v7.00 からの問題)
- USB on Windows CE: ストリーミング中に USB デバイスを取り外した際に、**Detach** 処理で遅延が発生する問題を修正 (v10.01 からの問題)
- USB streaming on Windows CE: **WDU_StreamOpen()** への呼び出しで 0 (無限) タイムアウトの処理の問題を修正; 以前はタイムアウトの設定なし (無限タイムアウト) ではなく、即座にタイムアウト エラーが発生 (v10.01 からの問題)
- Windows: WinDriver のドライバの起動時にマイナーな 20-byte のメモリー リークが発生する問題を修正
- Windows: Microsoft の Verifier ユーティリティ (**Verifier.exe**) を使用する際に発生するセキュリティ Warning が発生するレジストリ関連の問題を修正
- PCI on Windows: PCI デバイスのオープンの失敗で C# の WinDriver アプリケーションがクラッシュする問題を修正 (v8.00 からの問題)

終息:

- Mac OS X のサポートを終了
- Linux カーネル 2.2.x および 2.4.x のサポートを終了

V10.40 の変更点

新機能:

- Windows Embedded Standard 2009 のサポートを追加
- Windows Embedded Compact 7 ("Windows CE 7") のサポートを追加

- Linux kernel 2.6.39 のサポートを追加

強化:

- PCI on Mac OS X: 64-bit BAR アドレスのサポートを追加
- Windows DriverWizard: **<driver_name>_installation.bat** スクリプトを DriverWizard で生成されたドライバ コードに追加。名前を変更したドライバのインストールが容易になりました。
- PCI Altera sample: **WinDriver/altera/pci_dev_kit/lib/altera_lib.c** サンプルにライセンス登録を追加

修正:

- WinDriver Windows installation: **wdreg** ユーティリティを変更して ERROR_FILE_NOT_FOUND エラーが発生する問題を修正。RunOnce レジストレーション キーの定義の不足で発生していました。
- Windows CE: PCI デバイスのリソースの検出の問題を修正
- USB on Linux and Windows CE: マルチ インターフェイス デバイスのエンドポイントでアクションを実行する際に **usb_diag** サンプルで発生するアプリケーション クラッシュの問題を修正 (v10.0.2 からの問題)
- USB on Linux and Windows CE: マルチ インターフェイス デバイスのデバイス情報をプリントする際に **usb_diag** サンプルで発生するセグメンテーションの失敗の問題を修正 (v10.3.0 からの問題)
- Windows 64-bit: DriverWizard で生成される Borland C++ Builder のプロジェクトの WinDriver のライブラリのパスの問題を修正 (v8.0.0 からの問題)
- PCI: 64-bit カーネルで 32-bit アプリケーションを起動する際に割り込み有効関数 – WDC_IntEnable() / WD_IntEnable() – が返す割り込みタイプの問題を修正 (v9.1.0 からの問題)
- Visual Basic .NET (Windows): サンプルおよび DriverWizard で生成された Visual Basic .NET 2008 のプロジェクトのローディングの問題を修正 (v10.2.0 からの問題)
- DriverWizard on Windows: Select Code Generation Options と Select Additional Options ダイアログでキーボード入力のサポートの問題を修正 (v9.0.0 からの問題)
- PCI on Windows: DriverWizard の C# Kernel PlugIn コード生成の問題を修正 (v8.0.0 からの問題)
- Linux USB: Debian Squeeze ディストリビューションで WinDriver USB Linux GPL ドライバ – **windrvr6_usb.ko** – のインストールの問題を修正
- DriverWizard PCI: 割り込み転送コマンドを含む DriverWizard で生成されたユーザーモードのドライバ プロジェクトのコンパイル エラーの問題を修正 (v10.3.1 からの問題)
- USB on Linux: USB コントロール転送のメモリ リークの問題を修正
- PCI on Windows: WD_INTERRUPT のロス割り込みカウンター (dwLost field) で発生するエラーを修正
- PCI on Windows: DMA 割り当て (WDC_DMASGBufLock() / WD_DMALock() を使用した) の問題を修正 – フォールバックを無効にして DMA 割り当てを廃止 (v9.2.0 からの問題)

V10.31 の変更点

新機能:

- Windows 7 および Server 2008: Windows 7 および Windows Server 2008 R2 Service Pack 1 のサポートを追加
- Linux: Linux kernel 2.6.38 のサポートを追加

強化:

- PCI Xilinx サンプル: Xilinx Virtex5 のサンプル (**WinDriver/xilinx/virtex5**) を Xilinx Bus Master DMA (BMD) デザインの PCI Express カードの汎用的なサンプル (**WinDriver/xilinx/bmd_design**) に変更; これによって、サンプル ファイル名と API の名前を変更。新しいサンプルは、Xilinx Virtex および Spartan 開発キットで動作検証しました。
- PCI Xilinx BMD Kernel PlugIn サンプル: KP_BMD_Open_32_64() 関数を **WinDriver/xilinx/bmd_design/kp/kp_bmd.c** へ追加 (以前は、**WinDriver/xilinx/virtex5/bmd/kp/kp_vrtx5.c**)。
- PCI: **pci_diag** サンプルおよび DriverWizard で生成される KernelPlugIn のライブラリとドライバ コードで、不必要なポインタ型を削除し、内部構造体型の名前を変更し、コード変換を統一し、混乱を回避

修正:

- USB on Linux: CentOS Linux ディストリビューションで、DriverWizard が USB デバイスを不正に識別する問題を修正 (v10.21 からの問題)
- Mac OS X: 64-bit Mac OS X プラットフォームで、DriverWizard と **wddebug_gui** を含む 32-bit アプリケーションの起動のサポートを修正。この修正に関連して、32-bit と 64-bit Mac OS X 用に別々の WinDriver のインストーラを用意
- USB on Windows: ユーザーモードからデバイスへ SET_CONFIGURATION リクエストを送信した後に、スタンバイから復帰する際に BSOD が発生する問題を修正 (v9.21 からの問題)
- Windows CE: **project_wd.bib** を修正して、WinCE6 プラットフォームで、WinDriver のカーネル モジュールのインストールをサポート
- PCI: 同時に開く可能性のある DMA ハンドルの制限数を削除
- Linux: Linux カーネルの real-time パッチで、ドライバのインストール エラーの問題を修正 ("init_MUTEX 関数の暗黙の宣言")

- Kernel PlugIn: 64-bit Kernel PlugIn ドライバで 32-bit ユーザーモード アプリケーションを使用した際に、Kernel PlugIn の割り込み有効／無効関数が呼ばれなかった問題を修正 (v10.30 からの問題)
- Kernel PlugIn on Windows: 有効な Kernel PlugIn ドライバがある時に、WinDriver のカーネル モジュールを無効／有効にすると BSOD が発生する問題を修正
- Kernel PlugIn: サンプルおよび DriverWizard で生成される C および C# の PCI コードを修正して、ユーザーモードから Kernel PlugIn ドライバへ正しいコンテキスト タイプを渡し、Kernel PlugIn high-IRQL のレガシー割り込みハンドラーのコンテキスト キャストを修正し、レガシー割り込みを受信する際に発生する OS クラッシュの問題を解消 (v10.30 からの問題)
- Kernel PlugIn: サンプルおよび DriverWizard で生成された Kernel PlugIn の open 関数の不具合を修正。以前は、ユーザーモードから Kernel PlugIn ドライバへハンドルを開く際に、OS クラッシュの問題が発生 (v10.30 からの問題)

終息:

- Windows 2000 のサポートを終了

V10.30 の変更点

新機能:

- Windows: .NET Framework 4.0 用の WinDriver .NET API DLL (**wdapi_dotnet<version>.dll**) のバージョンを追加
- Linux: Linux カーネル 2.6.37 のサポートを追加

強化:

- USB on Windows CE: ストリーミングのパフォーマンスを改良
- PCI on Linux and Mac OS X: WDC_GET_ENABLED_INT_LAST_MSG macro を改良 (最後に受信した MSI/MSI-X 割り込みのメッセージ データの取得用)
- USB on Windows: WDU_SelectiveSuspend() の .NET ラッパーを追加
- Kernel PlugIn (PCI on Windows, Linux, and Mac OS X): 32-bit アプリケーションから 64-bit Kernel PlugIn への通信を追加。この変更には、新しい KP_INIT 構造体のフィールド (funcOpen_32_64) を含み、PCI Kernel PlugIn のサンプル (**kp_pci**) および DriverWizard で生成した Kernel PlugIn のコードを改良

修正:

- Windows: **wdreg** – 一つの INF ファイルで複数のデバイスを有効/無効にする際に、一つのデバイスでエラーが発生する問題を修正
- PCI on Linux x86_64: PCI read/write の処理中にクラッシュする問題を修正 – 例えば、WD_Transfer() への呼び出しや、割り込み転送コマンドの実行中
- USB on 64-bit Windows: 32-bit アプリケーションから WDU_StreamGetStatus() を呼び出す際に発生する OS のクラッシュの問題を修正
- Linux 2.4.x: WinDriver のビルド処理の問題を修正 (v10.21 からの問題)
- USB on Windows: USB_ISOCH_NOASAP フラグをセットして、アイソクロナスの書き込み (OUT) ストリームの動作で発生する問題を修正
- Windows: Windows 7 WDK のビルド オプションのサポートを修正 (**ddk_make.bat** ビルド スクリプトの – win7 と win7_x64)。これで、Kernel PlugIn のビルドに関する問題も修正 (10.21 からの問題)
- PCI on Windows: WDC_PciDeviceOpen() – WinDriver と動作するように登録されていない PCI デバイスの認識の問題を修正 (v10.21 からの問題)
- Debug Monitor: コマンドラインの Debug Monitor ユーティリティ – **wddebug** – のデフォルトのデバッグ レベルを D_TRACE に変更 (GUI の Debug Monitor ユーティリティ **wddebug_gui** と一致)
- PCI on Mac OS X: 複数の転送用に DMA バッファを繰り返し使用する際に発生する OS のクラッシュの問題を修正 (v10.20 からの問題)
- PCI on Linux 2.6.x: 連続 DMA バッファの割り当ての際に発生する可能性があるカーネル パニックの問題を修正
- PCI on Mac OS X: Plug-and-Play / power-management PCI イベントの登録を解除する際に発生する OS クラッシュの問題を修正 (v10.20 からの問題)
- USB on 64-bit Linux: WDU_Transfer() の INFINITE (無限) タイムアウトの問題を修正
- PCI on Mac OS X: ハードウェアの割り込みが発生していない際のユーザー割り込みハンドラーへの呼び出しで発生する問題を修正
- PCI on Mac OS X: 連続 DMA バッファまたは共有バッファの割り当て用の WDC_DMAContigBufLock()、WDC_SharedBufferAlloc()、および WD_DMALock() で返す仮想カーネルモードのマップ アドレスの値の問題を修正。これで、データ転送で返した値を使用する際に発生する OS クラッシュの問題を解決
- PCI on Mac OS X: デバイスのメモリ領域の検出の問題を修正
- PCI on Mac OS X and Windows CE: WinDriver のアプリケーション クリーンナップの一部として、アプリケーションで明示的にクローズしていないデバイスに対する割り込みリソースのクリーンナップを追加
- PCI: WDC_ReadXXX() または WDC_WriteXXX() へ不正なデバイス ハンドルを渡す際に発生するアプリケーション クラッシュの問題を修正

- Windows Mobile ARMV4I: WinDriver のサンプルおよび WinDriver の Driver Wizard で生成された Platform Builder のプロジェクトの WinDriver API のライブラリへのパスが不正なためプロジェクトのビルドに失敗する問題を修正 (v7.01 からの問題)
- PCI on 64-bit OSs: 32-bit アプリケーションを起動する際にカーネルモードからユーザーモードのメモリ スペースのマップの解除を無視して発生するメモリ リークの問題を修正
- ISA/PCMCIA on Windows Vista and higher: ISA/PCMCIA デバイスに対して WDC_IntDisable() を呼び出す際に発生する OS クラッシュの問題を修正 (v9.20 からの問題)
- PCI on Mac OS X: 64-bit DMA アドレスの処理の問題を修正

V10.21 の変更点

新機能:

- Windows Embedded Standard 7 のサポートを追加
- Linux: Linux カーネル 2.6.30 - 2.6.35 のサポートを追加

強化:

- Windows: Microsoft Visual Studio 2010 IDE のサポートを追加
- Windows: ddk_make.bat ビルド スクリプト、および MS Visual Studio Kernel PlugIn のサンプルと DriverWizard で生成されるプロジェクトを強化:
  - Windows 7 用のビルド環境を追加
  - WDK (Windows Driver Kit) の versions 7600.x のビルド オプションのサポートを追加
  - PCI: WDC_PciDeviceOpen() のパフォーマンスを改良

修正:

