Smart Technology. Delivered.™

Software Integration Guide

Wireless Software for Windows Embedded

Windows Mobile and Windows Embedded Handheld
Window CE and Windows Embedded Compact

Version 2.0

REVISION HISTORY

Version	Date	Notes	Approver
1.0	22 Jul 2014	Initial Release	Doug Moore
1.1	14 Aug 2014	Added Staging LAIRD Flat Release Directory (FRD)	Doug Moore
1.2	30 Sept 2014	SD30 now available for WEC7 platforms, updated WEC7 content and pictures SD45 builds now always Debug Output Capable, updated pictures	Doug Moore
1.3	27 Oct 2014	Added Driver Requirements	Doug Moore
1.4	05 Mar 2015	LCM (.NETCF) is now available in some distros, updated content and pictures Corrected some SYSGEN Variables information Added WEC2013 information	Doug Moore
1.5	11 Mar 2015	Document work around for ".NET CF Initialization Error" Updated SYSGEN Variables and Driver Requirements sections	Doug Moore
1.6	31 Mar 2015	Added Troubleshooting	Doug Moore
1.7	04 May 2015	Bluetooth Catalog Item Reorganization, updated pictures	Doug Moore
1.8	16 Oct 2015	Added Approved By column	Sue White
1.9	18 Dec 2015	Bluetooth Catalog Item addition, updated pictures Added Bluetooth Driver Requirements section	Doug Moore
2.0	27 Mar 2015	SYSGEN Variables updated 50 Series now available, added content, updated pictures. Updated to new template	Doug Moore

CONTENTS

INTRODUCTION

This guide describes how to integrate Laird wireless software for Windows Embedded operating systems. You must obtain the Laird Windows Embedded Software Suite from Laird support and configure the software files as outlined in this document.

Obtaining Laird Software

The Windows Embedded Software Suite is distributed in two ways:

- Cabinet (.cab) files
- Archive (.zip) files

Note: The new ZIP Distro includes the integration files, referred to as a Catalog Item Distro (CID), as well as the binaries for all supported target CPU architecture, for each supported target OS.

The software suite may be obtained from Laird support. Contact ews-support@lairdtech.com and be prepared to provide the following information:

- Your version of Windows Embedded. For example: WM65, CE5, CE6, WEC7, or WEC2013
- Your platform's CPU architecture. For example: ARMV4I, ARMV7, or x86
- Your radio module series.
  For example: 10 Series (specify CF or SD), 30 Series, 40 Series, 45 Series, 50 Series, or BT8xx Series

Catalog Item Distro (CID)

The Catalog Item Distro (CID) is a set of files provided in our ZIP Distros which, when used as prescribed in this document, provide a means for integrating our product in the Windows Embedded Platform Builder build environment.

Once in place, the CID provides a series of checkboxes in Platform Builder's Catalog Item view. From the Catalog Item View, you can select which radio module distribution you want to integrate with your OS design. This ultimately produces an OS image for your platform that supports your chosen radio module.

Note: For Windows Mobile (WM) and Windows Embedded Handheld (WEH) which have no Catalog Item view, a series of changes must be made to integrate our product. These changes are outlined in this document.

CID users can have the Laird product integrated with their OS design within minutes of installing the CID.

Once installed, the Laird CID allows you to quickly and easily create an OS design that produces an OS image that supports any of the Laird radio modules. Our selected products are integrated and built into the OS image.

CAB Distros

The cabinet (.cab) file or CAB provides a means for a quick trial run of our product on your device. When a CAB is used, the Laird product can be installed and uninstalled on your device, but the product will not be built into the OS image as is achieved by using our CID with your OS design. Some users use CAB files as their primary installation/integration method with no intention of having the Laird product built into the OS image they build for their device.

If the Microsoft CAB File Installer / Uninstaller is included in your OS design, you can manually copy a .cab file onto a device and execute it to install our product.

A CAB Distro can be used to update what is currently built into an OS image.

ZIP Distros (with CID)

Our ZIP Distro or ZIP provides the set of binaries that make up a release of our software suite for all supported target CPU architectures of each supported OS. It also includes the Laird CID content.

If you have already completed all of the integration and if everything is working correctly, you can use the Laird ZIP distros to quickly update the binaries at your current point of integration. Or, you can quickly update the binaries on your device by replacing the contents of your device's Windows directory with the contents of the ZIP file.

