Vantage User Manual

Vantage Software Release 4.0.1-1903121200, 12 March 2019

This document is copyright 2001-2019 Verasonics. All world-wide rights and remedies under all intellectual property laws and industrial property laws are reserved. Verasonics Registered U.S. Patent and Trademark Office.

Table of Contents

1 Introduction

Thank you for your purchase of a Verasonics Vantage system. This document includes safety guidelines for proper use of the system, instructions for new customers on setting up and using the Vantage System, reference material on system administrative utilities and diagnostic functions, and a summary of other documentation included with the system or available from Verasonics.

Customers with an existing Vantage system using an earlier software release should also review this document, since recent Vantage Software releases have made significant changes in terms of both how the software is installed and how it should be used.

Document Change History

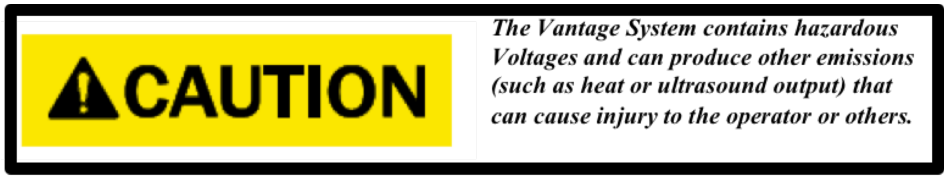
Date MM/DD/YYYY	Section(s)	Description
02/11/2019	Most	Vantage 4.0.1 software release
11/27/2018	All	Vantage 4.0.0 software release, Matlab compatibility restrictions, new UTA modules, incorporating changes described in the "Release Notes" document.
05/30/2018	2 Safety and Regulatory, 4 (compatibility updates), 11 (license for new impedance measurement tool)	Vantage 3.4.2 software release, incorporating additional safety warnings and user guidelines from revision E of the Vantage Risk Analysis Document.
11/17/2017	All	Vantage 3.4.0 software release, incorporating changes summarized in the "Release Notes" document.
1/27/2017	All	Vantage 3.2 software release, incorporating changes described in the "Release Notes" document.
11/01/2016	All	Vantage 3.1 software release, incorporating changes described in the "Release Notes" document.
02/02/2016	All	Vantage 3.0 release (minor updates and references to new 3.0-specific user documents).
09/04/2015	All	Initial 2.11 release; merge updates from Word version of Vantage User Manual with earlier *Getting Started* document.

2 Safety and Regulatory

Explanation of Symbols

Symbol	Meaning	Explanation
	"Refer to User Manual"	This symbol is used to direct the reader to this document.
	"Caution"	This symbol indicates a potentially hazardous situation which could result in minor or moderate injury. The User Manual (this document), and in particular the following Safety precautions must be consulted in all cases where this symbol appears.
	"Hazardous Voltages"	This symbol indicates the presence of hazardous voltages inside the cover, and that there are no user-serviceable components inside. *Do not remove side covers!*

Safety Precautions



Operator: It is important that you follow these Verasonics safety requirements to protect both yourself and the Verasonics System. Please require all new users of the system to become familiar with the safety instructions in this manual.

- **System Selt-Test** After initial system setup (and after any software installation change), run the built in system self-test routine (at the Matlab command prompt, type "VerasonicsVerificationTest"). Be sure that no transducer is plugged into the connector(s). Notify Verasonics Customer Support if there are any difficulties running this test or if any errors or failures are reported. This same test should be repeated after any other activity that might affect system functionality, such as transporting the system, disconnecting/ reconnecting system cabling or other partial disassembly, installing new SW, etc.
- **Never operate the system without all covers securely in place.** The covers are required for system cooling, EMC shielding, and protection of the user and the system. There are potentially hazardous voltages at exposed points inside the system enclosure. There are no user serviceable parts inside. The system should be returned to Verasonics for any HW-level repair, upgrade, or other maintenance. Evidence of unauthorized removal of the side covers may result in termination of the system warranty.
- **Do not store liquids on or near the system,** and prevent spilling any liquid on or into the enclosure to protect the user and the system from an electrical fault. If a spill occurs, immediately shut the system down (computer and VDAS) and notify Verasonics Support. Make sure that any measures taken to protect the system from spills do not also restrict cooling airflow.
- **Custom transducers may be poorly shielded** and may interfere with other instrumentation, and extra care must be taken in the presence of medical devices sensitive to RF radiation. If the system will be used in an environment where this may be a concern, the customer should conduct EMC tests of the overall system configuration (transducer, VDAS unit, host computer, and any other connected devices). EMC tests of the individual components is not a meaningful way to predict EMC performance of the overall system.
- **AC Line power isolation and grounding:** To build up a functional system, the VDAS unit must be interconnected with a host computer, display monitor, and potentially other devices each of which may have their own AC line power

connection. The user must be aware that the net AC line leakage current of these interconnected devices may exceed safe operating limits. Other errors in the AC line power connections, such as inadequate ground connections, may lead to potentially severe safety hazards to the operator and/or subjects of the system. It is recommended that the user subject the entire interconnected system to AC line safety tests for isolation, leakage current, and ground impedance. These AC line safety tests should be repeated whenever the system has been disassembled and reassembled, or subjected to rough transport or other environmental stresses.

- **System Self-test Reminder:** It is recommended that VerasonicsVerificationTest always be run after the system has been updated (software or hardware), transported, or disassembled and reassembled. To run the test, simply start up the system, open Matlab, navigate to the Vantage project folder for the Vantage software installation to be used, type "activate" from the Matlab command prompt and then type "VerasonicsVerificationTest", and respond to the prompts and dialog boxes as the test proceeds. It should complete and exit with a comprehensive "Pass" or "Fail" message at the end.
- **System Operating Environment:** Make sure the operating environment in which the Vantage system will be used conforms to the requirements given in this manual and the system specification in terms of ambient temperature, AC line power(voltage, frequency, stability), cooling airflow restrictions, mechanical shock and vibration, etc. Use outside of these limits may result in hazardous conditions.