- DriverWizard (Windows): 64-bit Windows 版の DriverWizard を使用して生成した 32-bit アプリケーションのコンパイルの問題を修正
- PCI on Linux PPC: Linux PPC カーネル 2.6.23 およびそれ以降で PCI カードのスキャンの問題を修正。WinDriver のユーティリティを使用して、接続されているすべての PCI カードを検出し、そのカードを開く問題がありました (v9.10 からの問題)
- Windows 64-bit: 64-bit Windows プラットフォームで 32-bit アプリケーションを実行する際に (アプリケーションの実行中に USB デバイスの電源を OFF にする場合も含む)、OS クラッシュが発生する場合など、内部競合の問題を修正

V10.20 修正版の変更点

修正:

- Windows、Linux および Mac: windrvr.h ファイルの WD_DEFAULT_DRIVER_NAME_BASE の定義を変更
  WinDriver のドライバ名を Windows と Linux では "windrvr6" へ、Mac では "WinDriver" へ参照を修正 (v10.20 の問題)

  注意: 修正版の WinDriver では、カーネル モジュールのビルド日付が 7月 15日になっています。

  Debug Monitor を起動して、Build Date: Jul 15 2010 と表示されることをご確認ください。

V10.20 の変更点

強化:

- WinDriver PCI: Mac OS X 10.5.x - 10.6.x での PCI および PCI Express のサポートを追加

修正:

- Kernel PlugIn (Linux): Linux で DriverWizard を使用してプロジェクト名に大文字が含まれる Kernel PlugIn プロジェクトのコードを生成する際に発生する問題を修正
- Windows USB: .NET および Delphi 用の WinDriver USB API で、USB_ISOCH_RESET のフラグの定義を追加
- DMA (Linux PCI): Linux で Scatter/Gather DMA アドレスの計算を修正 (v9.21 からの問題)
- Windows: コールバックの完了を適切に示さない際に、ThreadWait で発生する可能性があるデッドロックの問題を修正
- Windows: Windows Server 2008 と Windows 7 用にドライバの署名を追加

V10.11 の変更点

強化:

- Kernel PlugIn (Linux): Linux Kernel PlugIn ドライバの配布方法を変更、これによって、ソースコードなしでドライバの配布が可能になりました
- Kernel PlugIn (Linux): **kbuild** framework を使用した Kernel PlugIn ドライバのビルドのサポートを追加
- PCI: 新しい WDC_PciScanRegisteredDevices() 関数を追加し、WinDriver と動作するように登録したデバイスのみを PCI バスのスキャンをします

修正:

- USB: **usb_diag.c** サンプルと **usb_diag.c** ライブラリを修正し、multi-interface を持つデバイスで、アクティブでないインターフェイスのパイプの読み込み、書き込みおよびリセットをサポート
- USB ストリーミング: USB ストリーミング API と **usb_diag** サンプルを修正し、コントロール パイプでのストリーム転送の実行を防止
- Linux: WinDriver のカーネル モジュールのビルド用に --enable-usb-support=no 設定オプションのサポートを修正 (v10.00 からの問題)
- Linux と Windows のインストール: ACPI テーブルの不正利用によって、発生する可能性がある WinDriver のインストールのクラッシュの問題を修正
- Windows USB: WDU_Transfer() を使用して、non-NULL セットアップ パケットで data-pipe 転送を実行する際に発生する可能性のある OS のクラッシュの問題を修正 (v9.22 からの問題)
- Windows USB: bRequest=0x09 で Vendor または Class USB コントロール転送を発行する際に発生する OS のクラッシュの問題を修正 (v10.00 からの問題)
- Windows USB: high-speed USB デバイスへ full-speed の maximum-URB 転送サイズを割り当てる問題を修正
- PCI: 誤動作を起こす PCI デバイスに対して、WDC_PciScanDevicesByTopology() を呼んだ際に発生する OS クラッシュの問題を修正 (v8.00 からの問題)
- PCI/PCMCIA/ISA: WDC_DBG_DBM_FILE_ERR と WDC_DBG_DBM_FILE_TRACE macro の定義を修正
- USB ストリーミング API: デバイスからのデータ読み込みのサイズが、ストリームのデータ バッファの残りのスペースよりも大きい場合に、ブロッキング ストリームに対して、WDU_StreamRead() への呼び出しが失敗する問題を修正 (v9.01 からの問題)
- Windows: 外部 DLL から ThreadWait() を呼び出す際に発生する可能性があるハングの問題を修正 (v7.00 からの問題)
- Windows USB ストリーミング API: ストリーム バッファのリリース時にストリーム タイマーをリセット (v9.01 からの問題)

V10.10 の変更点

新規:

- Windows: Windows 7 (32-bit と 64-bit) のサポートを追加

修正:

- Windows USB: USB_ISOCH_NOASAP フラグを設定しない場合にアイソクロノス転送でパケット ロスを生じる問題を修正 (v10.02 からの問題)
- Linux USB: Linux 64-bit プラットフォームで WinDriver USB 32-bit アプリケーションを実行する際に生じる問題を修正 (v9.21 からの問題)

V10.02 の変更点

新規:

- Linux: Linux カーネル 2.6. 29 のサポートを追加

強化:

- Delphi (Windows USB): WDU_StreamXXX() USB streaming API 用に **wd_utils.pas** に Delphi の定義を追加
- PCI/PCMCIA/ISA: **wdc_defs.h** から **status_strings.h** ヘッダ ファイルの定義を削除、代わりに関連するソース ファイルから明示的に **status_strings.h** の定義を追加

修正:

- DriverWizard (Windows PCI/PCMCIA/ISA): Windows で保存した DriverWizard の PCI/PCMCIA/ISA プロジェクトを使用してコードを生成する際に、生成された INF ファイルに対象のデバイスの Vendor ID と Product ID が含まれない問題を修正 (v9.0.1 からの問題)
- C# のサンプル コードと DriverWizard で生成された USB のコード (Windows): WDU_Init() 処理中に DeviceDetach() を呼ぶ際に発生した競合条件の問題を修正
- DMA (Linux PCI): Linux カーネル 2.6.25 およびそれ以降で、DMA 処理の問題を修正、システム クラッシュが発生する場合がありました (v10.0.1 からの問題)
- DriverWizard (PCI/ISA): DriverWizard で ISA と Virtual PCI Device の Resource の編集でアクセス違反の問題を修正、DriverWizard がハングする場合がありました (v9.0.1 からの問題)
- 割り込み: 割り込みロスのカウンターによる問題を修正、割り込みが生成されなかった場合でも、ユーザーの割り込みハンドラーを繰り返し呼び出す場合がありました
- USB (Windows): タイムアウトによってシングル ブロッキング USB 転送を返す際に Windows OS がクラッシュする問題を修正 (v10.0.1 からの問題)
- USB (Windows CE): Windows CE で、WDU_StreamStop() と WDU_StreamClose() のリターンの遅延を修正 (v10.0.1 からの問題)

- DriverWizard (ISA): 以前は、ISA カードで I/O BAR と他のタイプのリソース (Memory BAR や割り込みなど) の両方を持つ場合、 DriverWizard で生成されたコードの I/O リソースの情報が他のリソースの情報に上書きされていた問題を修正
- Linux: `**make install**` で WinDriver のカーネル モジュールのロードに失敗する問題を修正、 モジュールを正しい場所にインストールしない Linux のディストリビューションがありました (v9.0.1 からの問題)
- DriverWizard: WinDriver のカーネル ドライバがロードされていない状態で DriverWizard を使用するとエラー メッセージの表示を追加 (v9.0.1 からの問題)

V10.01 の変更点

新規:

- Linux: Linux カーネル 2.6.28 のサポートを追加
- USB (Windows CE): Windows CE で USB ストリーミングのサポートを追加

強化:

- Linux: **windrvr6** ドライバのインストールで、ドライバのバージョンとインストール済みの windrvr6_usb ドライバのバージョンが一致するか確認
- Windows USB: Selective Suspend のサポートを **usb_diag** サンプルと DriverWizard で生成された USB コードに追加
- Debug Monitor (Windows CE): Windows CE で **wddebug** Debug Monitor ユーティリティの GUI 版のサポートを追加、これでコマンドライン インターフェイスをサポートしない Windows CE プラットフォームでも Debug Monitor ユーティリティの使用が可能
- USB ストリーミング API (Windows CE): Windows CE で、WDU_StreamOpen() へ新しい USB_STREAM_MAX_TRANSFER_SIZE_OVERRIDE フラグを追加し、読み込みストリームのデフォルトの最大転送サイズをオーバーライド

修正:

- DriverWizard (PCI/PCMCIA/ISA): Interrupt Request (IRQ) 番号を 10 進法で DriverWizard に表示し、Driver Wizard で生成されたコートとサンプル コードと一致 (v9.0.0 からの問題)
- 割り込み (Linux): MSI-X (Message-Signaled Interrupts) の有効に失敗すると、WinDriver が代わりに MSI (Message-Signaled Interrupts) を有効にする問題を修正 (v9.1.0 からの問題)
- Windows CE: Windows CE ARMV4I プラットフォーム用に **wdreg.exe** の問題を修正 (v9.0.0 からの問題)
- DMA (Windows): DMA 転送完了後、DMA バッファの不正なデータの問題を修正 (v9.2.1 からの問題)
- DriverWizard (Linux USB): 特定の Linux ディストリビューション (新しい Ubuntu ディストリビューションなど) で、DriverWizard の USB デバイスの認証の問題を修正
- Windows: **wdreg** で **-compat** オプションを使用して、同じハードウェア ID を持つ複数のデバイスのドライバをインストールする際に、ドライバのインストールの冗長の問題を修正
- Windows: **メモ帳** (Windows アプリケーション) を使用して正しく参照できるように windrvr6.inf を修正 (v10.0.0 からの問題)
- Linux: root の WinDriver ディレクトリから "make install" を実行する際に、WinDriver Linux ドライバのインストールの問題を修正 (v10.0.0 からの問題)
- PCI/PCMCIA/ISA: WDC_DIAG_ReadWriteAddr() および WDC_DIAG_ReadWriteBlock() **wdc_diag_lib** 関数を修正して、Cancel 入力を正しく処理。本修正は、**pci_diag** および **pcmcia_diag** サンプルと DriverWizard で生成されたコードに反映されます (v7.0.0 からの問題)
- USB ストリーミング API (Windows): WDU_StreamGetStatus() でタイムアウト時に WD_TIME_OUT_EXPIRED を返すように修正 (v9.0.0 からの問題)

V10.00 の変更点

新規:

- Linux: Linux カーネル 2.6.25 - 2.6.27 のサポートを追加
- Linux USB: Linux 用の WinDriver USB の実装を GPL (GNU General Public License) を持つ別のドライバ モジュール (**windrvr6_usb**) へ移行

強化:

- Windows: Windows Server 2008 OS の公式サポートを追加
- Windows Vista: Windows Vista SP1 で WinDriver が正常に動作することを検証
- Windows XP: Windows XP SP3 で WinDriver が正常に動作することを検証
- PCI: PLX 6466 Bridge の拡張サポートを追加。ライブラリ API (**plx_lib_6466**) とサンプルの診断アプリケーション (**p6466_diag**) が含まれます
- Windows: Microsoft Visual Studio 2008 IDE のサポートを追加
- Windows: **ddk_make.bat** ビルド スクリプト、サンプルの MSDEV の Kernel PlugIn プロジェクトおよび DriverWizard で生成したコードの強化
    - Windows Vista x64 用のビルド環境を追加

- WDK (Windows Development Kit) のバージョン 6001 (およびそれ以降) のビルド オプションのサポートを追加

修正:

- USB: multi-interface の USB デバイスで DriverWizard および **usb_diag** サンプルを使用した際に発生するクラッシュの問題を修正 (v9.21 からの問題)
- PCI Express (Linux 64-bit): Virtex 5 のサンプルの診断アプリケーション (**virtex5_diag**) の問題を修正: 以前は、Linux 64-bit で、サンプルでは、ユーザーの入力に関わらず、常に DMA バッファ サイズに 0 を設定していました (v9.20 からの問題)
- 割り込み (Windows CE): Windows CE で割り込みを有効／無効にする際に発生したメモリ リークの問題を修正 (v9.00 かのの問題)
- DriverWizard (Linux PPC 64-bit): Linux PPC 64-bit プラットフォームで保存した DriverWizard のプロジェクトを開く際の問題を修正
- PCI: WinDriver PCI のサンプルの診断アプリケーション (**pci_diag**) で、複数の Kernel PlugIn WDC デバイス ハンドルの同時オープンをサポート
- Windows: Windows の WinDriver カーネル ドライバ (**windrvr6.sys**) をインストール／アンインストールする際に、カーネル ドライバへの開いているハンドルがある場合に、**wdreg** が Warning を返す問題を修正 (v9.00 からの問題)
- Windows: **wdreg** (WinDriver のインストール ユーティリティ) のメモリ リークの問題を修正
- DriverWizard PCI/PCMCIA/ISA: DriverWizard のプロジェクトで Register Access モードの問題を修正: 以前は、保存した DriverWizard のプロジェクトを開く際に、最初に定義した Access モードに関わらず、定義した Register の Access モードを Read/Write に設定してました