Below are shortcuts to the CID User Guide sections for each supported OS:

- Windows Mobile (WM) / Windows Embedded Handheld (WEH)
- Windows CE 5.0 (CE5)
- Windows Embedded CE 6.0 (CE6)
- Windows Embedded Compact 7 (WEC7)
- Windows Embedded Compact 2013 (WEC2013)

SYSGEN VARIABLES

Our products are **dependent** upon certain SYSGEN variables or OS components being included in your OS design:

SYSGEN Variable	Description	Comment
SYSGEN_ETH_80211	Wireless LAN (802.11) STA – Automatic Configuration and 802.1x (WZC)	Optional, applies to WM, WEH, CE5, CE6, and WEC7. This component adds Microsoft's Wireless Zero Config (WZC) to the OS design. Including Automatic Configuration provides the option of configuring our connection manager application to use a ThirdPartyConfig wireless supplicant (Microsoft's wireless supplicant) rather than the default Laird wireless supplicant.
SYSGEN_ETH_80211_NWIFI	Automatic Configuration and 802.1x (Native Wi-Fi)	Optional, applies to WEC2013. This component adds Microsoft's Native Wi-Fi to the OS design. Including Automatic Configuration provides the option of configuring our connection manager application to use a ThirdPartyConfig wireless supplicant (Microsoft's wireless supplicant) rather than the default Laird wireless supplicant.

SYSGEN Variable	Description	Comment
SYSGEN_NETWORKING	Networking Support	Applies to WEC2013. Combines SYSGEN_ETHERNET, SYSGEN_WINSOCK, SYSGEN_NDIS, SYSGEN_IPHLPAPI, and SYSGEN_TCPIP.
SYSGEN_ETHERNET	Wired Local Area Network (802.3, 802.5)	Applies to WM, WEH, CE5, CE6, and WEC7. (Included in SYSGEN_NETWORKING for WEC2013.)
SYSGEN_WINSOCK	Winsock Support	Applies to WM, WEH, CE5, CE6, and WEC7. (Included in SYSGEN_NETWORKING for WEC2013.)
SYSGEN_NDIS	Network Driver Architecture (NDIS)	Applies to WM, WEH, CE5, CE6, and WEC7. (Included in SYSGEN_NETWORKING for WEC2013.)
SYSGEN_IPHLPAPI	TCP/IP IP Helper API	Applies to WM, WEH, CE5, CE6, and WEC7. (Included in SYSGEN_NETWORKING for WEC2013.)
SYSGEN_TAPI	Telephony API (TAPI 2.0)	All operating systems. (For Bluetooth radio.)
SYSGEN_TCPIP	TCP/IP (TCP/IPv6 is supported by our products but is optional.)	Applies to WM, WEH, CE5, CE6, and WEC7. (Included in SYSGEN_NETWORKING for WEC2013.)
SYSGEN_NDISUIO	NDIS User-mode I/O Protocol Driver	All operating systems.
SYSGEN_CERTS	Certificates (CryptoAPI 2.0)	All operating systems. (For our supplicant.)
SYSGEN_SERVICES	Core Server Support	All operating systems. (For our tray icon, supplicant, and Bluetooth service.)
SYSGEN_DOTNETV35 **SYSGEN_DOTNETV35_SR**	.NET Compact Framework 3.5	Applies to CE5, CE6, and WEC7. Only required if .NET CF application is selected from catalog items for inclusion in the OS design, Laird Connection Manager (LCM) (.NETCF), for example.
SYSGEN_NETCF_WINFORMS **SYSGEN_NETCF_SR**	.NET Compact Framework 3.9	Applies to WEC2013. Only required if .NET CF application is selected from catalog items for inclusion in the OS design, Laird Connection Manager (LCM) (.NETCF), for example. *LCM (.NETCF) is the only CM option available for WEC2013.*

WLAN Driver Requirements

If you are using an SDIO radio module you must include a SD Bus driver and a SD Host Controller (SDHC) driver, **SYSGEN_SDBUS** and **SYSGEN_SDHC_STANDARD,** or similar, platform specific, SDBUS and SDHC drivers supplied by platform's BSP.

Note: Our SD radio module drivers work with all revisions/QFE levels of Microsoft's SD Bus drivers, 1.0 or 2.0.

When a 40 Series SDIO radio module fails to operate on a given platform it may be due to the fact that the platform's SD Host Controller driver is not configured for, or does not support, multi-block read and write operations. The 40 Series radio driver will indicate this failure with the following debug output:

```
loading firmware: ERROR: verifying    270 KB
loading firmware: verify sdio host controller supports multi-block operations
```

The solution to this issue is to insure that your platform's SD Host Controller (SDHC) driver does not set **SD_SLOT_USE_SOFT_BLOCK_CMD53_READ** nor **SD_SLOT_USE_SOFT_BLOCK_CMD53_WRITE** when calling **SDHCDSetSlotCapabilities** and that the driver does, in fact, support hardware multi-block reads and writes.

If you are using a CF (Compact Flash) radio module you must include a PC Card Host [Controller] (PCMCIA) driver supplied by the platform's BSP.

Note: PCMCIA support was dropped by Microsoft starting with Windows Embedded Compact 7 (WEC7). Our CF radio modules cannot be used on WEC7 and higher operating systems without complete support from the platform's BSP and its drivers.

Bluetooth Driver Requirements

Our catalog items let you specify the Bluetooth Configuration parameters for the following: *Bluetooth Stack*, *BTUART – COM Port*, *BTUART – Bit Rate*, *Protocol*, and *Radio Module*. Using the CID's catalog items automates the integration, however, if you choose not to use our CID and its catalog items the platform specific settings must be done by hand.