Additional Safety Warnings for use of the system on humans or live animals

The Vantage system as supplied by Verasonics is not certified for use on human subjects, but it can be safely used in this context if the following guidelines are adhered to and the user has gone through the appropriate IRB or other regulatory/safety reviews. Refer to the "Verasonics Safety Guidelines for Imaging Human Subjects" and "Vantage Risk Analysis" documents (in the Documentation / Safety subfolder) for additional information.

- **Operator Training:** The Vantage system user interface is designed for flexibility and ease of use in the research environment, and differs significantly from the user interface provided by typical commercial ultrasound systems designed for use in a clinical environment. Clinicians or sonographers who will be using a Vantage system with human subjects should be given training and guidance on how to use the system.
- **Probe Patient Contact:** Custom probes or probes purchased from a third party supplier should be reviewed and tested for potential hazards associated with direct patient contact such as biocompatibility, sterilization, electrical isolation, surface temperature, acoustic output, etc. This also applies to probes purchased through Verasonics.
- **Acoustic Output Control and Testing:** It is the user's responsibility to conduct sufficient testing and characterization of acoustic output per the applicable regulatory and safety requirements, when using the Vantage system on human subjects.
- **User Script Verification:** The Vantage system is designed to be very flexible and easily programmable for a wide variety of applications. An inherent risk in this flexibility is that a user's system programming error might result in significantly different system behavior or acoustic output than what was intended, with no warning or fault condition indication from the system. It is the user's responsibility to conduct sufficient verification testing of their scripts (in conjunction with the probes to be used) to ensure the actual output and functionality matches what was intended, before using the system on human subjects. This same caution applies doubly to the definition of transmit waveforms and transmit beam steering/focusing: Sufficient testing of the transmit waveform and acoustic output must be conducted to ensure the actual output waveform and power level match what was intended and are within the required regulatory/patient safety limits.
- **Hardware-Software Handshaking and Fault Detection:** When the Vantage system is running, the actual acquisition and processing functions are divided into two independent event sequences that are very loosely coupled with each other: one running within the hardware system, and the other in software on the host controller. This "loose coupling" creates a potential hazard condition for a system being used on human subjects, if one of the sequencers has malfunctioned and the other is not aware of it. A mitigation for this potential hazard (as identified in the Vantage Risk Analysis document) is to implement "watchdog functions" in both the hardware and software sequencers, that will trigger a fault condition if the other sequencer does not respond to a periodic query within a user-programmable time limit.
  - The Vantage system includes a software watchdog timer running in the hardware system, that will terminate execution of the hardware sequencer and trigger a system fault condition if the timer has not been reset by the end of the timeout interval. The timer is reset periodically by the Vantage software running on

the host computer, and thus will not be reset if the software has locked up or crashed. This watchdog feature is always enabled whenever the system is executing a sequence. Refer to the Sequence Programming manual for details of the software watchdog, and how to program the timeout interval.

  - The Vantage system does not include an automatic watchdog going the other way, to allow the system software to determine whether the hardware sequence is functioning properly. It is up to the user to insert something in the event sequence that can be used as a "hardware watchdog". The simplest way to do this is with a 'sync' command, which includes a user-programmable timeout interval. If the software event sequence is waiting at a sync command point for the signal from the hardware system that it has also reached that point, and the sync timeout interval expires before that signal is received, the software will report an error condition and force both sequencers to stop. Similar hardware watchdog handshaking can also be achieved through the use of synchronous DMA transfer ("transferToHost") commands, that have a user-programmable DMA timeout interval which can be used to shut down the system with a DMA timeout error condition. Note however that in some operating states the DMA timeout may be disabled even though synchronous DMA transfers are being used (one such example is Recon processing of the RF data while using the -1 "most recent frame" RF data frame pointer). Refer to the Sequence Programming manual for details on how to use the sync command and synchronous DMA commands, and how to program their respective timeout intervals.

- **Audible Noise:** The cooling fans used in the Vantage system plus those in the host controller and any other devices that may be included in an investigational system may lead to a combined audible noise level that could be annoying or distracting to the operator and/or a human subject. Testing and evaluation of the overall audible noise level should be conducted before using the system in a clinical environment.

Damage in Shipping and Handling

When a new system has been received (or when a system has been shipped back to you after an upgrade or repair), carefully examine the shipping container for any signs of physical damage. Also examine the shock sensors attached to the shipping carton; notify Verasonics immediately if any of them have tripped, or there are any other signs of damage. After unpacking the system, also inspect it carefully for any evidence of physical damage.

EMI Immunity

If obvious noise appears in the ultrasound image there may be a radiated or conducted immunity problem. Please contact support@verasonics.com.

Protective Grounding

On installation of your Vantage system, ensure that the system is connected to an outlet that provides reliable protective grounding. Also ensure that the system is connected to the grounded outlet with the appropriate three pronged (grounded) plug supplied by Verasonics.

System Transportation

Remember to be careful when lifting the Vantage system – it weighs up to 44 lb. (20 Kg.). A system with the HIFU option installed can weigh up to 57 lb. (26 Kg.). Before lifting the system, make sure all cables have been disconnected. Lift it from the side with your arms at the front and back, and your fingers extending under the bottom of the chassis. Do not attempt to lift or move the system using the handles of the UTA adapter module!

When transporting the Vantage system always use the shipping box provided by Verasonics. The box is designed to protect the system when transporting or moving the system and has handles for easy lifting. Remember to bend your knees.

3 Vantage System Overview

The Verasonics Vantage system includes the Vantage hardware unit and the Vantage software.

The Vantage hardware unit consists of a single chassis which may contain one or more Acquisition Boards, and other supporting circuit boards, including the system Backplane (BKP), Transmit Power Controller (TPC), and a UTA Baseboard or ScanHead Interface (SHI).

The Vantage software consists of several subcomponents, the Verasonics Matlab scripts (as well as user-defined scripts), which are the application-level software running on Matlab, the Verasonics Hardware Abstraction Layer (HAL), and other supporting items such as diagnostic tools and hardware device drivers. The Vantage software also includes code for the FPGA devices located in the Vantage hardware unit. This code must be downloaded to the hardware system whenever a new version of the software is installed.

