Le71SDKAPIL VP-API-II Software P2.24.0 CSLAC, ZL880 and miSLIC™ Series

| Document ID# 140202 | Version 11 | Nov 13, 2014 |

This document describes changes and known errata from release P2.23.0 to release P2.24.0 of the Le71SDKAPIL software. Refer to the CSLAC VoicePath™ API-II Reference Guide, Version 20 and ZL880/miSLIC VoicePath™ API-II Reference Guide, Version 8 (install_dir/documents directory with this release) for VP-API-II P2.24.0 interface details.

1.0 REVISION SUMMARY

These are the release notes and errata from release P2.23.0 to release P2.24.0 of the VP-API-II. This document exists in the directory `install_dir/release_notes` for this installation.

1.1 Document Changes

The following documents have changed since the P2.23.0 release:

Table 1–1 P2.24.0 Document Changes

P2.23.0	P2.24.0
ZL880 VP-API-II Reference Guide, Rev 7	ZL880/miSLIC VP-API-II Reference Guide, Rev 8
CSLAC VP-API-II Reference Guide, Rev 20	N/A

For a detailed list of document changes see the Appendix "Revision History" in the corresponding VP-API-II Reference Guide.

1.2 New Silicon/Configuration Support

As documented in the ZL880/miSLIC VP-API-II Reference Guide, Version 8, table 1-3 the VP-API-II P2.24.0 release supports the following new silicon and supply configurations:

Table 1–2 P2.24.0 New Supported Silicon/Configurations

Silicon	Supply Mode	Supply Configuration	Required Software
Le9652	Tracker	FB, IB	VP-API-II P2.24.0 or later
Le9642	ABS	FB, BB	Profile Wizard P2.7.0 or later
Le9651	Tracker	FB, IB, BB	(If using Line Test SW)
Le9641	Tracker	FB, IB, BB	ZL880SLVVP P1.3.0 or later

Table 1–3 is a complete list of silicon supported in this release.

Table 1–3 Supported Silicon in P2.24.0

miSLIC	ZL880	VE880	VE890	VE790	VE580
Le9672*	ZL88601/2	Le88010	Le89010	Le79228	Le58QL02/21
Le9662*	ZL88701/2	Le88111/116	Le89116/136	Le79Q224x	Le58QL03/31
Le9661*	ZL88801	Le88131/136	Le89156		Le58QL061/63
Le9652* (new)		Le88211/216	Le89316/336		Le58QL022
Le9642* (new)		Le88231/236/276			Le58QL032
Le9651 (new)		Le88506 (VE8820 Chipset)			Le58083
Le9641 (new)		Le88536* (V8830 Chipset)			Le58DL021
		Le88221/241/246			
		Le88264*			
		Le88266/286			
		Le88311/331			

Note: * ZSI only host interface

1.3   Line Test API Compatibility

VP-API-II release P2.24.0 is compatible with the following Microsemi VeriVoice (Line Test API) releases:

Table 1–4 Compatible VeriVoice releases

miSLIC	ZL880	VE880		VE890	
ZL880SLVVP	ZL880SLVVP	VE880SLVVP	VE880SLVV	VE890SLVVP	VE890SLVV
P1.3.0	P1.3.0	P1.8.0	P1.4.0	P1.4.0	P1.2.0
P1.2.0	P1.2.0	P1.7.2	P1.3.1	P1.3.2	P1.1.2
	P1.1.0				

For support information on combinations not previously listed, contact your local Microsemi Sales or Technical Support Representative.

Note: The VP-API-II DTMF detection feature MUST be disabled before starting a ZL880/miSLIC VeriVoice test prior to version P1.3.0.

1.4   Upgrading to P2.24.0

The content of this release replaces ALL content of previous releases. Users should follow the instructions below when converting from any previous revision to P2.24.0:

- All files that existed in the previous release but do not exist in this release must be removed from the build path and Makefile listing.
- Files that are added in this release must be added to the Makefile (build system).

- Files of the same name (except `vp_api_cfg.h`) in this release MUST be used to replace files from the previous release.
- Updating/Replacing `vp_api_cfg.h`:
  - Values that exist ONLY in the new file MUST be used.
  - Values that exist ONLY in the previous file MUST be removed.
  - Values that exist in both files should use the setting from the previous file.

See Section 5.0 for a list of the supplied files and complete directory structure of P2.24.0.