If you are using a CSR Bluetooth radio module (i.e. BT8xx or 50NBT), and an older distribution of our software, or not using our CID's catalog items for integration, you may need to manually configure the following settings in the distro's _BT.reg file and the distro's Bluetooth patch file (e.g. the PSR file).

The following settings, in each file, must match.

- **@ Catalog Items View @ *Laird / [Bluetooth Configuration] / [BTUART – Bit Rate]***
  You have specified the COM port on your platform that is responsible for being the Bluetooth UART, or BTUART for short. This is your platform's UART interface that is tied to our Bluetooth radio's UART interface.
  The operational baud rate is not presented to the user in our catalog items, so you must set your platform's operational baud rate for your platform's BTUART by manually editing the integration files.
  Uncomment the line with the desired operation baud rate, or define a macro that is to be used to specify the desired operational baud rate. Note that the baud rates must match.

Note: Newer distributions of our software have catalog items for bit rate and this step is no longer required if the catalog items are used as the integration method.

Catalog Item	_BT.reg file (define _BAUD_ macro) Applies to: BT8xx, 50NBT, or 40NBT	PSR file (uncomment line with desired operation baud rate) Applies to: BT8xx and 50NBT
[BTUART]	`#define _BAUD_ $(BAUD_115200)`	`&01ea = 0001 c200            // 115200`

- **@ Catalog Items View @ Laird / [Bluetooth Configuration] / [Protocol]**

Catalog Item	_BT.reg file (automatic) Applies to: BT8xx, 50NBT, or 40NBT	PSR file (uncomment following line) Applies to: BT8xx and 50NBT
BCSP protocol	`"Protocol"=dword:2 ; UART link running BCSP`	`&01f9 = 0001 // UART link running BCSP (if MSBT bthcsr.dll)`
H4 protocol	`"Protocol"=dword:1 ; UART link running H4`	`&01f9 = 0003 // UART link running H4 (if MSBT bthuart.dll)`

Note: Selecting 40 Series under Bluetooth Configuration Radio Module automatically selects H4 protocol.
The 40 series Bluetooth radio is only capable of operating using the H4 protocol; no edits are required.

- **Bluetooth MAC Address (BDADDR)** – Applies to: BT8xx only
  Reference **PSKEY_BDADDR** section in Distro's PSR file where everything is explained.

TROUBLESHOOTING

- Our .NET CF based applications may fail to launch, with the system reporting a, ".NET CF Initialization Error", if the following registry entry exists in the registry on the device:

```
[HKEY LOCAL MACHINE\SOFTWARE\Microsoft\.NETCompactFramework\Managed Debugger]
"AttachEnabled"=dword:1
```

- When building applications with Visual Studio 2005, that link in our sdk.lib, you may see the following error message and the build may fail:

```
Error Message : sdk.lib(confg strings.obj) : fatal error LNK1103: debugging
information corrupt; recompile module
```

  The solution to this issue is to apply http://support.microsoft.com/en-us/kb/949009 to your VS2005 build environment (since our sdk.lib binaries are built using VS2008).

- When a 40 Series SDIO radio module fails to operate on a given platform it may be due to the fact that the platform's SD Host Controller driver is not configured for, or does not support, multi-block read and write operations. The 40 Series radio driver will indicate this failure with the following debug output:

```
loading firmware: ERROR:      verifying   270 KB
loading firmware: verify sdio host controller supports multi-block operations
```

The solution to this issue is to insure that your platform's SD Host Controller (SDHC) driver does not set **`SD_SLOT_USE_SOFT_BLOCK_CMD53_READ`** nor **`SD_SLOT_USE_SOFT_BLOCK_CMD53_WRITE`** when calling **`SDHCDSetSlotCapabilities`** and that the driver does, in fact, support hardware multi-block reads and writes.

Staging Laird Flat Release Directory (FRD)

Some customers may have already completed their own integration and may not want to use the recommended method of integration that is outlined in this document. If all you need is a *flat release directory* for this product, the steps to obtain this are outlined in this section.

To conserve space, our ZIP distros are designed so there are no duplicate files in the distro.

The intent is that you use the catalog item method of integration, in which case the distro's postlink.bat is executed when the OS design's subprojects are built, and the components for the selected Laird catalog items are integrated with your OS design.

For customers who only want to update their integration, the **_stage.bat** is included in every ZIP distro.

When run, **_stage.bat** creates a Laird FRD (Flat Release Directory) for each target CPU of each supported OS found in the distro. This is located in a folder named **_stage**.

With the appropriate target OS and target CPU, Laird FRD can then be used to merge with existing integration, updating that integration to what is in the distro.