終息:

- Windows 98/Me: WinDriver の配布モジュールから Windows 98/Me 用のファイルを削除 (Windows 98/Me のサポートは、v9.21 で終了しました)

V9.21 の変更点

新規:

- Linux: Linux カーネル 2.6.24 のサポートを追加

強化:

- Debug Monitor (Windows Vista): Windows Vista 上で Debug Monitor (wddebug_gui / wddebug) を使用して、Kernel Debugger メッセージのログを取得するサポートを追加
- Kernel PlugIn (Windows Vista): Windows Vista 用に WinDriver の Kernel PlugIn のプロジェクトにビルド環境を追加

修正:

- Windows Vista: Windows Vista で WinDriver ツールキットをアンインストールした際に、Windows スタート メニューから WinDriver のエントリを削除しなかった問題を修正
- サンプル (Windows): MSDEV 6.0 IDE を使用して、WinDriver のサンプルのコンパイルの問題を修正
- Kernel PlugIn (Windows XP): Windows XP で Kernel PlugIn のサンプルのロードの問題を修正 (v9.20 からの問題)
- Windows XP Embedded: XP Embedded で、OS のドライバの後に WinDriver がロードされるように XP Embedded の WinDriver のコンポーネント (windriver.sld) を使用する WinDriver のインストールを修正
- USB (Windows): USB デバイスを接続後すぐに USB デバイスを取り外した際に発生するドライバ クラッシュの問題を修正
- USB (Windows CE): Windows CE で multi-interface の USB デバイスの処理の問題を修正
- Windows: wdreg を使用して、複数のデバイス用の INF ファイル (15 個以上のデバイス) をインストールする際にハングする問題を修正
- 64-bit プラットフォームで 32-bit アプリケーションを実行する際の問題を修正 (v9.10 からの問題)
- Windows: wdreg install コマンドで、複数のデバイス用の INF ファイルをインストールする際に、既に二つ以上のデバイスがホストに接続されている状態でも pre-install となる問題を修正
- DriverWizard (Windows): DriverWizard で生成された Virtual PCI デバイス用の INF ファイルの Hardware ID の問題を修正 (v9.00 からの問題)
- 割り込み (Windows): Delphi と VB 6.0 の割り込み API をアップデートし、MSI (Message-Signaled Interrupts) と MSI-X (Extended Message-Signaled Interrupts) をサポート (v9.10 からの問題)
- Kernel PlugIn: Xilinx Virtex 5 PCI Express のサンプル (kp_vrtx5) のコンパイル エラーを修正 (v9.20 からの問題)
- Windows: Windows のスタート メニューから Jungo のパートナー ページへのリンクの問題を修正

終息:

- Windows 98/Me のサポートを終了
- Linux Itanium プラットフォームのサポートを終了
- USB: Linux PPC プラットフォームで USB のサポートを終了

V9.20 の変更点

強化:

- DriverWizard (PCI): MSI (Message-Signaled Interrupts) および MSI-X (Extended Message-Signaled Interrupts) のサポートを追加。GUI の表示、INF ファイルの生成、ユーザーモードおよび Kernel PlugIn のコードの生成が含まれます
- PCI Express: BMD (Bus Mastering DMA Validation Design) ファームウェアを持つ Xilinx Virtex 5 PCI Express チップ用の Kernel PlugIn のサンプル (**kp_vrtx5**) を追加。新しいサンプルに対応した Virtex 5 BMD ライブラリ (**virtex5_lib**) を修正。サンプルでは、MSI (Message-Signaled Interrupts) または MSI-X (Extended Message-Signaled Interrupts) の Kernel PlugIn の処理を実装
- 割り込み (Linux): Power PC Linux プラットフォームで IRQ 0 のサポートを追加
- 割り込み (Windows CE): WinDriver の割り込みハンドラーを修正して、共有 PCI 割り込みを正しく処理
- 割り込み: WinDriver の割り込みハンドラーを修正して、割り込み転送コマンドに関わらず、非共有割り込みのオーナーシップを要求

修正:

- Kernel PlugIn (Linux): Linux で Kernel PlugIn ドライバをロードする際に発生する問題を修正 (v9.10 からの問題)
- USB (Linux): Linux で複数の USB 転送リクエストのロードによって、不正な転送タイムアウト エラーが発生する問題を修正 (v9.00 からの問題)
- USB (Windows): Windows でスタンバイ モード中に USB デバイスを取り外した際に発生するリリースの問題を修正 (v9.10 からの問題)
- USB (Windows): ストリーミング API を使用してバルク パイプのリッスンをする際に、ユーザーモードの WinDriver USB アプリケーションの異常終了によって Windows クラッシュが発生する問題を修正 (v9.01 からの問題)
- ISA 割り込み: ISA 割り込みのサポートを復活 (v9.10 からの問題)
- Linux: WinDriver 64-bit Linux のインストール時のクラッシュの問題を解決 (v9.10 からの問題)
- Kernel PlugIn (Linux): Kernel PlugIn の malloc() の実装を使用したメソッドと互換性を持つように Linux Kernel PlugIn の free() 関数の実装を修正 (v9.10 からの問題)

V9.10 の変更点

新規:

- PCI 割り込み: Windows Vista と Linux で、Message-Signaled Interrupts (MSI) と MSI-X のサポートを追加
- PCI Express: Bus Mastering DMA Validation Design (BMD) ファームウェアを持つ Xilinx Virtex 5 PCI Express チップの拡張サポートを追加 - ライブラリ APIs (**virtex5_lib**) と DMA と Message-Signaled Interrupts (MSI) の処理を実行するサンプル アプリケーション (**virtex5_diag**) が含まれます
- Linux: Linux カーネル 2.6.22 と 2.6.23 のサポートを追加
- Windows XP Embedded: Windows XP Embedded プラットフォームで自動ドライバ インストール用の新しい Windows XP Embedded WinDriver コンポーネント (**windriver.sld**) を追加

修正:

- USB (Linux): Linux でデータのない Control 転送で以前は、invalid parameter error で失敗した問題を修正 (v9.01 からの問題)
- USB (Linux): Linux で大きいデータ バッファを使用した WDU_Transfer() write (OUT) 転送の問題を修正 (v9.01 からの問題)
- USB (Windows): recepient 'other' で type 'class' の Control request を以前は vendor request として転送していた問題を修正
- Linux: **wdreg** スクリプトを使用して、名前を変更した WinDriver カーネル モジュールのインストールの問題を修正 (v9.00 からの問題)
- Linux: openSUSE 10.2 Linux distribution での WinDriver の設定の問題を修正
- Linux: enable-usb-support=no configuration オプションで WinDriver のカーネル モジュールのビルド用に OS_get_device_property のサポートを追加 (v9.00 からの問題)
- DMA: 32-bit Windows および Linux プラットフォームで WD_DMALock() 64-bit アドレスの処理の問題を修正 (v8.00 からの問題)
- DMA (Linux): Linux プラットフォームで contiguous DMA バッファをロックする際のメモリ リークの問題を修正 (v8.10 からの問題)
- Kernel PlugIn (Linux): embedded Linux 2.6 distribution で Kernel PlugIn のメモリ割り当ての問題を修正 (v7.00 からの問題)
- DriverWizard (Linux): Linux 用の DriverWizard を QT ライブラリに静的にリンクし、対象のプラットフォームに QT ライブラリがインストールされているかに関わらず、これらの OS で Wizard の使用できるように修正 (v9.00 からの問題)
- PCI (Windows および Windows CE): WDC_PciScanDevicesByTopology() を WinDriver API DLL (**wdapi<version>.dll**) の export に追加 (v8.00 からの問題)

- large string を正しくサポートするように CCString::Format class method を修正。その結果、Windows で長い **wdreg** のログ メッセージの問題を修正

V9.01 の変更点

新規:

- Linux: Linux カーネル 2.6.20 と 2.6.21 を含む最新の Linux カーネルのサポートを追加
- USB Streaming API: 新しい WDU_StreamGetStatus() 関数を追加 - Stream のカレント ステータスを取得

強化:

- USB: Stream 転送の実行テストを WinDriver USB 診断サンプル (**usb_diag**) に追加
- DriverWizard PCI: 既存の DriverWizard の PCI プロジェクトを開く際に、デバイスの Plug-and-Play 情報を確認
- DriverWizard GUI: 1024 x 768 以上の解像度のフル スクリーンで DriverWizard を表示 (v9.00 の問題)
- DriverWizard: Windows 英語版以外の OS での DriverWizard の使用をサポート
- DriverWizard: キーボード コマンドのサポートを改良 (v9.00 の問題)
- Kernel PlugIn: サンプルの Windows Kernel PlugIn ドライバ - **kp_pci.sys** のデジタル署名の追加、これによって、デジタル署名のないドライバのインストールを禁止または制限する Windows プラットフォームへのドライバのインストールをスムーズにします (Vista x64 など)
- USB Streaming API: 新しい WDU_StreamOpen() フラグ - USB_STREAM_OVERWRITE_BUFFER_WHEN_FULL を追加 - 転送を完了するのに Stream のデータ バッファに十分な空き容量がない場合に read stream のデータを上書き
- SB: WDU_Transfer() を拡張し、バッファ サイズを 0 に設定、または NULL バッファ ポインタを渡すことのいずれかで、zero-length 転送の初期化が可能
- DriverWizard: その他、GUI の改良

修正:

- Windows 98: Windows 98 プラットフォームへのドライバの配布の問題を修正 (v9.00 の問題)
- Windows: UNICODE プリプロセッサ定義を含むプロジェクトのコンパイルの問題を修正 (v9.00 の問題)
- PCI/PCMCIA/ISA: DriverWizard で生成されたコードに転送コマンドがない場合に発生するコンパイルの問題を修正 (v9.00 の問題)
- DriverWizard PCI/PCMCIA/ISA: DriverWizard で割り込みのないデバイスのコードを生成する際に DriverWizard がクラッシュする問題を修正 (v9.00 の問題)
- Linux PCI: ページ アラインされていないバッファの Scatter/Gather DMA ロックの問題を修正 (v6.20 からの問題)
- PCI/PCMCIA/ISA: WDC_DriverOpen() を呼ぶ前に WDC_Version() を呼び出し可能 (v7.00 の問題)
- USB Steaming API: アプリケーションが以上終了した場合に streaming API の cleanup の問題を修正 (v9.00 の問題)
- USB Steaming API: zero-length 転送のクラッシュの問題を修正 (v9.00 の問題)
- DriverWizard USB: 以前の "write to pipe" で入力データの変更を生じていた問題を修正 (v9.00 の問題)
- DriverWizard USB: multiple-interface デバイスの INF ファイルの生成の問題を修正 (v9.00 の問題)
- PLX: PLX ライブラリでマスター デバイス用に使用したローカル ベース アドレスの問題を修正 (v7.00 からの問題)
- PCI/PCMCIA/ISA 割り込み: WD_IntWait() の戻り値の問題を修正
- 64-bit アーキテクチャで 32-bit アプリケーションの exit cleanup の問題を修正 (v6.00 からの問題)
- USB: 64-bit アーキテクチャで起動する 32-bit アプリケーションから WDU_GetDeviceRegistryProperty()、WDU_GetDeviceAddr() および WD_GetDeviceProperty() の使用を有効
- DriverWizard ISA: DriverWizard のコード生成のクラッシュの問題を修正 (v9.00 の問題)
- Windows: **wdreg** インストール ユーティリティを修正し、**-compat** フラグなしでプログラムを起動する場合にのみ Driver Install Frameworks API (DIFxAPI) DLL をロード (v8.10 からの問題)
- Windows: **wdreg** インストール ユーティリティを修正し、WinDriver のカーネル モジュール (**windrvr6.sys** またはこのドライバの名前を変更したドライバ) をインストールする際に、新しいハードウェアに自動的に再スキャンするのを回避 (v8.10 からの問題)
- DriverWizard PCI/ISA: WinDriver の以前のバージョンで生成した古い DriverWizard のプロジェクトの Open をサポート (v9.00 の問題)
- Windows Mobile 5: DriverWizard のプロジェクト ファイルのコード生成の問題を修正 (v8.01 からの問題)
- .NET PCI/PCMCIA/ISA: WinDriver .NET API DLL (**wdapi_dotnet<version>.dll**) を修正し、割り込みが有効な際に、割り込み処理のために使用した転送コマンドの転換による garbage collector を回避 (v7.00 からの問題)。
- USB: Silicon Laboratories C8051F320 サンプル (**c8051f320_diag**) を修正し、4KB の倍数ではないバッファ サイズをサポート
- Windows CE 6.0: 割り込みおよびイベントの優先順位の決定の問題を修正 (v9.00 の問題)

V9.00 の変更点

新規:

- Windows USB: Windows 2000 以降で同期および非同期のストリーム転送をサポートするストリーミング USB API を追加

- Windows CE: Windows CE 6.0 と Windows Mobile 6.0 のサポートを追加
- Linux: Linux カーネル 2.6.18、2.6.1.19 と 2.6.20 を含む最新の Linux カーネルのサポートを追加
- ドライバの配布: Windows、Linux および Solaris で WinDriver カーネル モジュールの名前を変更する機能のサポートを追加
- Windows USB: Ellisys Explorer 200 USB バス アナライザを DriverWizard に統合
- USB on Windows: 新しい WDU_SelectiveSuspend() 関数 - Windows XP SP2 およびそれ以降で selective bus suspend のサポートを追加
- USB on Windows: ポート サイクリングのサポートを追加 - 物理的なリセットなしで再エミュレーション - WDU_ResetDevice() 関数 (新しい CYCLE_PORT フラグ)
- PCI/PCMCIA/ISA: WDC C および .NET ライブラリへ新しい WDC_Version() 関数を追加 - WinDriver のカーネル モジュールのバージョンを取得

強化:

- GUI: DriverWizard と Debug Monitor ユーティリティの GUI (Graphical User Interface) を改良
- Windows: 最新バージョンの Driver Install Frameworks API (DIFxAPI) DLL - **difxapi.dll 2.1.0.0** を使用するように **wdreg** インストール ユーティリティをアップグレード
- Windows 2000 およびそれ以降: WinDriver のカーネル ドライバ (**windrvr6.sys**) を DTM (Microsoft Driver Test Manager) ツールでテスト済み、新しい WHQL 証明書に必要な PREfast 出力ファイルを追加
- Windows: Windows 2000 およびそれ以降には、WHQL 互換の WinDriver のカーネル ドライバ (**windrvr6.sys**)、Windows 98/Me およびそれ以降には、WHQL 非互換のドライバを提供
- USB: USB のサンプル (**usb_diag**) と DriverWizard で生成されるコードを改良

修正:

- DMA: 32-bit Windows プラットフォームで 64-bit DMA 転送の処理を改良
- PCI/PCMCIA/ISA: Windows 98/ME で WD_ITEM_DO_NOT_MAP_KERNEL フラグの動作を修正
- Linux: 64-bit Linux プラットフォームで 32-bit アプリケーションを起動する際の大きいメモリ アドレス (> 32-bit) へのアクセスを修正
- .NET (PCI/PCMCIA/ISA): WinDriver API DLL (**wdapi_dotnet<version>.dll**)、C# PCI サンプル (**PCI_Sample**) と PLX ライブラリ (**plx_lib_dotnet**) を修正。割り込みハンドラー ルーティンから割り込み転送コマンドの配列にアクセスするための特定のサポートを提供するため
- Windows USB: 大きい configuration descriptor を持つ USB デバイスの configuration が WD_USB_DESCRIPTOR_ERROR エラーで失敗する問題を修正
- Windows USB: USB 割り込みおよびバルク転送の最大サイズ (Windows Vista) を修正 (以前は 4KB までの制限)

既知の制限:

- Windows 98/Me: Windows 98/Me で DriverWizard を使用できません。
  ただし、Windows 2000 またはそれ以降で開発したドライバは、Windows 98/Me PC へ配布可能です。

v8.11 の変更点

新規:

- Linux: Linux カーネル 2.6.x で PowerPC 64 ビットのサポートを追加
- Linux: Linux カーネル 2.6.17 - 2.6.18 および Fedora Core 6 pre-releases のサポートを追加
- USB: Agere USS2828 USB チップの拡張サポートを追加 - ライブラリ API (uss2828_lib および uss2828_diag_lib) とデバイスのデフォルト ファームウェアのサンプル アプリケーション (default_firmware_sample) を含む

強化:

- USB: USB のサンプル (usb_diag) と DriverWizard で生成されるコードを改良
- USB (Windows): バルクおよび割り込みパイプで、USB Request Blocks (URB) の最大サイズをオーバーライドするための WDU_Transfer() フラグ (USB_BULK_INT_URB_SIZE_OVERRIDE_128K) を追加
- Linux and Solaris: Windows / Windows CE v8.1 で既に追加された強化を実装
  - DMA: DMA 同期化関数 (WD_DMASyncCpu() / WD_DMASyncIo() / WDC_DMASyncCpu() / WDC_DMASyncIo()) のパフォーマンスを改良
  - PCI Express: 4GB を超えるマップするコンフィグレーション スペースのサポートを追加
  - USB (Linux): Cypress EZ-USB API を拡張 - 複数デバイスのサポートと新しい EZ-USB 診断ライブラリ (ezusb_diag_lib)
  - 新しい ISA サンプル: Intel 8253/8254 のプログラム可能なタイマー チップ用のサンプルを追加 (IBM PC で利用可能)

修正:

- Linux PCI: 32-bit Linux アーキテクチャでの 64-bit メモリ ブロック転送を修正

- Windows PCI: 割り込みのページング可能 (pageable) なメモリの使用を回避
- Linux: 一部の Linux カーネル 2.6.15 で発生していたインストールの問題を修正
- 後方互換性 (PCI): 新しい WinDriver カーネル モジュールと古い WinDriver PCI アプリケーションを使用する場合の問題を修正 (v8.00 以降の問題)
- Windows CE: 拡張した割り込みのオプションを使用するアプリケーションのコンパイルの問題を修正
- DMA: AMCC ライブラリの DMA API (WinDriver/amcc/lib/ amcclib.c) を修正
- VB.NET (Windows): DriverWizard で生成された VB.NET のコードに WinDriver のライセンス コードを自動的に挿入する問題を修正
- Windows USB: USB デバイスを取り外して直ぐに際挿入した際に、発生するクラッシュを修正
- Linux and Solaris: Windows / Windows CE v8.1 の修正を実装
  - Kernel PlugIn: Kernel PlugIn の初期化のメモリ リークを修正
  - Kernel PlugIn: アプリケーションが異常終了した際に、割り込みを自動的に無効
  - DMA: 64 ビット アーキテクチャで起動する 32 ビット アプリケーションの DMA 処理を修正 (Linux AMD64 / Solaris SPARC 64)

v8.10 の変更点

新規:

- Windows Vista Beta 2 (およびそれ以降) for x86 と x64 のサポートを追加
- Windows: WinDriver のドライバの配布に WinDriver のシンボル ファイル (windrvr6.pdb) を追加 - マイクロソフト社の WHQL (Windows Hardware Quality Labs) のデジタル署名を取得するのに必要
- WinDriver USB Device (Firmware Development Kit): MCC18 コンパイラ version 3.04 で Microchip PIC18F4550 ファームウェアのコンパイルのサポートを追加

強化:

- DMA: DMA 同期化関数 (WD_DMASyncCpu() / WD_DMASyncIo() / WDC_DMASyncCpu() / WDC_DMASyncIo()) のパフォーマンスを改良
- PCI Express: 4GB を超えるマップするコンフィグレーション スペースのサポートを追加
- USB: Cypress EZ-USB API を拡張 - 複数デバイスのサポートと新しい EZ-USB 診断ライブラリ (ezusb_diag_lib)
- Windows: wdreg.exe インストール ユーティリティを改良し、マイクロソフト社の WHQL (Windows Hardware Quality Labs) の要件に準拠
- New ISA サンプル: Intel 8253/8254 のプログラム可能なタイマー チップ用のサンプルを追加 (IBM PC で利用可能)
- .NET: ネイムスペースに "Jungo" プレフィックス を持つように編集
- Windows CE: wdapi<version>.lib WinDriver API ライブラリの場所を変更

修正:

- Kernel PlugIn: Kernel PlugIn の初期化で発生したメモリ リークを修正
- Kernel PlugIn: アプリケーションが異常終了した際に、割り込みを自動的に無効
- Windows USB: 8 バイト以下のバッファでのアイソクロノス転送の処理を修正
- Windows USB: 以前はブロックしていた DriverWizard からの no data stage の read (IN) コントロール コマンドを発行
- Windows CE USB: 他のドライバの共有割り込みへの Listen に影響を与えないように、共有割り込みを無効
- Windows CE USB: リセット パイプのトグル ミスマッチの問題を修正
- DMA: 64 ビット アーキテクチャで起動する 32 ビット アプリケーションの DMA 処理を修正 (v8.00 からの問題)
- Windows PCI: 64 ビット アーキテクチャで起動する 32 ビット アプリケーションから呼ばれる際の複数転送関数 (WD_MultiTransfer() / WDC_MultiTransfer()) で MASK コマンド (CMD_MASK) を処理 (v8.00 からの問題)

廃止:

- Windows 版での Windows NT 4.0 のサポートを終了

v8.02 の変更点

新規:

- Windows Mobile 5.0 のサポートを追加。
- Linux: Linux カーネル 2.6.14 - 2.6.16 のサポートを追加。
- WinDriver USB Device (Firmaware Development Kit): Silicon Laboratories C8051F340 開発ボードのサポートを追加。
- Windows CE: Platform Builder カタログへ WinDriver コンポーネントを追加し、ドライバ開発を簡素化。
- WinDriver USB Device (Firmware Development Kit): MCC18 コンパイラ version 3.0 で Microchip PIC18F4550 ファームウェアのコンパイルのサポートを追加。

強化:

- PCI/PCMCIA/ISA: WD_Transfer() と WD_MultiTransfer() およびこれらの関数を使用する WDC API (WDC_Read/WriteXXX(), WDC_AddrXXX() および WDC_MultiTransfer()) のパフォーマンスを改良。
- USB (Windows): Windows XP および Server 2003 で、高速デバイス用の USB 転送のパフォーマンスを改良。
- PCI (Linux): 最大 contiguous DMA バッファ サイズの制限 128KB を解除。
- 小さい sleep 値の WD_Sleep() の精度を改良。

修正:

- PCI/PCMCIA/ISA: clean-up 機能を修正および改良: WD_CardCleanup() / DC_CardCleanupSetup() のパフォーマンスを改良し、WinDriver アプリケーションが終了する際に、最初にこれらの関数を呼ぶように clean-up コマンドが確実に設定。
- DMA: Linux でアラインされていない scatter-gather DMA バッファのロックを修正。
- Windows 64-bit): 64-bit システムで 32-bit アプリケーションを起動時に発生する DMA と割り込みの問題を修正。

v8.01 の変更点

強化:

- .NET USB (Windows): 同じ USB デバイスで複数のインターフェイスとの同時通信のサポートを追加。
- Windows: 対象のデバイスの INF ファイルがインストールされていない場合でも、PCI Express のコンフィグレーション スペースにアクセスが可能。
- Windows: INF ファイルまたはフィルタ ドライバでフィルタされたリソースを持つデバイスの処理を改良。
- Solaris: install_windrvr WinDriver インストール スクリプトのすべてのインストール トレースを remove_windrvr WinDriver アンインストール スクリプトで削除。
- Linux: wdreg ドライバ インストール スクリプトに auto オプションを追加し、ターゲット プラットフォームで undev サポートの自動検出を可能。

修正:

- DMA: 64-bit カーネル (Windows x64 / Linux x86_64 / Solaris SPARC64) で起動する 32-bit アプリケーションの DMA 同期化関数 (WD_DMASyncIo() / WDC_DMASyncIo() / WD_DMASyncCpu() / WDC_DMASyncCpu()) の使用を修正。
- Linux - 一部の Linux カーネル 2.4 での WinDriver のインストールを修正 (v8.00 の問題)。
- PCMCIA (Windows): WDC_PcmciaSetWindow() の動作を修正 (v7.00 の問題)。
- .NET (Windows): .NET Framework 2.0 の PCI イベントの処理を修正 (v8.00 の問題)。
- .NET (Windows): Fixed the .NET implementation of the WDU_GetLangIDs() と WDU_GetStringDesc() 関数の .NET の実装を修正。
- .NET (Windows): PLX MSDEV 2003 .NET ソリューション ファイルを修正 (v8.00 の問題)。
- .NET (Windows): PCI C# サンプル -- エラーのイベントでの PCI_Device.Open() の動作を修正 (v8.00 の問題)。

v8.00 の変更点

新規:

Windows: Intel EM64T および AMD64 アーキテクチャ ("x64") で、Windows XP Pro x64 および Windows Server 2003 x64 をサポート。

- Windows: x86 および EM64T で Visual Studio 2005 をサポート。サンプルおよび Driver Wizard で生成されたコードには、C、C# および Visual Basic .NET プロジェクトが含まれます。

  注意: このサポートの追加によって、WinDriver C および .NET サンプルと Driver Wizard で生成されたプロジェクトのディレクトリ構造が変更されました。
- Linux: Intel Itanium および Itanium 2 アーキテクチャ (IA64) をサポート。
- Linux: Linux カーネル 2.6.14 をサポート。
- Solaris: Solaris 10 および OpenSolaris をサポート。
- WinDriver USB Device (Firmware Development Kit): Philips PDIUSBD12 用のデバイス ファームウェアの開発をサポート。
- WinDriver USB Device (Firmware Development Kit): Microchip PIC18F4550 開発ボード用の mass storage class をサポート。
- .NET (Windows): Driver Wizard で生成されるコードに C# (USB と PCI) と Visual Basic .NET (USB) を追加。
- .NET (Windows): C# の .NET PCI のサンプルを追加。
- USB: Philips PDIUSBD12 のサンプルを追加。