2.0   OPERATIONAL NOTES

N1   **Operational Note:** All ZL880, miSLIC, VE880 and VE890 FXS Applications must call `VpCalLine()` or apply calibration coefficients to the system using `VpCal()`. This is required to meet Data Sheet Specifications.

N2   **Operational Note:** The ZL88801 Shared Tracking Supply design cannot be used in a Ground Start OR Ground Key Application.

N3   **Operational Note:** When using any ZL880 or miSLIC ABS design, both channels of the device must be initialized and calibrated, even if one is not physically populated.

N4   **Operational Note:** When using a Shared Buckboost ABS design, line contexts (`VpLineCtxType`) for both channels must be initialized and linked to the same device context (`VpDevCtxType`) when calling any line-specific function (e.g., `VpSetLineState()`, `VpSetLineTone()`, etc..).

3.0   CORRECTED ERRATA FROM RELEASE P2.23.0

The items in this list were reported as errata in the P2.23.0 release notes or were discovered after the P2.23.0 release date.

3.1   Bug Fixes

ZL880 / miSLIC

C1   Corrected a VP-API-II `VpSetOption()` `VP_OPTION_ID_TIMESLOT` configuration issue in ZSI mode when VP_OPTION_ID _CODEC had previously been set to `VP_CODEC_LINEAR`, `VP_OPTION_LINEAR_WIDEBAND`, `VP_OPTION_ALAW_WIDEBAND, or VP_OPTION_MLAW_WIDEBAND`. Attempting to set the timeslot to 0 under the aforementioned configuration would cause VpSetOption() to incorrectly return a status of `VP_STATUS_INPUT_PARAM_OOR`.

C2   Corrected `VpGetLoopCond()` to properly measure/return voltage values when run with the line in the `VP_LINE_DISCONNECT` line state.

C3   Modified the VP-API-II ring trip debounce time to a value of 100ms to be consistent with the VE880 and VE890 CSLAC behavior.

C4   Corrected a calibration coefficient corruption issue when a re-calibration sequence is performed by the customers application. A re-calibration sequence consists of a call to `VpCalLine()` followed by a call to `VpCal()` in order to `NULL` out a lines calibration coefficients, followed by an additional call to `VpCalLine()`.

C5   Corrected a calibration coefficient corruption issue when a call to `VpInitLine()` or `VpConfigLine()` is made prior to a call to `VpCalLine()`.

VE880

C6   Corrected a rare VE880 VpCalLine() corruption error when VpCalLine() is running on both channels of a device simultaneously.

3.2   New Features and Changes In P2.24.0

ZL880 and miSLIC

C7   Utilized new hardware features of the Le9652, Le9642, Le9651 and Le9641 devices to improve DTMF detection algorithm for these devices in the Analog to Digital path by attenuating the 4 wire echo.

C8   Added option `VP_OPTION_ID_DTMF_PARAMS`. This option provides run time configuration of DTMF detection thresholds. See section 3.3.23 of the VoicePath API-II Reference Guide for details.

C9   Added "average detection level" to the `eventData` parameter of the `VP_LINE_EVID_DTMF` event. This value is an indication of the level of the DTMF digit detected by the VP-API-II. See section 8.4.10 of the VoicePath API-II Reference Guide for details.

C10   Added option `VP_DEVICE_OPTION_ID_RING_PHASE_SYNC`. This option provides the ability to offset the ringing phase, of the two different lines of a single device, by 90 degrees to reduce peak current power draw of the line circuit supply. See section 3.3.25 of the VoicePath API-II Reference Guide for details.

C11   Added option `VP_OPTION_ID_RINGTRIP_CONFIRM`. This option provides a runtime configurable ring trip mask duration. See section 3.3.26 of the VoicePath API-II Reference Guide for details.

C12   Added option `VP_DEVICE_OPTION_ID_FSYNC_RATE`. This option provides a runtime, device level selection of an 8 kHz or 16 kHz frame sync input signals. This option is only supported by Le9651 and Le9641 devices. See section 3.3.24 of the VoicePath API-II Reference Guide for details.

C13   Added `VpShutdownDevice()` interface function. This function will place all lines of a device into the `VP_LINE_DISABLED` line state, shut down all power supplies, mask all interrupts and cancel all running VP-API-II timers. See section 9.2.15 of the VoicePath API-II Reference Guide for details.