> **Note**
> The system requires a user-licensed version of Matlab and the Signal Processing Toolbox. Activation of the user's Matlab license must be done prior to running the Verasonics software.

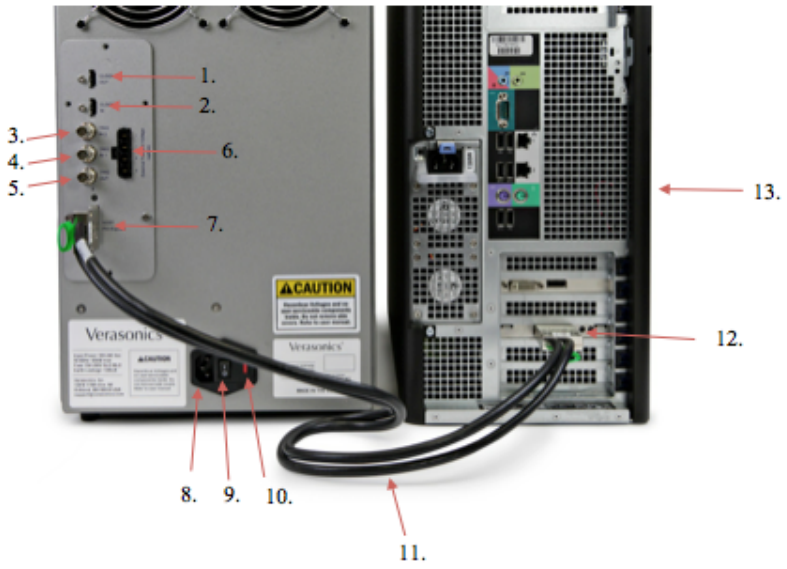
Connections

Front View

1. Scan Head Interface, Probe Connection

NOTE: Your system may look slightly different than the above image. The functionality will be the same.

Rear View



1. Clock Out
2. Clock In
3. Trigger In 2
4. Trigger In 1
5. Trigger Out
6. External Transmit Power connection (HIFU option ONLY)
7. Vantage PCIe Connector
8. AC Power Connector
9. Power Switch
10. Fuse Holder Location
11. PCIe Cable (1 meter)
12. Host Controller PCIe Connector
13. Host Controller (for Host Controller Information refer to Dell User Manual that comes in computer box.)

Technical Specifications

AC Line Power

- Vantage HW system: 100-240 VAC, 50-60 Hz; 600 W. max
- host controller, and display monitor: These units have their own separate AC line connections. Refer to the user manuals provided by their manufacturers for line power and other environmental specifications.
- HIFU Option: The external power supply for HIFU use also has its own separate AC line connection. Refer to the manufacturer's user manual provided with it for line power and other environmental specifications.

Host Controller Requirements

If you did not purchase a host controller through Verasonics and are setting up your own computer for use as the Vantage system host controller, it must provide the following minimum features:

Physical Platform:

- A PCI-express expansion slot that can accept the Host Adapter card provided with the Vantage system. For full performance this expansion slot must provide a PCIe interface of at least 16 lanes at Gen 2 speed or 8 lanes at Gen 3 speed. Note that the system will automatically adapt to the PCIe interface that is available even if it does not meet these requirements; in this case the system will still be fully functional and no errors will be reported but DMA transfer rates will be reduced. If you are using a computer that does not provide PCIe expansion slots you may be able to connect an external device to support the host controller through a Thunderbolt or other connection to the computer, but DMA transfer rates may be reduced. Contact Verasonics Technical Support (support@verasonics.com) for additional information.
- At lest 16 Gigabytes of system memory installed in the host computer. If you will be acquiring and processing large volumes of RF data from the Vantage system or reconstructing large images at high frame rates, additional memory should be installed. The Vantage system and software can utilize all the host controller memory that is available, up to the limits supported by Matlab and the host controller with its Operating System.
- The CPU processor must support the SSE 2 vector processing instruction set, and should support multicore processing with at least 8 cores. Future Vantage software releases will include updates to take full advantage of the AVX 2 instruction set so if you are buying a new computer it should ideally provide the AVX 2 level of functionality.

Operating System: Any of the following three operating systems are supported by the Vantage software:

- Windows: Windows 7 Ultimate. (Windows 10 can also be used, but there are a few known anomalies when the Vantage software is used with Windows 10. Contact Verasonics Technical Support (support@verasonics.com) for details.)
- MacOS: Yosemite 10.10.5 or later. More recent versions of MacOS beyond Yosemite can not be used with the Vantage software. Contact Verasonics Technical Support (support@verasonics.com) for details.
- Linux: Ubuntu Version 16.04 LTS. Note that earlier Vantage releases up through 3.3.0 required Ubuntu version 14.04 and are not compatible with 16.04. Similarly, the 3.4.N releases cannot be used with Ubuntu version 14.04. If you were previously using Vantage 3.3.0 or earlier, you will first have to install the Ubuntu Version 16.04 LTS operating system before installing the 3.4.N software release and then install the new version of the driver for the Vantage hardware system. If you want to switch back to the older Vantage software releases, you will have to re-install the older Linux OS and driver. See the "Vantage Software Installation" document for detailed instructions. Note however that all of the 3.4.N releases (3.4.0, 3.4.1, and 3.4.2) use the same version of Linux and the driver so you can easily switch between installations of those releases without changing the OS or driver.

Matlab: The Vantage 4.0 software releases can be used only with Matlab versions 2017a and 2017b; Verasonics recommends the use of 2017b. Matlab versions older than 2017a do not provide features used by the Vantage 4.0 software and cannot be used. In addition, Matlab versions 2018a and 2018b have changed the format of some data types, making them incompatible with the current Vantage software releases. Verasonics is working to resolve this issue as quickly as possible, to allow full compatibility with current Matlab releases in a future Vantage software release. For the current Vantage 4.0 releases an error will be reported and the system cannot be used if any Matlab version other than 2017a or 2017b is present. Contact Verasonics Technical Support (support@verasonics.com) for the latest information on Matlab compatibility.