強化:

- Windows USB: 同一の USB デバイスの複数のインターフェイスとの同時通信をサポート。

- DMA: Windows、Linux および Solaris で 64 ビット DMA アドレスをサポート (新しい DMA_ALLOW_64BIT_ADDRESS DMA フラグ)。
- .NET (Windows): .NET アプリケーションからの Kernel PlugIn ドライバとの通信をサポート:
  - Kernel PlugIn ドライバへのハンドルを開く新しい WDC_KernelPlugInOpen() 関数。
  - Driver Wizard で、Kernel PlugIn ドライバと通信するユーザーモード アプリケーションのコードを生成。
  - PLX .NET ライブラリ API を修正し、Kernel PlugIn ドライバとの通信をサポート。
- Driver Wizard: Driver Wizard で生成されたコードで、レジスタの read / write 関数を Kernel PlugIn から呼ぶオプションを追加。
- WinDriver のライブラリ、DLL および共有オブジェクトのバージョニングを追加。
  - **wd_utils.dll** -> **wdapi<version>.dll** (例、**wdapi800.dll**)
  - **libwd_utils.so** -> **libwdapi<version>.so** (例、**libwdapi800.so**)
  - **wdapi_dotnet.dll** -> **wdapi_dotnet<version>.dll** (例、**wdapi_dotnet800.dll**)
  - **kp_nt.lib** -> **kp_nt<version>.lib** (例、**kp_nt800.lib**)
  - **kp_linux.o** -> **kp_linux<version>.o** (例、**kp_linux800.o**)
  - **kp_solaris.o** -> **kp_solaris<version>.o** (例、**kp_solaris800.o**)
- WinDriver のインストール ディレクトリ以下の WinDriver の source ファイル ディレクトリの場所を移動。
  - WinDriver の C のソース ファイルを src/ ディレクトリから src/wdapi.net/ ディレクトリへ移動。
  - WinDriver の .NET のソース ファイルを wdapi.net/ ディレクトリから src/wdapi.net/ ディレクトリへ移動。
- DMA: 双方向の DMA バッファのロックをサポート (新しい DMA_TO_FROM_DEVICE DMA フラグ)。
- PCI: トポロジで PCI バスをスキャンするオプションを追加 (新しい WDC_PciScanCardsByTopology() 関数を追加)。
- USB - Windows CE: Windows CE 5.0 で WDU_ResetDevice() をサポート。
- Solaris: 特定のカードに関連してない contiguous DMA buffer の割り当てをサポート (他の OS では、すでにサポート済み)。

修正:

- USB: アクティブな alternate setting だが、異なる interface として、同じ alternate setting number への set interface を修正。
- USB - Windows: WDU_Transfer() および WDU_TransferIsoch() で、isochronous パイプの 0 サイズ転送を禁止。
- USB - Windows: isochronous パイプをリスニング中に休止状態から起動する際に Windows が再起動する問題を修正。
- WinDriver USB Device (Firmware Development Kit): Microchip PIC18F4550 コードのダブル バッファの問題を修正。
- Delphi (Windows): サンプルおよび Driver Wizard で生成された Delphi のコードを修正および改良。
- Linux: Kernel PlugIn ドライバを使用中に Kernel PlugIn ドライバの削除を防止。
- Linux: IOMMU mapped DMA バッファのサポートを改良。
- 割り込み: 遅延した割り込み処理の実行が mask コマンドの結果に依存するように割り込みのマスクを修正 (v7.02 からの問題)。

v7.02 の変更点

新規:

- WinDriver USB Device (ファームウェア開発キット): Microchip PIC18F4550 開発ボードをサポート
- PCI 割り込み: カーネル モードの割り込み転送コマンドで割り込みマスクを定義するオプションを追加
- USB Microchip PIC18F4550 開発ボード:
  - ファームウェア *.hex をロードおよびデバイスをプログラムする bootloader ファームウェアのサンプル bootloader_demo を追加
  - ボードの WinDriver 診断ライブラリ (pic18f4550_diag_lib) を追加
- Linux: x86 64-bit プラットフォームで Linux カーネル 2.4.x の後方サポートを追加

強化:

- Linux: Linux カーネル 2.6.13 のサポートを追加
- WinDriver USB Device: I2C インターフェイスを初期化、アクセス、および Cypress EZ-USB FX2LP CY7C68013A 開発ボードの digit LED をコントロールするための API を追加
- .NET (Windows): PLX C# ライブラリ (plx_lib_dotnet)、およびサンプル診断アプリケーション (PLX_Sample) を強化
- PLX: 新しい割り込みマスク機能を使用するために PLX ライブラリ (plx_lib - C ; plx_lib_dotnet - C#) の割り込み API を修正
- DriverWizard (Windows): DriverWizard と .wdp ファイルの自動関連付けを追加
- PCI/PCMCIA/ISA: カードの memory/IO 領域でシェアしていない重複のロックを防止
- PCI/PCMCIA/ISA (Windows NT 4.0/2k/XP/Server 2003 および CE): 新しい WD_ITEM_ALLOW_CACHE フラグを使用して、物理 PC メモリ (RAM) をキャッシュとしてマップすることが可能に
- 64-bit Linux および Solaris: 64-bit WinDriver に 32-bit バージョンの wd_utils 共有オブジェクトを追加

- Linux: SuSE 製品をサポートするように WinDriver の設定ファイルを更新
- USB: Microchip PIC18F4550 ライブラリ (pic18f4550_lib) と default_firmware_demo サンプルを修正、強化し、場所を library に移動
- USB (Windows): デバイス列挙の問題、USB デバイスのエンドポイント記述子が不足している問題を修正

修正:

- DriverWizard USB: Windows で、低およびフル スピード デバイスのアイソクロノス パイプの Listen を修正
- PCI: WDC PCI configuration space API のカーネル プラグイン サポートを追加
- USB on Linux: WDU_Transfer() にタイムアウト 0 を設定した場合に、すぐにタイムアウトしていたのを無限に変更
- Linux: WD_Sleep() を使用しての non-busy スリープの CPU 消費を最小に軽減
- Windows: Microsoft の DriverVerifier ユーティリティ (verifier.exe) を WinDriver と同時に実行した際にエラーが発生する問題を修正
- .NET (Windows): wdapi_dotnet.cpp ファイルを wdapi_dotnet DLL のプロジェクト ファイル (wdapi_dotnet.vcproj) に追加
- .NET (Windows): C# .NET PLX ライブラリのプロジェクト ファイル (plx_lib.csproj) が diag_lib.cs ではなく log.cs にポイントするように修正
- WDU_Transfer() がバッファ ポインター (pBuffer) が NULL の場合に 0 以外のバッファ サイズ (dwBufferSize) を指定できないように修正
- Windows 98/Me: 古い windrvr6.sys ドライバをアンインストールすることなく、古いバージョンの WinDriver を置き換えることが出来るように wdreg16.exe を修正

v7.01 の変更点

新規:

- .NET (Windows):
  - C++ WinDriver DLL (wdapi_dotnet.dll) 用の新しい .NET Managed Extension (管理拡張) を追加
  - 新しい PLX .NET C# DLL (plx_lib_dotnet.dll) および GUI サンプル (PLX_Sample) を追加
  - 新しい USB .NET C# DLL (usb_lib_dotnet.dll) を追加
- USB: Microchip PIC18F4550 開発ボード用のサンプルを追加
- WinDriver USB Device (Windows): サンプルと生成されたコードにベンダー コントロール リクエスト用のフレームワークを追加
- Solaris: CPU および I/O キャッシュの DMA 同期化を行う API を追加
- Kernel PlugIn (PCI/ISA) on Solaris: スピンロック、インターロック操作を含む、シングルおよびマルチ CPU システム用の同期化 API を追加

強化:

- Windows CE: 生成された DriverWizard コードには Platform Builder 5.0 XML プロジェクト ファイルが含まれます
- Solaris/Windows CE: 新しい wd_utils DLL/共用オブジェクト、WDC API の使用法を含む、Windows および Linux のサンプルおよび DriverWizard で生成されたコードに基いて、サンプルおよび DriverWizard で生成されたコードを修正および強化
- Linux/Solaris: WinDriver の配布ファイルに wd_utils makefile を追加
- Windows: .NET USB C# と Visual Basic のサンプルを修正および強化。サンプルでは、WinDriver v7.00 の wdnetlib.dll DLL を置き換えた新しい wdapi_dotnet.dll と usb_lib_dotnet.dll DLL を使用します。
- PCI/PCMCIA/ISA: DriverWizard で生成された C のコードは、Wizard で定義したすべてのローカル レジスタの列挙を定義し、これらのカスタム レジスタをリード/ライトするための関数を含みます
- PLX: Scatter/Gather DMA の連続使用を可能にするため、PLX_DMAStart() ライブラリ関数を修正
- サポートされる OS 間でコードの移植を容易にするために、サンプル、DriverWizard で生成された C のソース コード および makefile を改良

修正:

- WinDriver USB Device (Windows): バルク データ トグル同期化を修正 (v7.00 の不具合)
- USB on Linux: 高速アイソクロノス転送を修正
- USB on Linux カーネル 2.4: 割り込み転送のメカニズムを修正
- USB on Windows: デバイスにリード オンリー データ バッファの書き込みを可能にするために WDU_Transfer() を修正
- USB on Windows: デバイス記述子が不正な場合の、システムの停止、クラッシュを解消することにより、悪いファームウェアの影響を軽減します
- Windows CE: プロセスの他のスレッド (割り込み処理を含む) からメイン スレッドのメモリ スペースへアクセスが可能
- PCI/PCMCIA/ISA: 新しい WDC_CardCleanupSetup() 関数を追加し、WDC_xxxDeviceOpen() の pDeviceCleanup パラメータを予約パラメータに置換 (v7.00 の不具合)
- USB on Linux: マルチ インターフェイスを持つ USB デバイスのサポートを修正

- PCI/PCMCIA/ISA: DriverWizard で生成されたコード、pci_diag および pcmcia_diag のサンプルでグローバル ハンドルを使用しないように修正
- Windows 2K/XP/Server 2003: すべてのプラットフォームで "Start Device" メカニズムを有効に修正
- wd_utils: windrvr_hotswap.c API への依存性を解消 (v7.00 の不具合)
- USB: サンプルと生成された USB コードへのマイナー修正。処理が失敗し失敗した場合に、main() が適切なエラー コードを返すように修正
- C++ モードでサンプルおよび DriverWizard で生成された C のコードをビルドする際に発生するリンク エラーを修正
- Windows 98/Me: WinDriver のライセンスの登録を修正 (v7.00 の不具合)
- PCI/PCMCIA/ISA: DriverWizard で生成された C の割り込みコードを修正。以前のコードでは、Wizard で割り込みに割り当てた最初のレジスタを常に使用していた問題を解消 (v7.00 の不具合)
- USB on Windows: デバイスを接続した後すぐに切断するとクラッシュすることがある問題を修正
- USB on Windows CE: Windows CE が起動する前に、ターゲット プラットフォームに USB デバイスが接続されていると、デバイスにアクセスできないことがある問題を修正 (v6.23 の不具合)
- USB on Windows: 古い WinDriver の WD_UsbScanDevice() 関数との互換性問題を修正。関数が WD_SINGLE_INTERFACE を返すべきところを実際のインターフェイス ナンバーを返すという問題。(v7.00 の不具合)
- USB on Windows: 実際の転送が開始される前に転送がタイムアウトした場合にシステムが停止する問題を修正
- Windows CE: レジストリの PCI バス ドライバに登録されていないカードを呼ぶと WD_DMALock() が失敗するという問題を修正
- USB on Windows: DriverWizard がデバイス リストを作成する時にデバイスが使用されている場合に、DriverWizard が WinDriver に登録されている USB デバイスを WinDriver と動作しない設定であると認識する問題を修正
- DriverWizard: Wizard で、割り込みに対して、同じレジスタへ複数のアクセス コマンドを割り当てた場合、最後のコマンドのみが *.wdp DriverWizard プロジェクトに保存されるという問題を修正
- DriverWizard: パラレル ポートのコードの生成を修正 (v7.00 の不具合)

v7.00 の変更点

新規:

- WinDriver USB Device: Cypress EZ-USB FX2LP CY7C68013A 開発ボードのデバイス ファムウェアの開発をサポート
- PCI Express の拡張スペースへのアクセスをサポート
- Linux: x86 64 ビット アーキテクチャをサポート: Linux カーネル 2.6.x で AMD64 と Intel EM64T ("x86_64")
- Linux PowerPC: Linux カーネル 2.6.x をサポート
- WinDriver Card (WDC) Library: PCI、PCMCIA、CardBus、ISA、ISAPnP、EISA、CompactPCI および PCI Express デバイス ドライバの開発用に新たなラッパー API を追加
- Windows: C# および Visual Basic .NET の .NET USB サンプル プログラムを追加
- PCI/PCMCIA (Windows): CPU と I/O キャッシュの DMA 同期化を行う API を追加
- PCMCIA: PCMCIA バス コントローラの設定を編集するAPI を追加