C14   Added `VP_LINE_DISABLED` to the `VpLineStateType` enumeration. This enumeration is only supported by the `VpSetLineState()` function for tracking supply types. This enumeration will place the line into the `VP_LINE_DISABLED` line state, shut down the channel's associated power supply, mask the channel's interrupts and cancel the channel's running VP-API-II timers. See section 9.2.2 of the VoicePath API-II Reference Guide for details

C15   Added `VP_ADAPT_RING_SINGLE_BB_TRACKER` to the `VpAdaptiveRingingModeType` enumeration. This enumeration is only supported by the `VP_DEVICE_OPTION_ID_ADAPTIVE_RINGING` option for single channel Buck Boost Tracking supply designs. See section 3.3.20 of the VoicePath API-II Reference Guide for details.

C16   Removed Tip/Ring transients when `VpInitDevice()` and `VpCalLine()` are issued.

C17   Masking device clock fault and timestamp rollover interrupts after calling `VpFreeRun()` `VP_FREE_RUN_START` to avoid waking up host controller in a sleep mode.
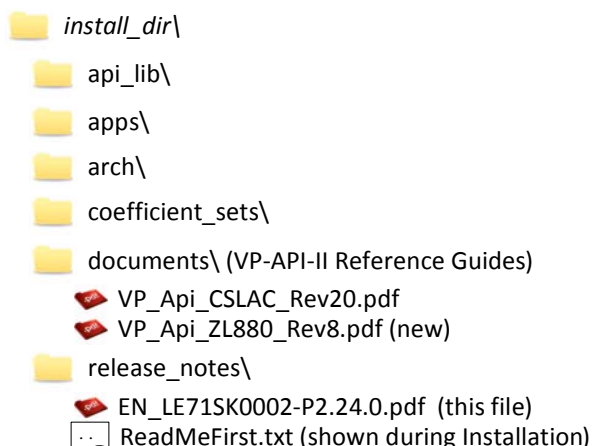
VE890

C18   Improved VE890 thermal fault detection to prevent spurious false `VP_LINE_EVID_THERM_FLT` events from being generated if the CREF capacitor in the customer's design is populated with a value less than 4.7 uF.

4.0   KNOWN ERRATA IN P2.24.0

None.

5.0   LE71SDKAPIL P2.24.0 DIRECTORY ORGANIZATION

Figure 5–1 Le71SDKAPIL P2.24.0 Directory Organization

- 📁 *install_dir*
  - 📁 api_lib\
  - 📁 apps\
  - 📁 arch\
  - 📁 coefficient_sets\
  - 📁 documents\ (VP-API-II Reference Guides)
    - 📕 VP_Api_CSLAC_Rev20.pdf
    - 📕 VP_Api_ZL880_Rev8.pdf (new)
  - 📁 release_notes\
    - 📕 EN_LE71SK0002-P2.24.0.pdf (this file)
    - 🗨 ReadMeFirst.txt (shown during Installation)

Note that the directories "\documents" and "\release_notes" only contain the documents shown in Figure 5–1. They are:

- *install_dir***\documents\VP_Api_CSLAC_Rev20.pdf**: This is the <u>CSLAC VP-API-II Reference Guide</u>. It should be used for VE880, VE890, VE580 and VE790 VP-API-II designs only.

- *install_dir***\documents\VP_Api_ZL880_Rev8.pdf**: This is the <u>ZL880/miSLIC VP-API-II Reference Guide</u>. It should be used for ZL880 and miSLIC designs only.

- *install_dir***\release_notes\EN_Le71SDKAPIL-P2.24.0.pdf**: This document.