Additional Specifications

For full specifications on environmental operating conditions and system performance, refer to the *Vantage Product Specification* document.

Power On Procedure

Remember to ensure that the system is connected to an outlet that provides reliable protective grounding and that the system is connected to the grounded outlet with the appropriate three pronged (grounded) plug supplied by Verasonics.

First turn on the power switch on the back of the Vantage system. Next, turn on the computer. The Vantage system detects the computer power signal through the PCIe cable and then fully powers up the hardware. The cooling fans should turn on, and an LED indicator behind the front panel transducer connector also turns on.

Power Off Procedure

Shutdown your computer. Since the Vantage system detects the computer power signal through the PCIe cable, the system will automatically power off (fans off).

You may leave the Vantage system power switch in the ON position as it will only consume full power when the computer powers up.

4 Vantage Software and Hardware System Compatibility Constraints

The Vantage Hardware System is now available in a large number of system configurations and optional hardware-level features. Support for all Hardware system configurations is included in a single Vantage software release, so even if you have multiple Vantage HW systems in different configurations you can run them all with the same Vantage SW installation. (At system power-up and initialization the software checks for the presence of a HW system, and if one is found the software automatically configures itself for use with that particular hardware configuration).

In general, Verasonics strives to make each new software release "backward compatible" to all older hardware system configurations and also to user acquisition scripts that were developed on older versions of the Vantage software and/or hardware. There are exceptions to this rule, however, such as situations where maintaining backward compatibility would be prohibitively complex or would lead to unacceptable performance compromises. Listed below are specific situations where a compatibility restriction exists, and also some general guidelines on what you may need to do when installing a new software release for use with an older hardware system.

Supported Vantage Software Releases

Verasonics will now provide the latest Vantage software releases to all existing Vantage customers in good standing, even if their hardware system is out of warranty and they do not have an active service agreement. If you request customer support from Verasonics and you are not using the current Vantage software release, we may require you to update your software installation before we can respond to your request. However, Verasonics recognizes that some customers may need to continue using an older release if they are in the middle of an ongoing research project or clinical study, and we will provide support and access to the older releases as needed in those situations.

Obsolete Vantage Hardware Configurations

P00732-02 Acquisition Module and P00729-03 Backplane These two hardware module revision levels were used only in the earliest Vantage systems, built in 2013 or early 2014. They are supported by the 3.0.N Vantage software releases but have been declared obsolete and cannot be used with 3.2.0 or later software. If you have a hardware system containing either of these modules and would like to use more recent Vantage software releases, a hardware upgrade will be required. Contact Verasonics Technical Support (support@verasonics.com) for assistance.

New Hardware Configurations not supported by older Software

As the Vantage product line evolves, Verasonics frequently releases new hardware-based features and/or new versions of hardware modules offering improved performance. In many cases the new hardware features cannot be used with older versions

of the Vantage software, and in some cases the older software may not function at all. Contact Verasonics Technical Support (support@verasonics.com) for more information and assistance, if you have a situation that requires the use of an older Vantage software release on a newer hardware system.

User Setup Scripts (and Verasonics Example scripts) Compatibility

Generally speaking, acquisition scripts developed for use with an earlier Vantage software release can be used with new releases as they become available, but please be aware that this compatibility applies to the "SetUpxxx.m" script itself and not to the .mat file it creates!! When switching to a different software release you must re-run the script with the new software to create a new .mat file; do not expect the older .mat file to be usable with a different software release.

Note also that significant changes were made to example script functionality in the 4.0 release. Scripts written for 3.4 or earlier may need some modifications to get them to run optimally with 4.0 and later releases. Refer to the document "User Guide to Changes in 4.0" (included in the Documents folder of the Vantage 4.0 software installation) for guidelines.

5 Software Installation and Initial Setup of the Vantage System

> **Note**
>
> Please keep the shipping box and its contents. You must use them if it becomes necessary to ship your Vantage system back to Verasonics for any reason. The shipping box has been custom designed to keep the system safe during shipping. You will be charged for a new shipping container if your system needs to be returned to Verasonics and you no longer have the original shipping box.

The steps needed to set up a system with the Vantage 4.0.1-1903121200 release will depend on the situation. Of the four sub-headings shown below for this section of the document, select the one that most accurately describes your situation. It will guide you through the specific steps needed, including references to other documents in the Documentation folder.

The Documentation folder can be accessed through the customer download folder on the Verasonics server for your customer account, in the folder named Release 4.0.1-1903121200. A copy of the Documentation folder with all of the same contents will also be found in the "Vantage project folder" after the SW installation is complete.

> **Note**
>
> The term "Vantage project folder" is used throughout the Vantage system documentation to refer to the main folder containing an installation of the Vantage software, with a default name such as "Vantage 4.0.1-1903121200". When using the Vantage software, the Vantage project folder **must** be selected as the current Matlab working directory.

New Vantage System including New host controller purchased through Verasonics

If you just received a new system from Verasonics including the host controller, software installation is not needed because the host controller has been preconfigured by Verasonics with a full Vantage 4.0.1-1903121200 software installation. To set up the system, follow the instructions in the *Vantage Initial Setup* document that should be included with the system in the shipping carton, but can also be found in the Documentation folder.

New Vantage System using User-Supplied host controller

If you just received a new Vantage hardware unit from Verasonics but are providing your own host controller, use the following instructions to configure the host controller for use with the system and then install the Vantage 4.0.1-1903121200 software.

1. Carefully unpack your Vantage system. The shipping carton should also contain the PCIe adapter card for your host controller.