To stage the distro's content, follow these steps:

1. Open command prompt @ distro's staging point.

   For example:

   `\WINCE700\3rdParty\Laird\SD45_23_3_3_5`

   a. Execute the following command (to stage the distro's content): **`_stage`**

      You should see output similar to the following as the disto is staged:

```
    C:\WINCE700\3rdParty\Laird\SD45 23 3 3 5> stage

    WINCE500 ARM
    WINCE500 x86
    WINCE600 ARM
    WINCE600 x86
    WINCE700 ARM
    WINCE700 x86
    WINCE800 ARM
    WINCE800 x86
    WM500 ARM
    WM6530 ARM

    C:\WINCE700\3rdParty\Laird\SD45 23 3 3 5>
```

2. Copy, or use to merge, the appropriate target OS and target CPU Laird FRD found in the _stage folder with the existing integration, updating that integration to what is in the distro.

Note: The distro's BIB, REG, and DAT files depend upon certain environment variables (such as **LAIRD_CM_LCM**) that are normally set automatically when the catalog item method of integration is used. With this method of integration, you must ensure that these environmental variables are set for the build process and integration to complete successfully.

The distro's BIB, REG, and DAT files are hard-coded to use the path **$(_FLATRELEASEDIR)\LAIRD\**.

If your method of integration does not result in the appropriate target OS and target CPU LAIRD FRD ending up at this location, the LAIRD FRD's BIB, REG, and DAT files must be altered to reflect the correct path.

Windows Mobile (WM)/Windows Embedded Handheld (WEH)

To load the Windows Mobile (WM) or Windows Embedded Handheld (WEH) CID, follow these steps:

1. Open a build window for your WM platform's OS design by doing the following:
   a. Open command prompt (run as Administrator)
   b. Change directory to current OS design:
   ```
   cd %_WINCEROOT%\BuildScripts\<OSDesign>
   ```
   c. Execute the following command (to set up OS design's build environment):
   ```
   baseosenv
   ```
2. Inspect the Build Environment by doing the following:
   a. In your WM platform's OS design build window, enter the following command to list the build environment variables that begin with the underscore character:
   ```
   set _|sort
   ```
   Inspecting these build environment variables helps you to know what to do in the steps to follow. The following are the variables of interest:
   ```
     FLATRELEASEDIR=C:\WM6530\release\WPC M3SKYPLUS Retail\WPC
     TARGETPLATROOT=C:\WM6530\platform\M3SKYPLUS
     TGTCPU=ARMV4I
     TGTCPUFAMILY=ARM
     TGTPLAT=M3SKYPLUS
     WINCEOSVER=502
    _WINCEROOT=C:\WM6530
   ```
3. Extract the contents of the Laird distro's compressed file and align the content with the following path (this creates a folder named *Laird)* :
   ```
   %_WINCEROOT%\3rdParty
   ```

4. Set up Environment variables.

   By inspecting the distro's PBCXML files which are part of our CID and are found in the distro's CATALOG folder, you can see which entries are required to set up the environment variables that our CID's BIB, DAT, and REG files need.

   Note: Some variables can be ignored. For example, **LAIRD, LAIRD_BT, LAIRD_BT_STACK, LAIRD_BT_BTUART, LAIRD_BT_PROTOCOL, LAIRD_BT_RADIO, LAIRD_CM, LAIRD_RADIO** variables are used by the PBCXML files to set the state of the checkboxes in Platform Builder's Catalog Item view and are not needed to meet our objective.

   For example, in the distro's PBCXML files you see **<Variable>** entries like these:

   ```
   <Variable>SD45_23_3_3_5</Variable>
   <Variable>LAIRD_WLAN_45_SERIES</Variable>
   <Variable>LAIRD_WLAN_45_SERIES_BT_SUPPORT</Variable>
   <Variable>LAIRD_BTSTACK</Variable>
   <Variable>LAIRD_BTUART_COM2</Variable>
   <Variable>LAIRD_BT_PROTOCOL_H4</Variable>
   <Variable>LAIRD_CM_LCM</Variable>
   ```

   These entries translate into the following entries being made by hand in your OS design's .bat file after the section ***REM Variables required by Optional Components***: This .bat file is found in the following location:

   `%_WINCEROOT%\BuildScripts\<OSDesign>\BaseOSEnv.bat`

   ```
   set LAIRD_DISTRO=SD45_23_3_3_5
   set LAIRD_WLAN_45_SERIES=1
   set LAIRD_WLAN_45_SERIES_BT_SUPPORT=1
   set LAIRD_BTSTACK=1
   set LAIRD_BTUART_COM2=1
   set LAIRD_BT_PROTOCOL_H4=1
   set LAIRD_CM_LCM=1
   ```

   Inspect our CID's BIB, DAT, and REG files, found in the distro's *Files* folder, as confirmation.

5. Edit your %_TARGETPLATROOT%\FILES\platform.bib file by adding:

   ```
   #include "$(_FLATRELEASEDIR)\LAIRD\_SD45.bib"
   ```

6. Edit your %_TARGETPLATROOT%\FILES\platform.reg file by adding:

   ```
   #include "$(_FLATRELEASEDIR)\LAIRD\_SD45.reg"
   ```

7. Edit your **`%_TARGETPLATROOT%\FILES\platform.dat`** file by adding the content of the distro's **`FILES\*.dat`** file(s).