- *install_dir***\release_notes\ReadMeFirst.txt**: High level release information shown during the VP-API-II installation.
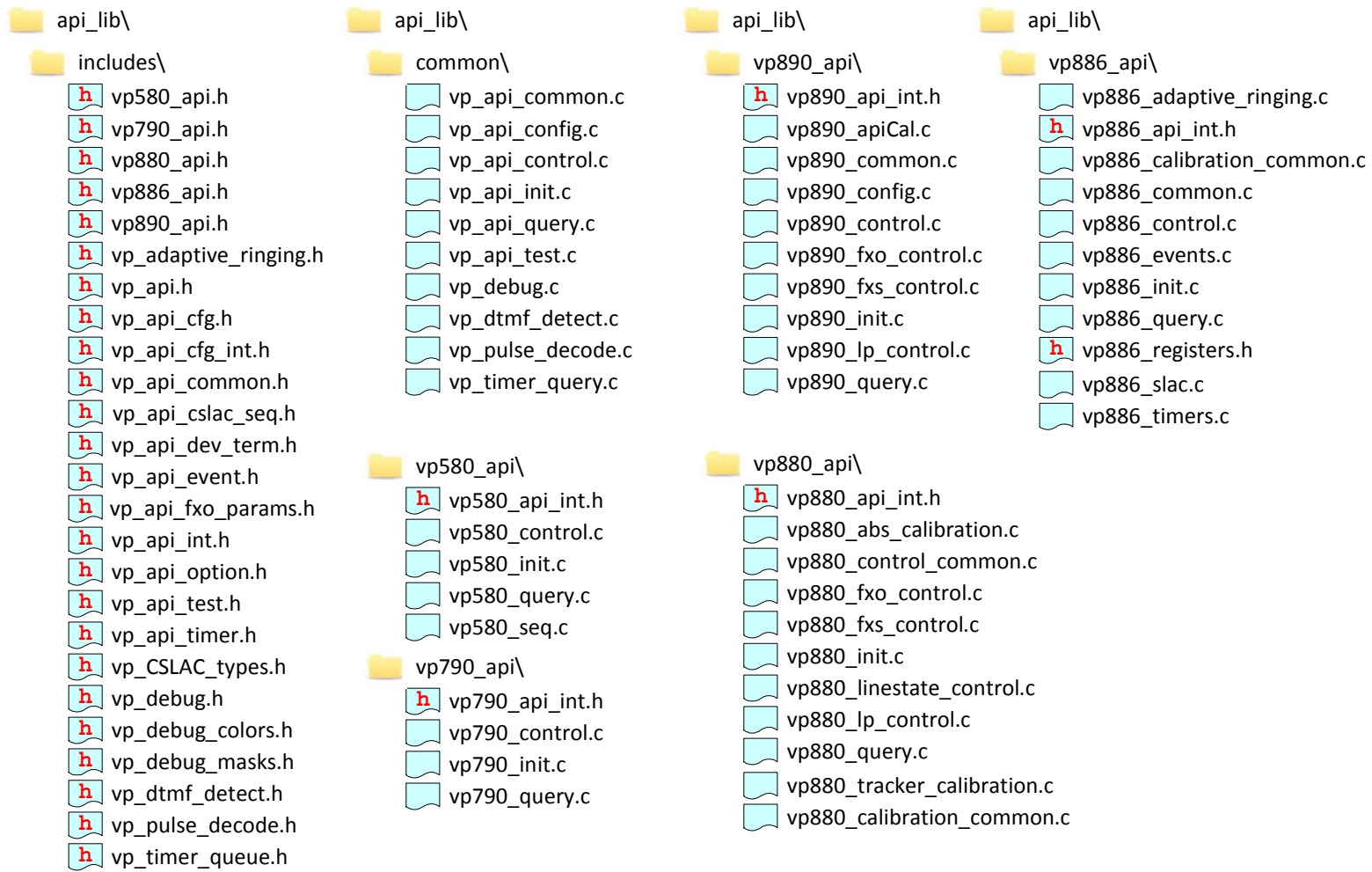
The remaining folders are discussed in more detail in the following sections:

- *install_dir*\api_lib\ (see "VP-API-II Core" on page 6)
- *install_dir*\apps\ (see "Sample Customer Applications" on page 7)
- *install_dir*\arch\ (see "System Service and Hardware Abstraction Layer (HAL) Examples" on page 7)
- *install_dir*\coefficient_sets\ (see "Reference Design Project Files" on page 8)

5.1 VP-API-II Core

The compile-time settings in `vp_api_cfg.h` controls the size of the compiled VP-API-II library, as well as the data memory (i.e., sizes of `VpDevCtxType`, `VpLineCtxType`, and device type specific device and line objects instantiated by the Application). So while a given installation may contain the source for several (or all) of the devices supported by the VP-API-II, there is no harm in retaining these files in the locations shown in Figure 5–2. Content that is unneeded by the Application will be removed in the compilation process.