強化:

- Windows: WinDriver のインストールおよびアンインストール手順を改良
- サンプルおよび DriverWizard で生成された Linux のプロジェクトと Windows Borland C++ Builder のプロジェクトで wd_utils DLL を使用 (Windows: wd_utils.dll; Linux: libwd_utils.so)
- PCI/PCMCIA/ISA 用のサンプルおよび生成されるコードの改良 (新しい WDC API を使用)
  - PLX: 拡張サポートの PLX チップセット - 9030, 9050, 9052, 9054, 9056,9080, 9656 用に新しい共有ライブラリ (plx_lib) を追加
    およびこれらのチップセット用のサンプルも改良
  - PCMCIA 診断サンプル プログラム (pcmcia_diag) を追加
  - 共有 PCI/PCMCIA/ISA の C 診断 API (diag_lib, wdc_diag_lib) を追加
  - Kernel PlugIn PCI サンプル プログラム (kp_pci) (以前の Kernel PlugIn サンプル プログラム kptest と置換) を追加
  - DriverWizard で生成される C のコードを改良
  - 汎用的な PCI 用の C のサンプル プログラムを編集 (pci_diag, pci_diag, pci_scan, pci_dump)
- PCI/ISA/PCMCIA: PCI/ISA/PCMCIA のメモリ リソースをカーネルモードへマップするのを回避するオプションを追加 (リソースをユーザーモードの仮想アドレス空間へのみマップする)。これで、システムに負荷を掛けずに、大きいサイズのメモリ範囲の登録が可能。
  注意: デフォルトでは、WinDriver は、デバイスのメモリ リソースをユーザーモードとカーネルモードの両方にマップします (以前のバージョンでは、そのように行っていました)。
- Linux PowerPC: メモリ転送用に、WinDriver の転送 API (WD_Transfer() / WD_MultiTransfer() / WDC read/write API) が自動バイト オーダー転送を実行 (ビッグ / リトル エンディアン)

修正:

- Linux: Linux カーネル 2.6.x のサポートを修正および改良

- Linux: メモリ リークを修正
- USB (Windows): WDU_Transfer() / WDU_TransferIsoch() を USB_ISOCH_NOASAP フラグで呼び出した際に発生した Windows のクラッシュを修正 (v6.23 の不具合)。
- USB (Windows): DriverWizard または usb_diag サンプル プログラム、DriverWizard で生成されたコードでアイソクロノス パイプの Listen を実行した際に、以前では、特定の USB host controller および高速デバイスで失敗していたのを修正
- USB: usb_diag サンプル プログラムおよび DriverWizard で生成されたコードの DeviceAttach() 関数で、接続されたデバイスの有効な代替設定 (alternate setting) を使用するように修正 (以前のバージョンでは、代替設定 (alternate setting) 0 を使用)。
- USB (Linux カーネル 2.4.x): 8 フレーム以下のインターバルで EHCI 割り込み転送で発生するクラッシュを修正

v6.23 の変更点

強化:

- PCMCIA: PCMCIA attribute space への書き込みの実装を追加

修正:

- WinDriver USB for Windows
  - SET_INTERFACE コマンド メカニズムを改良
  - Windows USB ドライバのアンロード メカニズムを改良
  - Windows USB: USB デバイスの列挙中に USB デバイスの切断をサポート
- HID: 時間制限なしで読み込み要求をできるように読み込みメカニズムを改良
- Windows: マイクロソフト社の WHQL テスト メカニズムとの互換性を向上

v6.22 for Windows CE 6.22 の変更点

新規:

- Windows CE 5.0 のサポートの追加
- Windows CE で USB 2.0 のサポートを追加

強化:

- プラグ アンド プレイ サポートの向上
- サンプル: すべてのサンプルでプラットフォーム ビルダをサポート
- 割り込みレイテンシ (遅延時間) の向上
- USB: USB 1.1 と 2.0 のアイソクロナス転送を飛躍的に向上
- USB: レジストリで USB デバイスの永続的な登録をサポート

v 6.22 の変更点

新規:

- Linux カーネル 2.4 で PowerPC のサポートを追加
- Silicon Laboratories の C8051F320 USB ボードの拡張 USB サポートを追加
- Windows 2000/XP/Server 2003 で PCMCIA カードのサポートを追加

強化:

- Linux: ユーザーの選択により、ユーザーが使用している特定のカーネル、またはインストールされている別のカーネル ソースのいずれかを基に makefile を作成する新しいインストール メカニズム
- デバッグ: ユーザーがすべてのデバッグ セクションをワン クリックでチェック、またはクリアできるように、Debug Monitor ユーティリティ (wddebug_gui) に All sections チェック ボックスを追加
- デバッグ: ユーザーが PnP セクションを単独でモニタできるように、Debug Monitor ユーティリティ (wddebug_gui) に Plug-and-Play デバッグ セクションを追加
- PLX: PLX 9054 互換ボードのソフト リセットのサポートを追加
- PLX: PLX 9030、9054、9056、9656 サンプルで VPD メカニズムを通しての EEPROM アクセスのサポートをアップデート
- PLX 9054: サンプルに DMA_LARGE_BUFFER のサポートを追加
- ウィザード (PCI/PCMCIA/ISA): それぞれのレジスタに対してオフセットの表示を追加
- USB: デバイスのサポートする言語 ID および/またはサポートする言語 ID の数を返す新しい API 関数 WDU_GetLangIDs() を新しく追加
- USB: デバイスからストリング記述子を読み込む新しい API 関数 WDU_GetStringDesc() を追加
- PCI: マップされたメモリ範囲を使用して PCI デバイスにアクセスする際に、リトル/ビッグ エンディアンの問題を解消するエンディアン変換関数を追加。この関数はプラットフォームに依存しないデータの閲覧を可能にします。

修正:

- Solaris: WinDriver のハンドルを開いている "ゾンビ" プロセスを検出し、解放
- ウィザード (PCI/PCMCIA/ISA): コード生成に加えて、ウィザードは 64 ビットのレジスタも処理できるように修正
- USB: 以前のクラッシュを解消し、USB 転送を安定化
- PCI: マイナーな下位互換性問題を修正

v6.21 での変更点

強化:

- PCI/ISA: アプリケーション終了時にカード クリーンナップ操作を有効にするために API を追加
- デバッグ: コンソール モードの WDDEBUG の使用方法を簡素化
- Windows: wdreg.exe インストレーション ユーティリティを強化

修正:

- USB: マルチ インターフェイス USB デバイスの INF 生成、および DriverWizard からのインストール/アンインストールを修正
- 64 ビット プラットフォーム: ウィザードが PCI デバイスの I/O リソースを正常に表示するように修正
- Linux: 読み込みの時の DMA のロックにあった問題を修正
- 全プラットフォーム: PCI 設定スペースへのアクセスを改善し、オーバーヘッドを削減
- Windows USB: メモリ リソースが不足した場合にクラッシュする問題を修正し、WinDriver を終了してエラー メッセージを表示するように変更
- Solaris: Kernel PlugIn と WinDriver のバージョンが一致しない場合にクラッシュする問題を修正
- Linux: 古いバージョンの Linux が WinDriver のグラフィックをサポートしなかった問題を修正

v6.20 での変更点

新規:

- SPARC 上で 32 ビットおよび 64 ビット Solaris 8-9 をサポート
- x86 上で 32 ビット Solaris 8-9 をサポート
- 64 ビット Solaris kernel上で 32 ビット ユーザー アプリケーションをサポート
- Linux カーネル 2.6 のサポート
- PCI デバイスがまだ完成していない段階でドライバの開発を開始するオプションを追加
- ウィザードに基本の API ロギング機能を追加
- Texas Instruments の TUSB3410、TUSB3210、TUSB2136、TUSB5052 チップ用の拡張 USB サポートを追加 (ファームウェア サンプルをダウンロードしてください。)

強化:

- USB: USB 1.1 と 2.0 のアイソクロナス転送を飛躍的に向上
- SPARC Solaris DMA: SPARC プラットフォームでユーザー モードに Scatter/Gather DMA と連続カーネル バッファのサポートを追加
- x86 Solaris DMA: x86 プラットフォームで Scatter/Gather DMA と 連続バッファ DMA のサポートを追加
- Windows: パワー管理サポートの向上
- Windows: USB 解除メカニズムの向上
- USB: ウィザードに標準コントロール転送リクエストのサンプルを追加
- Windows: WHQL テストに通過する手助けをする *.sys と INF ファイルを改善
- USB: usb_diag サンプルと生成されたコードはマルチ デバイスを標準サポートします
- PLX サンプル: DMA 転送の実装のサンプル コードを p9054_diag.c に追加
- WDREG/WDREG16 ユーティリティ: INF ファイルへの相対パスの使用が可能
- コード: Kernel PlugIn Wizard の生成したコードは、ウィザードで割り込みに割り当てられたすべてのアクセス レジスタがメモリ レジスタであるとき、直接メモリ アクセスを実行します。
- コード: Borland C++ Builder の C++ モードでコンパイルした時の生成されたコードとサンプル コードを改善し、エラー、警告を削減
- USB: WDU_GetDeviceAddr などの特定のデバイス プロパティの取り出しを可能にする、新しい WD_GetDeviceProperty API 機能を追加 (Windows のみ)
- サンプル: 拡張されたサポート サンプルを含む、MSDEV .NET (7.0) プロジェクト ファイルをすべての WinDriver サンプルに追加
- MSDEV .NET 環境用にユーザー モード API ラッパー を DLL として提供
- DMA: WinDriver API に新しい DMA フラッグを追加し、データの一貫性を向上

- Wizard: License 登録をキャンセルするオプションを追加し、以前に登録ライセンスが使用されていたマシンで評価が可能

修正:

- Solaris: Sun の cc コンパイル警告を削減し、コンパイル エラーを修正
- PLX サンプル EEPROM: EEPROM が誤って通常のメモリ スペースとして扱われていた問題を修正
- INF: ウィザードの INF-create ダイアログが、自動的にマルチ インターフェイスの INF ファイルの作成を提案する問題を修正
- マルチ インターフェイス USB デバイス: ウィザードで登録されていないインターフェイスと登録済みのインターフェイスを表示するように修正
- WDREG ユーティリティ: WDREG ユーティリティでマルチ インターフェイス デバイスをアンインストール/有効/無効にできるように修正
- Windows Delphi USB: 生成されたコードとサンプル コードを改善
- USB 生成/サンプル C コード: 生成されたコードとサンプル コードを改善
- Windows Visual Basic および Delphi の ISA コード: 生成されたコードとサンプル コードを改善
- ウィザード: マイナーな GUI の修正

v6.03 での変更点

新規:

- Windows CE で ARM ベース CPU のプラットフォームをサポート
- USB リモート ウェイクアップ機能のサポート

強化:

- USB: Windows XP と Windows Server 2003 でバルク転送の速度を向上
- USB: デバイスのリセットとウェイクアップ関数の追加

修正:

- Visual Basic USB サンプルと生成されたコード: ベンダと製品 ID の値が 32,767 を超える場合、符号なしの値を使用するように修正 (V6.00 以降の問題)
- Visual Basic USB サンプルと生成されたコード: 生成されたコードのデフォルト代替セッティング番号が、以前は選択されたインターフェイス番号に設定された問題を修正 (V6.00 以降の問題)
- Visual Basic USB サンプルと生成されたコード: VB の マルチ スレッドと CreateThread() の機能不足による ListenToPipe() 関数とそれに関連する関数の欠陥を修正。(特に VB 6.0) ThreadStart/Stop Visual Basic の関数は VB_ThreadLoopStart/Stop DLL 関数に置き換えられ、Listen ループはユーザー スレッド関数の中ではなく DLL 内で実行されます。
- Delphi USB: usb_diag_lib.pas サンプルの PrintEndpoints() 関数で、エンドポイント データが不正確で、ゼロ エンドポイントが存在する場合にアプリケーションが強制終了される問題を修正。(V6.00 以降の問題)
- Delphi USB: DLL スレッド 関数で、Listen to Pipe 関数の問題を修正。(V6.00 以降の問題)
- Linux カーネル 2.4: 場合によって PCI カードのメモリ (リード オンリー 0xffffffff) からの読み込みに失敗する問題を、pci_enable_device() 関数を呼ぶことによって解消。
- Windows CE: スピン ロック、イベントの実装を向上。
- PLX サンプル: dwMask の計算ミスによりローカル アドレスからリード/ライトでアプリケーション エラーが発生する問題を修正。 (V6.00 以降の問題)
- VxD ドライバ: 98/Me で WDREG を使用してのインストールの問題を解消し、不正確なエラー メッセージが表示される問題を修正。 (V6.00 以降の問題)
- wdreg16.exe: Windows Me/98 プラットフォームでアプリケーションがクラッシュすることがあった問題を、makefile のスタック サイズを増加することにより修正。
- WHQL テスト: WHQL テストをパスするために、不正な IOCTL を受け取った場合に未知のバッファへのアクセスを防ぐためにコードを修正。
- WHQL テスト: WHQL テストをパスするために、プラグ アンド プレイ IRP ステータスを初期化。

v6.02 での変更点

新規:

- Windows Server 2003 のサポート
- Windows CE.NET 4.2 のサポート
- MIPS プロセッサ用の Windows CE サポート
- Delphi: サンプル コードとウィザード コード生成を含む、新しい USB API のサポート
- Linux カーネル 2.4.20 のサポート

強化:

- Windows: Debug Monitor (wddegug_gui または wddebug) の [Send debug messages to kerneldebugger] オプションに新しい機能を追加。この機能を有効にすると、次のブートで WinDriver が起動中に WinDriver デバッグ メッセージが Kernel デバッガに送信されます。
- Visual Basic PCI、ISA および USB のコード生成を向上。

修正:

- 接続されていないデバイスのドライバを更新するときに、wdreg は正常に実行されるがエラー ステータスのコードを返す問題を修正。
- Linux カーネル 2.4.20: コンパイル時の問題を修正。
- Linux: マルチ プロセッサ (SMP) システム Linux Scatter/Gather DMA でインストール時の問題を解消し、HIGHMEM サポートを正常に実行 (V6.00 以降の問題)。
- Linux: WD_DMALock() コールと小さな連続的 DMA バッファを繰り返し割り当てるとクラッシュする問題を修正。
- Linux: 同じ PC 上で、WinDriver V6.0x と WinDriver V5.22 またはそれ以前のバージョンを同時に使用した時に起こる問題を修正 (V6.00 以降の問題)。
- Wizard USB プロジェクト (*.wdp ファイル): Save (保存) オプションを削除。V6.00 よりこのオプションはサポートされていないが、誤って V6.00 と V6.01 に残っていたものを削除 (V6.00 以降の問題)。
- USB 生成コード: V5.22 またはそれ以前のバージョンで生成されたコードを V6.0x でリビルドすると、event_register() のアクティベートに失敗する問題を修正 (V6.00 以降の問題)。
- Windows: 前バージョン (V6.00 以降) で、まれに WD_UsbTransfer() と WD_IntWait() が不正なエラー ステータスを返し、システム クラッシュを起こす問題を修正。
- Windows: windrvr6.inf が Windows から IRQ を要求しないように修正し、1 バイトを採取するメモリ領域を拡張することにより、WinDriver のインストールに失敗する問題を修正 (V6.00 以降の問題)。
- Windows: API 機能 WDU_transfer() がキャンセル、またはタイム アウトで終了したときにメモリ リークが起こる問題を修正 (V6.00 以降の問題)。
- WDREG: 不要なメッセージを削除し、WDREG クラッシュの問題を修正 (V6.00 以降の問題)。
- Visual Basic ISA コード生成: コンパイル時のエラーを修正 (V6.00 以降の問題)。
- Visual Basic USB サンプルおよびコード生成: PrintDeviceConfigurations() ライブラリ機能で、メモリ リークが起こる可能性があった問題を修正。
- PLX9656 サンプル診断アプリケーション: DMA テストを修正 (V6.00 以降の問題)。
- V6.00 以前のバージョンで生成されたアプリケーションのポートの問題を軽減。
- WINCE.NET: このオプションを有効にすると WinDriver のデバッグ メッセージがすべてのケースで kernel デバッガに送られるように変更。

v6.01 での変更点

新規:

- wdreg ソース コードの提供。
- STMicroelectronics USB HID サンプル: USB/Serial ブリッジ (st/rs232u) および 92163 USB Evaluation Kit (st/92163)。

強化:

- INF ファイルをインストールする際にウィザードがフリーズする問題を修正。
- USB トランスファ ダイアログの File to Pipe オプションで大きなファイルを読めるようにウィザードを強化。
- Visual Basic で生成されたコード、サンプル コードの改善。

修正:

- デバイスを装着せずにドライバをアップデートすると、wdreg を正常に動作するがエラー ステータス コードを返すという問題を修正。
- Windows: USB 転送関数で短い USB バルク/割り込み転送が届いたときに、すべてのバッファが満たされるまで待機する問題を修正し、呼び出し元に返すように変更。
- Windows CE: CE のスタート アップ時に誤った dll (windrve6.dll ではなく windrvr.dll) をロードしようとする問題を修正。
- Windows 98: newdev.dll がユーザーのマシンに存在しない場合に、それをコピーするように修正。(この DLL は接続された USB デバイスの検知とこれらのデバイスの INF ファイルをインストールするのに必要です。)
- Linux USB: 転送に失敗した場合に、不正なパイプ タイプ検知により、割り込み転送がクラッシュするという問題を修正。
- Linux USB: Linux で時折不正なエラー ステータスを返す問題を修正。
- Linux USB: マルチ インターフェイスのデバイスのハンドリングの修正。
- Visual Basic: C DLL と VB アプリケーション間でのストリング転送のハンドリングを正すために、Stat2Str() VB 実装 (wd_utils.cls 内) を修正。
- Visual Basic USB: wd_utils.cls の関数宣言を修正。

- Visual Basic PCI: 生成されたコード: 関数に送られるライセンス ストリングの最後に NULL キャラクタを追加することにより、WD_License() を修正。これによりC DLL によって ANSI ストリングとして解釈されます。
- DriverWizard で USB Delphi コード生成は無効にされています。(v6.00 からサポートされません)

v6.00 での変更点

新規:

- Linux で USB のサポートを追加。
- Windows CE.NET で USB と PCI のサポートを追加。
- Visual Basic 6.0 用のコード生成機能を追加。(USB VB サンプルも追加)
- Visual Studio .NET プロジェクト生成のサポートを追加。
- USB HID デバイス用にユーザー モード API を追加。
- 全 OS で USB 用の API を向上。

強化:

- エラー ハンドリングを向上するため、すべての WD_xxx() 機能に戻り値を追加。
- Windows CE に DMA API のサポートを追加。
- 標準の配布に QuickLogic ファイルを追加。古い V3 ファイルと置換。
- 実行機能を include/windrvr_int_thread.h から src/ ディレクトリに移動。
- wd_vb.dll を wd_utils.dll と置換。
- WD_DMA 構造体に pKernelAddr を追加。連続バッファ DMA では、WD_DMALock() は dma.pKernelAddr に割り当てられたバッファのカーネル マッピングを返します。

修正:

- WDM Windows での windrvr.sys のインストール/アンインストール方法を変更。これにより、すべての OS で windrvr が windrvr6 に変更されました。windrvr6 は windrvr とバイナリ互換性がありません。
- Windows スタンドバイの状態から回復した後に、USB デバイスのインターフェイスの自動初期化を有効になるようにパワー変更操作を修正。
- Linux でウィザードと wddebug_GUI アイコンを修正。
- デバイスの登録に失敗したときに、Visual Basic のパラレル ポート サンプルが無限にメッセージ ボックスを表示する問題を修正。
- 範囲チェックが有効になっているときに Delphi のコンパイラ エラーが発生する問題を修正。
- 待機されていない semaphore_release で 0x8000000 割り込みが発生した後に winnt クラッシュが発生する問題を修正。

v5.22 での変更点

新規:

- gcc 3.x (例: RedHat 8.0) でコンパイルされた Linux カーネルのサポートを追加。
- 割り込みハンドラにユーザーがコードを追加できるように、windrvr_isr() hook を VxWorks に追加。
- BIOS によって PCI バスマスタをネイティブにサポートしない問題を解決するため、Linux 上で PCI バスマスタ機能を追加。

強化:

- USB 複合デバイスのコントロール転送機能のサポートを向上。
- WinDriver および KernelDriver のドキュメントの内容を修正、追加および再編集。

修正:

- Windows : Windows 95 でエラーが発生する WDREG ユーティリティのバグを修正。
- Linux : "make install" で間違ったファイルがインストールされる問題を解決するため、Linux 上で生成された Kernel PlugIn makefile を修正。これにより、makefile はインストール時に数々の問題を識別し、自動的にこれらを修正します。
- USB: アイソクロナス転送を行っているときに、USB デバイスを抜き取るとブルースクリーンが発生する問題を修正。
- USB : USB デバイスでブルースクリーンが発生する同期化の問題を修正。
- KernelDriver : Win32 プラットフォームで KernelDriver の機能、kd.lib の KD_OpenWinDriver() のバグを修正。この機能への最初の呼び出しのみ成功し、その後の呼び出しには失敗するという問題がありました。

v5.21 での変更点

強化:

- 割り込みを通知する転送コマンド、および割り込み通知のための転送コマンドを DriverWizard から設定が可能。
- デバッグ モニタ バッファ オーバーフローの強化。

- WD_DMALock に DMA_ALLOW_CACHE オプションを追加。Windows NT、2000、XP でロック メモリを CPU にキャッシュできるようにすることにより、DMA スピードを大幅に向上。(このオプションを使用するとデータが破壊される可能性があります)
- WinDriver はコンピュータに接続されていないデバイスの INF ファイルを自動的にインストールすることが可能。
- WINDRVR.SYS をロードする前に KernelDriver を起動することが可能。

修正:

- Windows 2000、XP でDMA を継続して実行したり、WD_DMAUnlock を呼ぶとメモリ破壊と BSOD が引き起こされる問題を修正。
- KernelDriver が 高い IRQL から WD_IntCount を 呼ぶと BSOD が引き起こされる問題を修正。
- 割り込み (2^31) が生成され、それを処理する登録されたアプリケーション/ドライバがない場合に BSOD が起こる問題を修正。
- Kernel PlugIn へのそれぞれの呼び出しが BSOD を引き起こす可能性があるバグを修正。(特に、Microsoft の Driver Verifier を使用してデバッグビルドしている場合)
- Cypress の EZ-USB ファームウェアのダウンロードが失敗する問題を修正。(特に再コンパイルの場合)
- Visual Basic および Delphi を使用している場合にコンパイル エラーが発生する問題を修正。
- Linux で誤ったバイト カウントを最初と最後のページに与える Scatter/Gather DMA 操作を修正。
- USB の生成されたコードで XXX_CountDevices 関数が失敗する問題を修正。
- Kernel PlugIn のインストールで、再起動後に Kernel PlugIn が起動に失敗する問題を修正。

v5.20 での変更点

新規:

- カーネルおよびユーザー レベルでプラグ アンド プレイとパワー通知イベント ハンドリングをサポート。これにより、標準の Windows Driver Model (WDM) 機能性に準拠します。
- x86 マシンでの 64ビット データ転送をサポート。この機能により、既存の 32 ビット Windows と Linux OS で新しい 64 ビット バスを利用することを可能にし、高機能のドライバを作成します。
- PLX Technology の新製品ライン、PCI 9656 I/O Accelerator チップのサポート。このサポートには PLX 9656 アクセスとコントロールを導入する特別な API セット、および skeletal ドライバとして使用できるサンプル アプリケーションを含みます。
- KernelDriver はすべての Windows OS 用の DDK を使用してのドライバの作成をサポート。また、Visual C++ 用のサポートを拡張。
- KernelDriver で Windows 98/ME/2000/XP 上の WDM を完全にサポート。
- ユーザー ドライバのコード作成で、コードのヘッダと関数本体のドキュメントを生成。

強化:

- wdpnp.sys と windrvr.sys を一体化し、1 つのドライバ ファイル (windrvr.sys) を作成することにより、より容易な操作性と配布。
- 評価版と登録版を一体化することにより、登録版への更新の簡易化。
- Linux 上で、デュアル プロセッサと SMP をサポート。
- Linux カーネル 2.4 で 分散/集束 DMA をサポート。
- USB オペレーションにステータス情報を追加。
- マルチ インターフェイス USB デバイス用の API を追加。
- Windows OS で DriverWizard から INF ファイルを自動的に生成、およびインストール。以前必要だったマニュアルのステップを削除。
- Borland C++ Builder 6 のコードを生成するオプションを追加。
- 生成された INF ファイルに機能性を追加。

修正:

- 共通: 使いやすさを念頭においた、様々な DriverWizard の変更。
- Windows: すべての USB 2.0 HC と マルチ インターフェイス デバイスをサポートする USB スキャン プロセスを再実装。
- Windows: ファイルからデータを USB パイプに転送できるように USB file to pipe オプションを修正。
- Windows: ハイ スピード デバイスを認識するために、usb_scan ユーティリティを修正。
- Linux: CTRL-Z を押すとファントム インターラプトになる問題を修正。
- Solaris: Solaris 2.6 で存在しない Kernel PlugIn を開こうとするとシステムがクラッシュする問題を修正。

v5.05 での変更点

新規:

- PLX Technologies 新製品 PCI 9056 I/O アクセラレータ チップをサポート。このサポートは、PLX 9056 アクセスおよびコントロールを実行する API 関数および雛型として仕様できるサンプル アプリケーションを含みます。