2. Install the Verasonics PCIe adapter card into your computer. You will need to install it into a PCIe expansion slot that allows either "PCIe Gen 3 x8" or "PCIe Gen 2 x16" data throughput to achieve maximum throughput from the Vantage system. The computer you choose is arbitrary as long as it meets that PCIe expansion slot requirement and supports one of the operating systems supported by the Verasonics System Software. In general, the Verasonics System Software, which will run on your computer from within Matlab, will benefit from faster CPU speeds and a greater number of CPU cores, additional RAM, and faster graphics displays. The software is highly parallelized, and optimized to use as many CPU cores as are available.
3. With both Vantage system and computer powered off, connect the PCIe cable to the receptacle at the back of the Vantage system and to the Verasonics PCIe adapter card receptacle at the back of your computer. Be sure that the PCIe cable is fully seated at both ends (i.e., you should feel a "click" when properly seated).
4. Install the Vantage 4.0.1-1903121200 software. Follow the instructions in the *Vantage Software Installation Instructions* document, that can be found in the Documentation folder.

Updating an Existing Vantage System to the 4.0.1-1903121200 Software

If you have a fully configured Vantage system running an earlier release of the Vantage software, go to the *Vantage Software Installation Instructions* document, that can be found in the Documentation folder. The instructions given in that document will guide you through installing the 4.0.1-1903121200 release and disabling the earlier release. The earlier release will be left in place, however, so you will be able to easily switch between both software installations if you choose to do so.

Stand-Alone Software Installation on a User's Computer

If you have obtained "stand alone" simulation-only SW licenses from Verasonics for use of the Vantage software on additional computer(s) that will not be used with a Vantage hardware system, the software installation process is exactly the same as for a host controller that will be used with a Vantage hardware unit. Follow the following steps.

1. Install the Vantage 4.0.1-1903121200 software. Follow the instructions in the *Vantage Software Installation Instructions* document, that can be found in the Documentation folder. If the computer is using the Windows operating system, you do not need to install the driver after completing the Vantage software installation. The absence of the driver will indicate to the Vantage software that this is a stand-alone installation and thus no attempts to communicate with a hardware system will be made. However, if your computer is using Linux or MacOS, you will still have to go through the additional steps to install the driver as explained in *Vantage Software Installation Instructions*. This driver requirement for Linux and MacOS will be removed in future Vantage software releases.
2. If this will be the first time this computer will be used with a Vantage software simulation-only installation, you will need to obtain an updated license file from Verasonics that includes licensing for this specific computer. Note however that you can complete the software installation, and run utilities such as *Version* and *systemInfo*, even if the computer does not yet have a valid simulation license. After the software installation is complete, run the *systemInfo* utility to obtain the information you will need to request an updated license file from Verasonics. Refer to the *Software Licensing* section of this document for additional information on requesting and installing an updated license file.

6 Final Setup and System Tests

After completing the initial setup and (if required) software installation described in the preceding sections, your system is now ready for use but a few additional steps are needed to make sure everything is working properly.

Please read through all of the instructions in this section carefully. This is especially important for experienced users who are familiar with earlier V1 or Vantage releases, since significant changes have been made in each new release.

Matlab License Activation

Before you can use the Vantage software, you must have Matlab and the Signal Processing Toolbox installed on your host controller. If you have just received a new host controller purchased through Verasonics the Matlab and Signal Processing Toolbox applications will already be installed, but you must activate them with licenses you provide. Please see the Mathworks website for further details on obtaining and activating your Matlab license.

You can contact Mathworks at https://www.mathworks.com.

Use of the *activate* Command

Vantage project folders must be activated before they can be used. The command for activating a Vantage project folder is *activate*. This command was introduced in the 2.11 release; it did not exist in earlier Vantage software releases.

From within the Matlab environment, the procedure for activating a Vantage project folder is:

1. Set the Matlab current directory to the root of the Vantage project folder you are activating.
2. Issue the *activate* command from within the Matlab console window.

These steps must always be completed whenever you start Matlab or when switching to a different Vantage project folder.

> **Note**
> If you are switching between Vantage project folders and have already run activate in the one you are switching away from, you *must* quit Matlab, then restart it and select and activate the Vantage project folder you are switching to. (You do not need to restart the host controller.) Failure to do this may result in erratic operation or a Matlab crash.
> Doing this helps to ensure that shared libraries and Matlab extensions loaded in the Matlab application memory space are of the same revision level.
> If you are switching between different SW versions that require different FPGA files in the HW system, the SW will prompt you that an FPGA update is required which will then require a system power cycle to take effect. This will be the case when switching between 3.4 and 4.0 since several FPGA files were changed between these releases.

Verifying Software Installation

At this point, if all of the preceding steps have been completed successfully, the Vantage software and Matlab installation should be ready for use. We recommend that you first confirm the SW itself is functioning properly with no Vantage HW system connected. If you have a Vantage HW system connected to your host controller, shut down the host controller to a full power-off state. Then turn off the power switch on the rear panel of the Vantage system, and/or disconnect the PCI Express cable assembly. Next, proceed with the following steps.

1. Restart the host controller and log in.
2. Launch the Matlab application.
3. When the Matlab user interface is available, select the 4.0.1-1903121200 Vantage project folder as the Matlab working directory and then issue the *activate* command at the Matlab command prompt.
4. Confirm that no error messages are given during the system startup process.
5. Issue the *Version* command at the Matlab command prompt and confirm that the software release version number is report correctly, and that no error or warning messages are displayed.

If any of the above steps result in error or warning messages, please contact Verasonics Technical Support (support@verasonics.com) for assistance. Please do not attempt to use your Vantage system if there are questions as to the integrity of the installed software.

Updating Your Vantage Hardware FPGA Code

If the Vantage software is functioning properly, you are now ready to run it with the Vantage HW system. From time to time Verasonics releases updated code for the FPGAs in your Vantage hardware unit. This release does provide new FPGA code. Please perform the following steps to update the FPGA code in your Vantage hardware.