Note: The #include method does not work with DAT files.

8. Build Distro. In your WM platform's OS design build window type the following commands:
   a. `cd %_WINCEROOT%\3rdParty\Laird\%LAIRD_DISTRO%`
   b. `build`

      Note: The chosen distro's sources project is built and its *postlink.bat* file stages everything your OS design needs `@ $(FLATRELEASEDIR)\LAIRD`

   c. `dir %_FLATRELEASEDIR%\laird` (to verify)
9. Build the BSP/OS image.
10. Load the image on the device and run (device specific).
11. You can customize the *Parameter* files as needed, in the following directory:

    ```
    %_WINCEROOT%\3rdParty\Laird\<chosen distro>\FILES
    ```
12. If customized, repeat steps 8 thru 10.

WINDOWS CE 5.0 (CE5)

To load the Windows CE 5.0 (CE5) Catalog Item Distribution (CID), follow these steps:

1. Extract the contents of the compressed file and align the content with the following path (this creates a folder named *Laird)* :

   `%_WINCEROOT%\3rdParty`

2. Open your OS design in Platform Builder.
3. Add Laird Catalog Items to the Platform Builder Catalog by doing the following:

   a. On Platform Builder menu, click **File > Manage Catalog Items...**

   b. Click **Import...** and navigate to the following directory:

   `%_WINCEROOT%\3rdParty\Laird\CATALOG`

   c. If not already imported, select **_laird.cec** and click **Open**.

   d. Repeat these steps to import each desired radio's build distro's CEC file.

4. Extract the contents of the compressed file and align the content with the following path (this creates a folder named *Laird)* :

   `%_WINCEROOT%\3rdParty`

5. Open your OS design in Platform Builder.
6. Add Laird Catalog Items to the Platform Builder Catalog by doing the following:

   a. On Platform Builder menu, click **File > Manage Catalog Items...**

   b. Click **Import...** and navigate to the following directory:

   `%_WINCEROOT%\3rdParty\Laird\CATALOG`

   c. If not already imported, select **_laird.cec** and click **Open**.

   d. Repeat these steps to import each desired radio's build distro's CEC file.

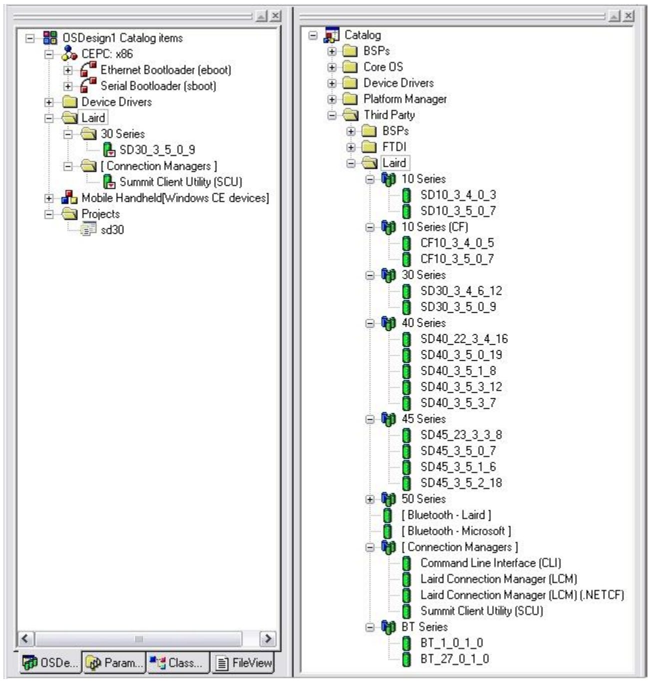
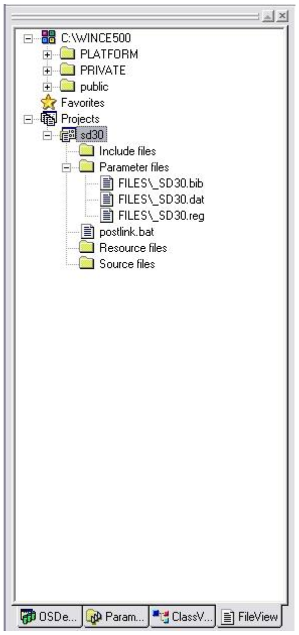


Figure 1: OS design in Platform Builder



7. In Platform Builder, right-click **Catalog Items View**, click **Refresh Catalog**, and expand the *Laird* folder.
8. In Catalog View, right-click the desired components and options and click **Add to OS design**.
    a. Hold the cursor over items to see more information about an item.
    b. In the OS design View, the items are added under *Laird* and *Projects*.
    c. In File View, the items are added under *Projects*.
9. If the OS image was previously built, select the following on the Platform Builder menu:
    a. **Make Run-Time Image After Build** (if it isn't already checked).
    b. **Build Projects**, **Build All Projects**.
10. Load the image on your device and run.
11. You can customize the *Parameter* files under each subproject in the Solution Explorer as needed.