強化:

- APIC (Advanced Programmable Interrupt Controller) チップを実装した Windows NT マシン上の割り込み処理をハードウェアから直接読み込む代わりに OS から PC I関連データを読み込むように強化。

修正:

- 誤った環境で警告メッセージ "The resource is overlapped with another resource (リソースが他のリソースと重複しています)" が表示される DriverWizard のバグを修正。
- Kernel PlugIn を使用するとき、割り込みがイネーブルされた直後発生した場合 Kernel PlugIn の DPC ルーチンで割り込みが処理されないことがあるバグを修正。
- Windows: DriverWizard が生成した INF ファイルを特定の PCI デバイスで使用するとき BSOD を引き起こすバグを修正 (v5.04 からのバグ)。
- Windows: Windows 95 で Kernel PlugIn プロジェクトをコンパイルするために必要な KP_95.lib が見つからないバグを修正 (v5.04 からのバグ)
- Unix: DriverWizard により いくつかの Unix 環境変数が上書きされるバグを修正。

v5.04 での変更点

新規:

- Windows 2000、Windows XP で USB 2.0 サポートの追加。
- WinDriver Hardware Debugger が OS のカーネル デバッガにデバッグ メッセージを送信する機能を追加。
- ファイルのプロパティで表示されるバージョン情報を wdpnp.sys に追加。
- WinDriver ユーザーズ ガイドに新しい関数リファレンスが追加されました。
- Windows 2000、Windows XP で USB 2.0 サポートの追加。

強化:

- WinDriver for USB: 新しい API: WD_UsbResetDeviceEx() を使用して無効になっていないデバイスのリセットが可能。
- WinDriver for USB: WD_UsbTransfer() に USB_SHORT_TRANSFER のオプションを追加。このオプションを設定すると、ショート パケット (最大パケット サイズ以下) がトランスファーされた時に、データ バッファが満たされたかどうかに関係なく、その関数はリターンします。
- WinDriver for USB: WD_USBTransfer() に USB_FULL_TRANSFER のオプションを追加。このオプションを設定すると、リクエストされたすべてのデータがトランスファーされた時に、その関数はリターンします。
- Warning level 4 で DriverWizard によって生成されたコードをコンパイルするときの、Warning (警告) メッセージの数を削減。
- DriverWizard よって生成されるコードの説明および注意が改善されました。

修正:

- PLX 9030、9054、IOP480 チップセットの EEPROM へのアクセスの改善。これらのチップセットの EEPROM へのアクセスに失敗すると、WinDriver はすべてのアクセス プロセスを再試行します。
- DriverWizard による割り込み操作の改善。ユーザーによって定義されたすべてのパラメータをインプリメントせずに割り込みを操作するバグを修正。 。
- 割り込みトランスファー コマンドの実行方法の改善。割り込みが無効にされるたびに割り込みトランスファー コマンドが実行されるバグが修正され、割り込みトランスファー コマンドは、アプリケーション操作の割り込みが不正終了する場合にのみ実行されます。
- Windows 2000 checked-build の環境下で、WinDriver がアプリケーション アサーションを発行するバグを修正。
- Delphi をサポートするように WD_MultiTransfer() を修正しました。
- 関連する PCI デバイスが I/O またはメモリ リソースがないときに XXX_AccessRanges(hXXX) の生成を無効にするよう DriverWizard を修正しました。
- Windows NT 4.0: デュアル CPU プラットフォームでの割り込み操作のサポートの修正。割り込みラインとデバイスのリソース情報は、(ハードウェアから直接ではなく) OS から読み取ります。
- Windows 9x: 共有不可の割り込みに対して Auto Masking を設定。共有可能な割り込みを使用している時に、WD_IntDisable() が他の割り込みも無効にする場合があるバグを修正。
- Linux - Linux の環境で、連続する DMA の WD_DMALock() の操作を修正。アロケーションがページ単位の場合に余分な 1ページが作成され、kfree の内でカーネル パニックが発生 (以下の順序で実行された場合: lock、lock、free、free) するバグを修正。
- RedHat Linux 7.0 またはそれ以上でコンパイルのために user makefiles の修正。

- Windows: windrvr.sys と wdpnp.sys 間の依存問題の修正。wdpnp.sys が windrvr.sys/vxd の前にロードされると、windrvr がエラー コード 37 (STATUS_INTERNAL_ERROR) を発生するバグを修正。このバグの修正でリソース管理は windrvr.sys/vxd の代わりに wdpnp.sys が操作するので、wdpnp.sys の後に windrvr.sys/vxd がロードされると、wdpnp.sys の代わりに windrvr.sys/vxd がカード リソースのトランスファーを行います。また wdpnp.sys は windrvr.sys がロードされていると仮定し、二つのファイルがコミュニケーション パスを開いて短い操作を行って、そしてそのパスを閉じるということを行いません。これによりファイル間の依存性を削除し、安定性を強化します。
- Windows: Windows のプラットフォームでコンボ ボックス作用が呼び出される問題を修正。デフォルトで呼び出す IDE が "none" に設定されるバグを修正。この修正で呼び出す IDE は、プロジェクト ファイルを生成するためにユーザーによって選択された IDE に設定されます。デフォルトの設定は MS Visual C++ 6.0 です。
- WinDriver USB: DriverWizard の "send to pipe" 操作でロング データ ストリング (1024 文字以上) が入力できるように修正。ロング データ ストリングが送られたときに、DriverWizard がアプリケーション エラーを発生するバグを修正。
- WinDriver USB: USB デバイスがシステムに接続されていない場合に、BSOD を回避するようにタイムアウトの期限を修正。
- WinDriver USB: WinDriver は OS の Pointer to Device Object (PDO) を使用する代わりに、OS の DeviceAddress を USB uniqueId として使用します。WinDriver は WD_GetConfiguration() と WD_DeviceRegister() のステージで USB デバイスと通信するための目的でのみ PDO を使用します。
- WinDriver USB: アイソクロナス データ転送では、WinDriver は最後のパケットを別々のトランザクションとして送ります。ある環境では、OS USBD が最後のパケットを2回送ることがありました。
- Kernel Driver: Debug Monitor GUI が Linux でも実行できるように DriverWizard を修正。
- Kernel Driver: WD_License() を実行するために IOCTL を使用してコードを生成するように DriverWizard を修正。

v5.03 での変更点

新規:

- WinDriver は、Windows CE 3.0 をサポート
- WinDriver と KernelDriver は、Windows XP をサポート
- WinDriver と KernelDriver は、Linux カーネル 2.4.x をサポート

強化:

- Windows 版でのみ提供されていた Debug Monitor (wddebug_gui.exe) が Linux 版と Solaris 版でも利用可能
- WD_DevugAdd() の新 API 追加: デバッグメッセージを印刷可能なデバッグログに送ることができます。
- Linux、Solaris、VxWorks および CE 版の評価版での連続稼動時間制限を 30分から 1 時間に延長。
- USB デバイスのユニーク ID がデバイス オブジェクトへのポインタ (Windows 98/200/XP(RC1) から受け取る)になります。
- DriverWizardの Quick Start Info ウィンドウの表示

修正:

- (V5.00 以降) DriverWizard によるレベル依存割り込みの認識。修正前はレベル依存割り込みの認識時にシステム がハングすることがありました。
- PLX9030 ライブラリの割り込み認識レジスタ(INTSCR)の修正。これレジスタは、16bitではな く32bitとしてアクセスされていました。
- デュアルCPUでの割り込みハンドリング(Windows 2000)。割り込み値を読み込むため PnP WDM 関数を実行します。
- デバイスを取り外したときにポップアップした"Unsafe removal of device"メッセージを削除(Windows 2000)。
- pthread 使用時のメモリリーク(Linux および Solaris)。オープンされたthreadが正しくクローズされず、新規thread のオープンを妨害する可能性がありました。
- (V5.00以降) Kernel PlugIn 関数の funcIntAtDpc を"top half" ではなく"bottom half"で実行するように修正(Linux)。
- InterruptThreadEnable() で作成されたthreadからのメッセージをprint。デバイスのターミナルに代わってコ ーリング shell で表示(VxWorks)。

v5.02 での変更点

強化:

- Wizard が INF/KDF ファイルを生成する際に選択された名前の有効性をチェックします。
- 追加で割り込み/メモリ/IO を作成するときに既存のリソースと重複する場合は警告します。
- PCI ベンダ ID および デバイス ID リストの説明が最新情報に更新されました。
- USB read/write ダイアログボックスに [add pipe to file] と [save input] が追加されました。
- すべての read/write ダイアログボックスに [close] ボタンが追加されました。

修正:

- license string ダイアログで入力ストリングが半角 60 文字以上の場合、"Invalid license"となる不具合を修正。
- リソースを定義した内容によって、read または write 操作を許可するように修正。

- wddebug メッセージに選択された debug level と scetion を追加。
- 生成された Delphi コードの xxx_RegisterWinDriver() をコンパイルの不具合を修正。
- プロジェクトを保存するとき、プロジェクトを閉じたあと、ウィザードは新規のレジスタの属性をデフォルト値に戻しません。
- 既知の不具合: 特定のオフセットを定義されたレジスタ上で割り込みを設定するとき、割り込みからのデータに 0 が書き込まれます。
- USB デバイスの登録は1つのアプリケーションのみ許可するように修正。
- 転送に失敗した終了中の USB アプリケーションに "PROCESS_HAS_LOCKED_PAGES" を表示するブルー スクリーンを表示しません。
- その他、若干のマイナーバグを修正。
- 記述もれの WD_UsbResetDevice() を追加。

v5.01 での変更点

新規:

- リモート pci_dump (rpci_dump)サンプルの追加
- WinDriver と KernelDriverのヘルプファイルの起動
- マニュアルにリモートドライバ開発手順の追加

強化:

- Webダウンロード版でもフル機能を装備(ドキュメント類の一部は含まれません)。KernelDriver のWebダウンロード版はクラスおよびコンパイル済みのデバッグライブラリのソースを含んでいません。
- DriverWizardの機能強化
- 1MB以上のバッファの Scatter Gather DMA の説明を修正

修正:

- DriverWizardのいくつかのマイナーな不具合を修正
- Wizardで生成された PCI INF ファイルは、wdpnp.sys を %WINDIR%¥system32¥drivers にコピーします。
- Registered 版の WinDriver for USB コード生成で USB デバイスをスキャンする前に WinDriver registration をコールするように修正(以前は、コードが Usb_scan で失敗しました)。
- WinDriver for USB で、USB パイプへの Writeで ASCII データでなく HEX データを Writeするように修正。
- Windows Net.lib に Remote Access 機能で必要なすべての関数を含めました。
- Windows Net.lib 上で Remote Access 機能のために必要なすべての関数を含みます。
- Solaris (Sparc)で、PCI メモリ空間から読み込むときのバイト順を修正。
- USB のために Delphi が生成したコードのコンパイルの不具合 (fRead をDWORD でなく BOOLEAN として宣言するのでコンパイルできません) を修正。

v5.0 での変更点

新規:

- グラフィカルなユーザー インターフェイス (GUI) を WinDriver for Linux と WinDriver for Solaris に追加。
- リモート アクセス機能が WinDriver Wizard に統合。Remote Host 機能が Windows NT/2000、Linux および Solaris 上で現在サポートされています。リモート ターゲット機能 はWindows 95 / 98 / Me / NT / 2000 / CE,、VxWorks、 Solaris および Linux 上でサポートされています。
- Linux/Solaris / VxWorks / WinCE 上でのセッションごとの評価タイム アウトが30分に延長。
- PDF でリファレンス セクションにサムネール、ブックマークおよび相互参照のハイパーリンクを提供。HLP (WinHelp) では 、相互参照ハイパーリンクを提供。
- DriverWizard で、いくつかレジストリの読み込み時の問題を修正。

強化:

- Windows 用 DriverWizard ウィザードで WinDriver ヘルプ ファイルと KernelDriver ヘルプ ファイルの選択が可能。

修正:

- Windows 2000で DriverWizard により生成されたデータは Windows 2000 の推奨に準拠。
- DriverBuilder (VxWorks 用) で MPC860/MBX860 および PPC750/MCP750 のためのリソース割り当てを修正。
- Solaris (Spark および x86) での crash on WD_CardRegister() 上のクラッシュを修正。
- KernelDriver USB が Windows 2000 用の C および C++ SYS ファイルを生成。

テクニカルサポート
エクセルソフト Web ページ
https://www.xlsoft.com/jp/qa/

同封のユーザー登録用紙に必要事項をご記入の上、エクセルソフトまでご返送ください。
また、オンラインでもユーザー登録ができます。
https://www.xlsoft.com/jp/products/register/

エクセルソフト株式会社
〒108-0073 東京都港区三田 3 丁目 9 番 9 号 森伝ビル 6 階
Tel: 03-5440-7875 Fax: 03-5440-7876
E-MAIL: xlsoftkk@xlsoft.com
Web ページ: https://www.xlsoft.com/