1. Properly shut down your host controller. Your Vantage system should already be powered off.
2. With the host controller and the Vantage system both powered off, make sure the PCI Express cable between the host controller and the Vantage hardware unit is properly connected.
3. Ensure that the power switch on your Vantage hardware unit is set to the on position. Note that your Vantage hardware unit will not turn on at this time; it will only turn on when your host controller system powers up.
4. Power up your host controller and observe that both the host controller and the Vantage hardware unit power up. You should be able to hear the fans in the Vantage hardware unit when it powers up.
5. Log in to the host controller and launch the Matlab application.
6. Issue the *Version* command at the Matlab command prompt, after the Vantage project folder has been selected and activated.
    a. Confirm that the version number for this software release is 4.0.1-1903121200.
    b. Follow the command prompts in order to update the FPGA code in your Vantage hardware unit.
    As the update proceeds, you will see periodic status updates on the progress being made.
    It is critically important that you allow this process to complete on its own (some steps may take several minutes). If you interrupt it for any reason, your Vantage hardware unit may be left in an inoperable state.
    c. When the status updates indicate that the updates are complete and a power-cycle is required, close Matlab and shut down your host controller to a completely power-off state.
7. Restart your host controller in the normal manner. Your Vantage hardware unit should power up as well.
8. Start Matlab and run the *Version* command again. At this point in the process, it should complete with no error or warning messages of any kind.

> **Note**
> If the *Version* function failed to successfully complete the FPGA updates, you may need to do a forced update of all FPGA code using the *reprogramHardware* command (shutdown and restart the system from a full power off state, select the desired Vantage project folder and run activate, and then issue *reprogramHardware* at the Matlab command prompt and follow all prompts, noting this process will take several minutes. When *reprogramHardware* finishes, it will notify you that another full power down and restart cycle is required.) If you experience any difficulties or other issues with this procedure, please contact Verasonics Technical Support (support@verasonics.com) for assistance.

Software License File Updates

The software license file for your organization will be installed by the installer when it runs. The actions taken by the installer should be sufficient to ensure that your Vantage software is properly licensed by the time the installer finishes. If the *Version* utility does not report any licensing problems, you can skip this section and proceed directly to *System Self-Test: |VerasonicsVerificationTest Utility*.

The name of the software license file is *license.enc* and it exists at the top level of your Vantage project folder.

If you receive an updated license file from Verasonics, it may be named *license.enc.v2*. To install it, you should:

1. Delete or rename any existing *license.enc* file in your Vantage project folder.
2. Save the new license file into your Vantage project folder.

3. Rename the saved license file from *license.enc.v2* to *license.enc*.
4. Restart your Matlab environment.

When you have restarted your Matlab environment, your Vantage software should be properly licensed.

All questions and concerns related to software licensing should be sent to the email address licensing-request@verasonics.com.

When you do contact licensing-request@verasonics.com, we ask that you please attach a copy of the output from the *systemInfo* command. You generate this output as follows:

1. Launch your Matlab environment.
2. Navigate to the Vantage project folder you are using.
3. At the Matlab console prompt, issue the *systemInfo* command.
4. The *systemInfo* command writes a text file, *systemInfo.txt*, in the current directory.
5. Please attach the *systemInfo.txt* file that was generated to the email you will send to licensing-request@verasonics.com.

More information on Vantage software licensing is available in the *Software Licensing* section at the end of this document.

System Self-Test: *VerasonicsVerificationTest* Utility

As a final step in preparing your system for use, you should run this self-test script to verify that all digital and analog hardware subsystems are operating within specifications. The test is very easy to run, but should only be performed after the software and FPGA code are known to be properly installed. Before running the test it is a good idea to do a system power shutdown and restart, to ensure the system is in a known operating state. Also make sure nothing is connected to the system's transducer connectors. Then, at the Matlab command window prompt:

1. Navigate to your Vantage project folder and run *activate* if needed.
2. Type *VerasonicsVerificationTest* in the command window and press Enter.
3. On the GUI window that pops up, click the "System Verification Test" button and then respond to the prompt(s) that come up in the Matlab command window. The test takes a few minutes and the final result should indicate "ALL TESTS PASS" in the command window. If any test fails, and the "TEST FAILED" message appears, copy the output in the command window and email the text to support@verasonics.com.

7 Documentation, System Tools and Utilities

Summary of Contents of the Documentation Folder

This section provides a list of the documents you will find in the *Documentation* folder within the Vantage project folder, after the Vantage software has been installed on your host controller. A copy of this same documentation folder and all of its contents is also provided in the release folder for this release, within your customer download folder. This allows you to have full access to all of these files, even if you have not yet installed the software. When a new Vantage software release is announced and becomes available in your customer download folder, it is a good idea to review the documentation (especially the Release Notes and Software Installation Instructions!) before you start the installation process, so you will know what to expect and be aware of any changes to the installation procedure.

All documentation files are in .pdf format; you do not need to have Matlab running to access these files. Here is a listing and overview of the documents you will find in the Documentation folder:

Vantage User Manual (This document) – Guideline for first time users of the Verasonics System. This is the document you are reading now, and is required reading for all users.

Vantage Release Notes Important release information describing new features, bug fixes, and known errata with the current software release as well as a history of earlier software releases.

Vantage Software Installation Instructions This document provides detailed instructions on how to download and install the current release of the Vantage system software.

Vantage Initial Setup A duplicate copy of the document included in the shipping carton when you receive a new system from Verasonics. It provides a quick summary of unpacking, connecting, and starting up the system for a new customer.

Vantage Sequence Programming Tutorial An introduction to the Verasonics System and its programming interface, with a detailed user scripting example, written in a narrative form. This document is recommended reading for all users, and required reading for any users who wish to develop their own control scripts. The Vantage 3.N SW releases include numerous enhancements to the processing and display signal path, and to fully take advantage of these new features you will need to make changes in your SetUp scripts if you have scripts that were developed with 2.11 or earlier Vantage SW releases. The 3.0 version of the Vantage Sequence Programming Tutorial has been extensively updated to cover these changes.

Vantage Sequence Programming Manual The comprehensive User's Reference Guide to programming the Verasonics System from within the Matlab scripting environment. This document has now been updated to cover changes made to the system up through the current release.