Figure 5–2 Complete P2.24.0 Directory Content

📁 api_lib\
 📁 includes\
  🗎 vp580_api.h
  🗎 vp790_api.h
  🗎 vp880_api.h
  🗎 vp886_api.h
  🗎 vp890_api.h
  🗎 vp_adaptive_ringing.h
  🗎 vp_api.h
  🗎 vp_api_cfg.h
  🗎 vp_api_cfg_int.h
  🗎 vp_api_common.h
  🗎 vp_api_cslac_seq.h
  🗎 vp_api_dev_term.h
  🗎 vp_api_event.h
  🗎 vp_api_fxo_params.h
  🗎 vp_api_int.h
  🗎 vp_api_option.h
  🗎 vp_api_test.h
  🗎 vp_api_timer.h
  🗎 vp_CSLAC_types.h
  🗎 vp_debug.h
  🗎 vp_debug_colors.h
  🗎 vp_debug_masks.h
  🗎 vp_dtmf_detect.h
  🗎 vp_pulse_decode.h
  🗎 vp_timer_queue.h

📁 api_lib\
 📁 common\
  🗎 vp_api_common.c
  🗎 vp_api_config.c
  🗎 vp_api_control.c
  🗎 vp_api_init.c
  🗎 vp_api_query.c
  🗎 vp_api_test.c
  🗎 vp_debug.c
  🗎 vp_dtmf_detect.c
  🗎 vp_pulse_decode.c
  🗎 vp_timer_query.c

 📁 vp580_api\
  🗎 vp580_api_int.h
  🗎 vp580_control.c
  🗎 vp580_init.c
  🗎 vp580_query.c
  🗎 vp580_seq.c
 📁 vp790_api\
  🗎 vp790_api_int.h
  🗎 vp790_control.c
  🗎 vp790_init.c
  🗎 vp790_query.c

📁 api_lib\
 📁 vp890_api\
  🗎 vp890_api_int.h
  🗎 vp890_apiCal.c
  🗎 vp890_common.c
  🗎 vp890_config.c
  🗎 vp890_control.c
  🗎 vp890_fxo_control.c
  🗎 vp890_fxs_control.c
  🗎 vp890_init.c
  🗎 vp890_lp_control.c
  🗎 vp890_query.c

 📁 vp880_api\
  🗎 vp880_api_int.h
  🗎 vp880_abs_calibration.c
  🗎 vp880_control_common.c
  🗎 vp880_fxo_control.c
  🗎 vp880_fxs_control.c
  🗎 vp880_init.c
  🗎 vp880_linestate_control.c
  🗎 vp880_lp_control.c
  🗎 vp880_query.c
  🗎 vp880_tracker_calibration.c
  🗎 vp880_calibration_common.c

📁 api_lib\
 📁 vp886_api\
  🗎 vp886_adaptive_ringing.c
  🗎 vp886_api_int.h
  🗎 vp886_calibration_common.c
  🗎 vp886_common.c
  🗎 vp886_control.c
  🗎 vp886_events.c
  🗎 vp886_init.c
  🗎 vp886_query.c
  🗎 vp886_registers.h
  🗎 vp886_slac.c
  🗎 vp886_timers.c

5.2 Sample Customer Applications

This section refers to content in the (apps\) directory shown in Figure 5–1. The primary purpose of this content is to provide working example applications using the VP-API-II. These examples are designed to work on the Microsemi ZTAP Board. For these applications to work in the customer environment, some minimum changes are required to the following:

1. **System Service and HAL** - Customer Specific OS and Hardware Implementation Functions.
2. *deviceId* - Defined by customer for use with `VpMpiCmd()` to select the specific silicon CS. Note that `VpDeviceIdType` is defined to be (customer platform) HAL specific. Note in the example applications the use of `MPI_3WIRE`, `MPI_4WIRE`, `SM`, and `DIN` to create the `deviceId`. As mentioned, these are HAL (i.e., ZTAP) specific values.
3. **Device Profile** - IMPORTANT! This provides the PCLK, Supply Configuration and Supply Voltage. If incorrect, `VpInitDevice()` may return error the supply may not come up.

Microsemi has provided four VP-API-II example VP-API-II Applications, and a set of helper" functions. The directories are organized under (*install_dir*\apps\) as follows:

- **common\ (helper functions)**: This directory contains common board and print utilities that are used with the applications provided in the other three directories.
- **VE880_initialization**\: Basic VE880 VP-API-II Application.
- **VE890_initialization_1FXS_1FXO\:** Basic FXS and FXO Application using the VE890 silicon.
- **VE890_initialization_1FXS\:** Same as "VE890_initialization_1FXS-1FXO" with the FXO line object creation and functionality removed.
- **ZL880_initialization\:** Basic ZL880 VP-API-II Application.

5.3 System Service and Hardware Abstraction Layer (HAL) Examples

This section refers to content in the (arch\) directory shown in Figure 5–1. The content in these directories is for example only and must be replaced by a customer specific implementation. Please refer to the VP-API-II ZL880 Reference Guide for a description of the required System Service and HAL functions.