    If customized, repeat steps 6 and 7.

WINDOWS EMBEDDED CE 6.0 (CE6)

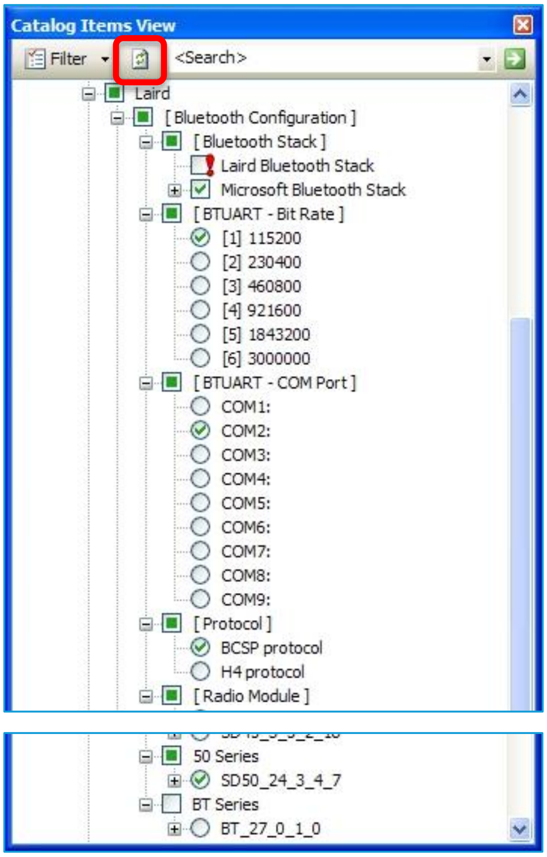
To load the Windows Embedded CE 6.0 (CE6) CID, follow these steps:

1. Extract the contents of the compressed file and align the content with the following path (this creates a folder named *Laird)* :

   ```
   %_WINCEROOT%\3rdParty
   ```

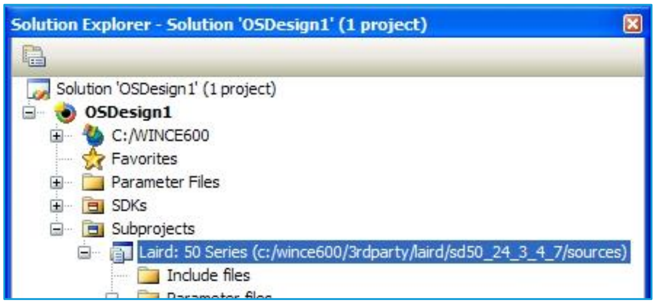
2. Open your OS design in Platform Builder.
3. Go to Catalog Items View and click the refresh button (**Error! Reference source not found.**).
4. Navigate to and expand the *Laird* folder to locate the *Third Party* folder.
5. In Catalog Item view, select the desired components and options (add checkmarks and fill in options).
   a. Hold the cursor over items to see more information.
   b. Square brackets **[ ]** are used as status indicators; they allow you to see which items are configured or have at least one item selected; they are not intended to be 'checked off'.
6. Use the Solution Explorer to display the subprojects related to what is added to the OS design. (Figure 3).
7. If the OS image was previously built, on the Platform Builder click **Build > Build All Subprojects** or do a targeted build by right-clicking one of the subprojects and clicking **Build**.
8. Do a **Build**, **Make Run-Time Image** if it does not automatically happen after you initiate the targeted build.
9. Load the image on your device and run it.
10. If necessary, you can customize the parameter files under each subproject in the Solution Explorer.

    If customized, repeat steps 6 through 8.



- ■ Laird
  - ■ [Bluetooth Configuration]
    - ■ [Bluetooth Stack]
      - □ Laird Bluetooth Stack
      - ☑ Microsoft Bluetooth Stack
    - ■ [BTUART - Bit Rate]
      - ✓ [1] 115200
      - ○ [2] 230400
      - ○ [3] 460800
      - ○ [4] 921600
      - ○ [5] 1843200
      - ○ [6] 3000000
    - ■ [BTUART - COM Port]
      - ○ COM1:
      - ✓ COM2:
      - ○ COM3:
      - ○ COM4:
      - ○ COM5:
      - ○ COM6:
      - ○ COM7:
      - ○ COM8:
      - ○ COM9:
    - ■ [Protocol]
      - ✓ BCSP protocol
      - ○ H4 protocol
    - ■ [Radio Module]
      - ■ 50 Series
        - ✓ SD50_24_3_4_7
      - □ BT Series
        - ○ BT_27_0_1_0

Figure 2: Catalog Items View



- Solution 'OSDesign1' (1 project)
  - **OSDesign1**
    - C:/WINCE600
    - Favorites
    - Parameter Files
    - SDKs
    - Subprojects
      - Laird: 50 Series (c:/wince600/3rdparty/laird/sd50_24_3_4_7/sources)
        - Include files
        - Parameter files

Figure 3: Solution Explorer

Note: A **red-X** in a selected catalog item's checkbox indicates that the selected distro, under *[Radio Modules]*, does not include the binaries for the X-ed out component.