AppNote_Receive Data and DMA Management Detailed information on receive RF data temporary storage in the Vantage hardware system and DMA transfers to the memory buffer in the host computer including DMA transfer rates, memory allocation and buffer size limitations, and use of the new subframe DMA feature that has been added to the system starting with the 3.2.2 release.

CDI Weak Flow Filter Responses Describes performance and provides frequency response plots for the Weak/Low-Flow wall filter coefficient sets available in the Doppler Process Object, as described in section 3.5.2 of the Sequence Programming Manual.

Vantage Product Specification Lists the features, options, and performance specifications for the Vantage system.

Vantage Trouble Shooting Lists some tips and techniques for avoiding common operator errors, correcting problems with the hardware system, and identifying and correcting programming errors in user-defined SetUp scripts.

Verasonics Safety Guidelines for Imaging Human Subjects Located in the *Safety* subfolder – Assistance for researchers intending to apply for IRB approval to do ultrasound imaging with human subjects.

Vantage Risk Analysis Located in the *Safety* subfolder – Documentation of the Risk Analysis that has been conducted by Verasonics for the Vantage system. This will be a useful reference document for customers planning to use their system on human subjects, in preparation for an IRB review or other regulatory/ safety reviews.

Other Documentation Included in the Software Release Package

Additional .pdf documentation files have been included at other locations within the Vantage project folder, so they can be easily found in the same location as the SW utilities they describe. Some of those documentation files that you should be aware of are listed here:

Script Matrix and Description This document is located in the *ExampleScripts* folder within the Vantage project folder. It provides a cross-reference summary of the types of example scripts included in the Example Scripts folder, and which of those types are included in the folders for each of the individual Probes supported by the system. A one-paragraph summary description of each script type is included in the document, following the matrix. By reviewing this matrix, you can easily identify and locate scripts you may be interested in without having to browse through all of the individual probe directories. Additional more detailed documentation for many example scripts can be found in comments included in the example script itself.

Specialized Applications Example scripts for several specialized applications, and/or associated features of the Vantage system, can be found in the *Examples_Biomedical / Specialty_Applications* subfolder of your Vantage project folder. Many of these example subfolders include additional pdf documentation or application notes files.

System SW Tools and Utilities

A number of useful tools and utilities can be found in the *Tools* subfolder within the Vantage project folder. Additional high-level system management and self-test utilities are located at the top level of the Vantage project folder. An overview of each of these utilities is listed here:

Version This function, invoked from the Matlab Command prompt, provides a quick summary of a system's software and hardware configuration and software license status, all displayed in the Matlab Command Window. It is a valuable check to use as a quick confirmation that there are no obvious problems with the overall state of the system. A more comprehensive system configuration report can be obtained by adding an argument to the function call, for example by typing *Version x* (the value or character used as the argument does not matter). In this case the output report will include the part number and serial number of all major HW modules in the system, FPGA code revision levels, etc. *Version* will also check the status of FPGA code installed in the Vantage hardware unit and ask the user whether to proceed with updating any FPGA revisions found that are incompatible with the SW release being used.

reprogramHardware This utility will force revision checks and updates to all FPGA code files in the HW system. It can be used to recover from a situation where one of the installed FPGA files has been corrupted, preventing high-level functions such as *Version* from functioning properly (for example, if FPGA updates through the Version command did not complete successfully because the system was shut down prematurely while the updates were still in process). When using the *reprogramHardware* utility, pay attention to the prompts and status updates displayed in the Matlab Command Window. In some cases an FPGA file update will take several minutes, and the system must not be interrupted while the update process is underway. After all updates are completed, you will be prompted to shut down the system to a full power-off state. A full power cycle is required for the newly loaded FPGA files to take effect.

ProbeConnectorStatus Typing this command at the Matlab command prompt will result in a display of the status of the transducer connector(s) on the system, including the connector type, whether a transducer is connected, and if so the ID and name of the transducer (if the ID is recognized by the system).

VerasonicsVerificationTest or "VVT" A comprehensive self-test of the entire system. See the "System Self-test.." heading within the "Final Setup and System Tests" section of this document, for instructions on running *VerasonicsVerificationTest*.

SystemInfo This utility provides a comprehensive status report of the configuration of your host computer, Vantage SW installation and SW license status, and Vantage HW system (if one is connected). Refer to the "Software Licensing" section of this document for more information on SW licensing and how to use the *SystemInfo* utility.

ArbWaveToolbox A collection of utilities and examples to help with defining and analyzing waveforms using the Vantage system's arbitrary waveform transmit features. Documentation on how to use these utilities is included within the *ArbWaveToolbox* subfolder of the *Tools* folder.

ElementToChannelMapping Utilities to help understand the actual mapping between transducer element numbers, system channel numbers, entries in the TX/Receive Apod arrays, and pin numbers at the transducer connector. See the "Element to Channel Mapping" heading of the "Guide for use of the UTA SystemConfiguration" section of this document for information on how to use these utilities.

EventAnalysisTool A very useful tool to assist with developing and debugging a SetUp script. After running a SetUp script, so its output is present in the Matlab workspace, just enter *EventAnalysisTool* at the Matlab command prompt and you will see a visual display of all events in the event sequence. Clicking on any Event will display the contents of structures indexed by that Event, making it easy to find and fix errors in indexing logic as well as helping you understand what the system will actually be doing as you step through each event in the event sequence.

RF_FiltersDesignTool A utility to assist with designing and reviewing the digital filters provided in the receive data path for each channel of the Vantage HW system. Documentation for this utility is located in its subfolder of the *Tools* folder.

RFDataViewer Provides plots and displays of RF data that has been acquired from the HW system and stored in the RcvData buffer in the Matlab workspace.

PTool Preset utility, for storing and recalling settings of the GUI controls associated with running a SetUp script.

showTXPD When a TX structure is present in the Matlab workspace, invoking *showTXPD* will produce and display beam plots of the transmit beam that will be produced by each individual TX structure.