Each directory contains eight files, but only four need to be modified. The files are as follows:

Table 5–5 System Service and HAL Files

Filename	Comment
hbi_hal.h	No modification necessary
mpi_hal.h	No modification necessary
sys_service_c	**Customer specific System Service functions required**
sys_service_h	**VpSysDebugPrintf needs to be defined. Default is printk (for Kernel mode), or printf (for User space mode)**
vp_api_profile_type.h	No modification necessary. Profile types must be const uint8 data.
vp_api_types.h	**Customer specific Types required** Note that VP_PRINT_DEVICE_ID and VP_PRINT_LINE_ID are defined based on user defined types VpDeviceIdType and VpLineIdType respectively. These macros must be modified if these types are NOT compatible with integer type (i.e., printed using format string "%d").
vp_hal.c	**Customer specific HAL functions required**
vp_hal.h	No modification necessary

There are 3 System Service/HAL Example sub-folders under this (*install_dir*\arch\) directory:

- **examples**: This directory contains skeleton placeholders of the System Service and HAL functions required by the VP-API-II.

- **ztap\kernel**: This directory contains working code of the System Service and HAL functions required by the VP-API-II running on the Microsemi ZTAP board. This is similar to the content in **ztap\user_space** except the content in this directory is targeted for the Linux Kernel. Note that these files are designed to work on the ZTAP board and are provided as example only. They are provided just in case customers find them helpful while developing their own System Services and HAL functions.

- **ztap\user_space\:** This directory is similar to **ztap\kernel** except provides the required System Service and HAL functions for the VP-API-II running in User Space on the Microsemi ZTAP board (instead of the Linux Kernel). The files in this folder are also provided as example only.

5.4   Reference Design Project Files

This section refers to content in the (*install_dir*\coefficient_sets\) directory shown in Figure 5–1.

This directory (*install_dir*\coefficient_sets\) and its sub-directories contain the Profile Wizard P2.6.0 Project Files that have been designed, tested and approved to be used with the standard Microsemi reference designs. Each Profile Wizard Project file in these directories includes the AC, DC, Ringing,/Ringing Cadence, Tone/Tone Cadence, Caller ID (includes Caller ID Profiles for Type I and Type II Caller ID), FXO/Dialing, and Metering profiles for a wide range of countries. The files are organized by the following naming convention:

- Supply Configuration: 'BB' = Buck-Boost, 'IB' = Inverting-Boost, 'FB' = Flyback.
- Supply Mode: 'ABS' or 'Tracker'.
- Digital Interface: 'ZSI' or 'PCM'.
- Voltage: '100V' or '150V'.
- AC Coefficients: 'WB' = Wideband Set, 'NB' = Narrowband Set.

   Notes:

   *Customers **MUST** use a different set of AC Coefficients (Profiles) when changing between Wideband mode and non-Wideband mode. To support this, Microsemi has provided two sets of AC Coefficients: 'WB' and 'NB' (mentioned above). Applications should store each of these in an AC profile to be used with VP-API-II function* `VpConfigLine()` *(or* `VpInitDevice()`*/*`VpInitLine()`*) when changing Wideband mode in the Application. See the corresponding VP-API-II Reference Guide under the section for option* `VP_OPTION_ID_CODEC` *for Wideband mode settings.*

The Project Files have been organized in the directories by device family as follows:

- *VE880 Base PW Project Files: install_dir*\coefficient_sets\VE880\
- VE890 *Base PW Project Files*: *install_dir*\coefficient_sets\VE890\
- ZL880 *Base PW Project Files*: *install_dir*\coefficient_sets\ZL880\
- miSLIC *Base PW Project Files*: *install_dir*\coefficient_sets\miSLIC\

IMPORTANT: Application Note:
1. Customers are strongly encouraged to use the Project files provided by Microsemi to generate the profiles used in the Application rather than create their own (excluding minor changes in the Device Profile normally required to work in their system). Customers that start a design using tested Microsemi profiles generally avoid some of the minor errors that tend to slow down normal product bringup.
2. When upgrading from a previous release to a new release, it is always recommended to upgrade the profiles as well. Quite often, new features or improvements will be available with both an VP-API-II upgrade as well as a specific Profile upgrade.

For more information about all Microsemi products
visit our website at
www.microsemi.com

TECHNICAL DOCUMENTATION – NOT FOR RESALE