Windows Embedded Compact 7 (WEC7)

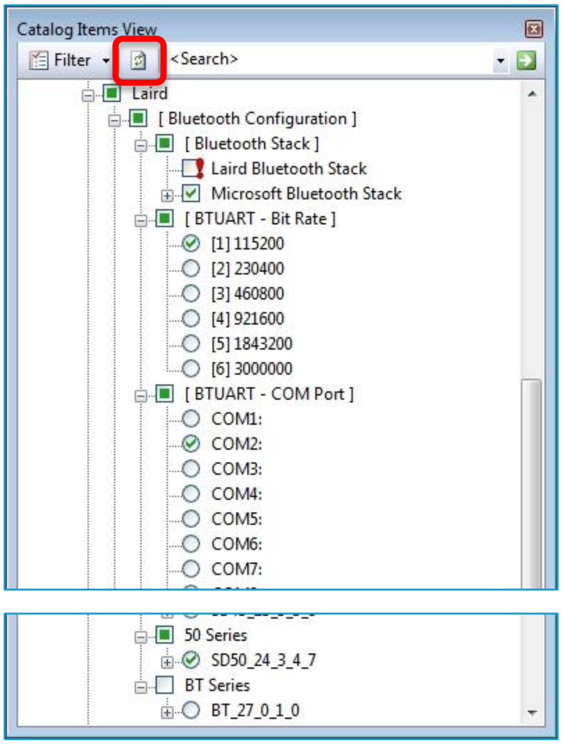
To load the Windows Embedded Compact 7 (WEC7) CID, follow these steps:

1. Extract the contents of the compressed file and align the content with the following path (this creates a folder named *Laird)* :

   `%_WINCEROOT%\3rdParty`
2. Open your OS design in Platform Builder.
3. Go to Catalog Items view and click the refresh button (Figure 4).
4. Navigate to and expand the *Laird* folder to locate the *Third Party* folder.

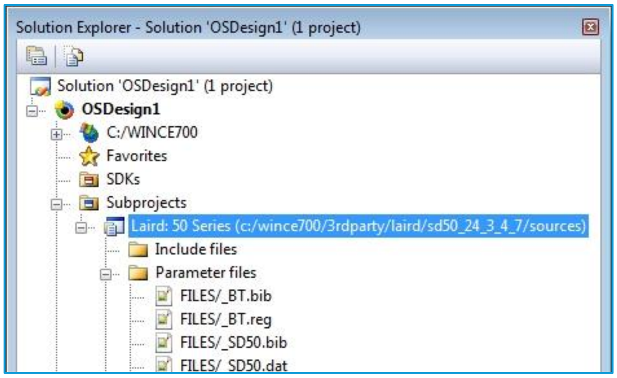
   Note: The 10 Series radios are not supported on the WEC7 operating system. They are dimmed / inactive in the Catalog Item View and therefore cannot be selected.
5. In Catalog Item view, select the desired components and options (add checkmarks and fill in options).
   a. Hold the cursor over an item for additional information.
   b. Square brackets **[ ]** are used as status indicators; they allow you to see which items are configured or have at least one item selected; they are not intended to be 'checked off'.
6. Use the Solution Explorer to display the subprojects related to what is added to the OS design.
7. On the Platform Builder menu click **Build > Build All Subprojects** or do a targeted build by right-clicking one of the subprojects and clicking **Build**.
8. Do a **Build**, **Make Run-Time Image** if it does not automatically happen after you initiate the targeted build.
9. Load the image on your device and run it.
10. If necessary, you can customize the parameter files under each subproject in the Solution Explorer.

    If customized, repeat steps 7 through 9.



```
Laird
  [ Bluetooth Configuration ]
    [ Bluetooth Stack ]
      Laird Bluetooth Stack
      ☑ Microsoft Bluetooth Stack
    [ BTUART - Bit Rate ]
      ✓ [1] 115200
      ○ [2] 230400
      ○ [3] 460800
      ○ [4] 921600
      ○ [5] 1843200
      ○ [6] 3000000
    [ BTUART - COM Port ]
      ○ COM1:
      ✓ COM2:
      ○ COM3:
      ○ COM4:
      ○ COM5:
      ○ COM6:
      ○ COM7:
    50 Series
      ✓ SD50_24_3_4_7
    □ BT Series
      ○ BT_27_0_1_0
```

Figure 4: Custom Items View



```
Solution 'OSDesign1' (1 project)
  OSDesign1
    C:/WINCE700
    Favorites
    SDKs
    Subprojects
      Laird: 50 Series (c:/wince700/3rdparty/laird/sd50_24_3_4_7/sources)
        Include files
        Parameter files
          FILES/_BT.bib
          FILES/_BT.reg
          FILES/_SD50.bib
          FILES/_SD50.dat
```

Figure 5: Solution Explorer

Note: A **red-X** in a selected catalog item's checkbox indicates that the selected distro, under *[Radio Modules]*, does not include the binaries for the X-ed out component.

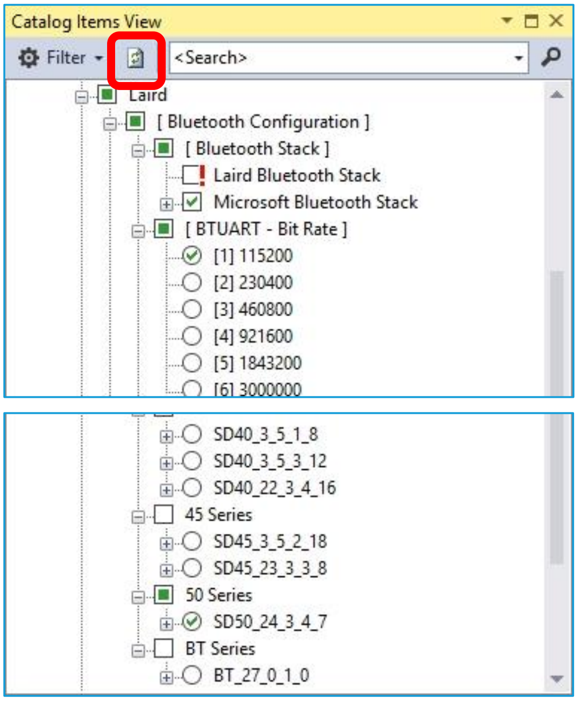
WINDOWS EMBEDDED COMPACT 2013 (WEC2013)

To load the Windows Embedded Compact 2013 (WEC2013) CID, follow these steps:

1. Extract the contents of the compressed file and align the content with the following path (this creates a folder named Laird) :

   `%_WINCEROOT%\3rdParty`
2. Open your OS design in Platform Builder.
3. Go to Catalog Items view and click the refresh button (Figure 6).
4. Navigate to and expand the *Laird* folder to locate the *Third Party* folder.

   Note: The 10 Series and 30 Series radios are not supported on the WEC2013 operating system. They are dimmed / inactive in the Catalog Item View and therefore cannot be selected.
5. In Catalog Item view, select the desired components and options (add checkmarks and fill in options).
   a. Hold the cursor over an item for additional information.
   b. Square brackets **[ ]** are used as status indicators; they allow you to see which items are configured or have at least one item selected; they are not intended to be 'checked off'.
6. Use the Solution Explorer to display the subprojects related to what is added to the OS design.
7. On the Platform Builder menu click **Build > Build All Subprojects** or do a targeted build by right-clicking one of the subprojects and clicking **Build**.
8. Do a **Build**, **Make Run-Time Image** if it does not automatically happen after you initiate the targeted build.
9. Load the image on your device and run it.
10. If necessary, you can customize the parameter files under each subproject in the Solution Explorer.
11. If customized, repeat steps 7 through 9.



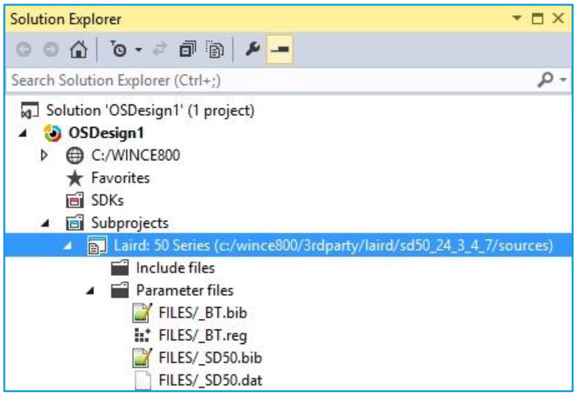
Catalog Items View

Filter <Search>

```
Laird
  [ Bluetooth Configuration ]
    [ Bluetooth Stack ]
      Laird Bluetooth Stack
      Microsoft Bluetooth Stack
    [ BTUART - Bit Rate ]
      [1] 115200
      [2] 230400
      [3] 460800
      [4] 921600
      [5] 1843200
      [6] 3000000
```

```
      SD40_3_5_1_8
      SD40_3_5_3_12
      SD40_22_3_4_16
    45 Series
      SD45_3_5_2_18
      SD45_23_3_3_8
    50 Series
      SD50_24_3_4_7
    BT Series
      BT_27_0_1_0
```

Figure 6: Custom Items View



Solution Explorer

Search Solution Explorer (Ctrl+;)

```
Solution 'OSDesign1' (1 project)
  OSDesign1
    C:/WINCE800
    Favorites
    SDKs
    Subprojects
      Laird: 50 Series (c:/wince800/3rdparty/laird/sd50_24_3_4_7/sources)
        Include files
        Parameter files
          FILES/_BT.bib
          FILES/_BT.reg
          FILES/_SD50.bib
          FILES/_SD50.dat
```

Figure 7: Solution Explorer

Note: A **red-X** in a selected catalog item's checkbox indicates that the selected distro, under *[Radio Modules]*, does not include the binaries for the X-ed out component.