RF Data Acquisition Time Tag

This is a user-programmable feature added in the 3.1 release. When enabled, the time tag feature will overwrite the first two samples of each receive data acquisition event with a 32 bit "time tag" value taken from a free-running counter in the hardware system. As a result, the RF data stored in memory from every acquisition event will include a time tag identifying the precise time at which it was acquired relative to other acquisition events with a resolution of 25 usec and a range greater than 24 hours. If desired you can also associate the relative time tag values with an absolute time and date taken from the host computer's time / date functions.

When a script starts running the time tag feature will be disabled by default. It can be enabled or disabled whenever desired by using the mex command "enableAcqTimeTagging" with the following syntax:

result = enableAcqTimeTagging(enable);

The argument "enable" must be a Matlab string, set to the value '0' to disable time tags or the value '1' to enable them. The returned value "result" will be a Matlab string variable, set to the value 'Success' if the command completed successfully, or some other string identifying the problem if there was an error.

The enable status is "global", in the sense that while time tags are enabled they will be added to every RF data acquisition event executed by the HW system, and they will be applied to every receive channel included in the DMA transfer to the Receive Buffer in the host computer. There is no mechanism for applying the time tag selectively to individual channels. The Recon function is not aware of the presence of time tags, so if the Recon processing makes access to the first two samples from a receive event, the time tag values will be used as if they were legitimate RF data. In most acquisition scripts this is not a problem since the first two samples will come from before or during the transmit activity, and thus the actual sample values are typically meaningless anyway. If it was necessary to prevent the time tags from corrupting the receive data samples you could either arrange the acquisition timing so the first two samples would not be used for RF processing, or you could use an external processing function to force them to zero after the time tag value had been read.

Enabling and disabling the time tag feature will not affect the state of the 32 bit time tag counter. It is reset to the zero state at system power-up and then free runs continuously as long as the system is powered up. Since it increments every 25 usec, the 32 bit value will wrap around back to zero after approximately 107,000 seconds or 29.8 hours. If desired you can also program the counter to wrap back to zero after exactly 24 hours or 86,400 seconds. You can also reset the counter back to zero whenever desired. Both of these options are controlled through a separate mex function, "setTimeTaggingAttributes", with the following syntax:

result = setTimeTaggingAttributes(wrap24, reset);

The argument "wrap24" must be a matlab string, set to either '1' or 'true' to enable the 24 hour wraparound, or '0' or 'false' to allow the full 32 bit count wraparound. The "reset" argument will force the counter value back to zero if it is set to '1' or 'true', at the point in time when the command is executed. The "reset" argument will do nothing if it is set to '0' or 'false'.

To map the time tag counter value to an absolute time and date, simply record the time and date from the host computer using an appropriate Matlab function, immediately after sending the "reset" command to the time tag counter. Then subsequent time tag values can be interpreted as an offset from that recorded time.

The example script "SetUpL11_4vFlashTImeTag" provides an example of enabling and disabling the time tag feature as well as extracting the time tag value in an external processing function and displaying it. The utility function "ShowTimeTag" (in the Tools folder) provides an example of extracting time tag values from the receive data buffer in the Matlab workspace.

Front Panel Status Indicators

There are four LED indicators visible on the front of the Vantage system chassis to provide information on the operating state of the hardware system. Numerous status signals are available to drive these indicators, and new SW features in the 3.1 release make it easier to select the desired signals from within a user's SetUp script.

A new field "Resource.Parameters.SystemLED' is used to specify the signal being used to drive each of the four indicators. The variable "SystemLED" must be defined as a 1 X 4 cell array of strings, where each of the four strings specifies the signal to be assigned to the associated indicator from left to right. The string values that can be used are listed below:

- **off** turns off the indicator
- **running** hardware event sequence is running

- **starting** LED will come on momentarily while hardware event sequence is starting to run
- **paused** sequence has paused for any of the individual reasons listed below
- **pausedOnTriggerIn** pause waiting for trigger
- **pausedDmaWaitPrevious** pause waiting for previous DMA to complete
- **pausedDmaWaitForProcessing** pause waiting for DMA processing complete signal
- **pausedOnMultiSysSync** pause waiting for trigger through the sync box
- **pausedSync** pause waiting at a Sync command
- **externalBoxFault** fault condition in external sync box
- **missedTimeToNextAcq** specified time interval was not achieved
- **missedTimeToNextEB** specified time interval was not achieved

If this field is not specified in a SetUp script the system will create it with default values of {'running', 'paused', 'activeTxAndOrRx', 'transferToHostComplete'}.

Probe Connector Illumination

Systems configured with the Cannon DL-260 or DL-360 probe connectors provide an LED located behind the hole in the connector for the ZIF latching mechanism. This LED is intended to make it easier to locate the connector when using the system in a darkened room. By default, the system will enable this LED at all connectors, while nothing is connected to them.

A new field "Resource.Parameters.ProbeConnectorLED" can be used to disable the connector LED's if you desire. This field must be set to a 1X4 array of Matlab doubles, with each value to set to 1 to enable the associated connector's LED or 0 to disable it. The four values represent the four possible connectors supported by the software; on a single-connector system the last 3 values are ignored or on a 2-connector system the last 2 values are ignored. If you do not specify this field in your script, VSX will create it and assign a default value of [1 1 1 1] to enable all connector illumination LED's. If you want to disable the LED's, simply specify

Resource.Parameters.ProbeConnectorLED = [0 0 0 0];

in your SetUp script (or any other combination of 1's and 0's as desired).

Note that this feature only applies to the two connector types mentioned above. The UTA modules UTA 408, UTA 408-GE, and UTA 160-DH/32 LEMO do not provide connector illumination LED's at all so the ProbeConnectorLED field has no effect for those modules.

TPC Profile 5 High Voltage Monitor

All Vantage system configurations and software releases have provided a built-in fault detection monitoring feature for the transmit HV supply when using imaging TPC provides 1 - 4. If the measured actual HV supply voltage (measured by a monitoring circuit that is independent of the control of the HV) differs from the programmed voltage by more than 5 Volts, a fault condition is reported by the hardware system and execution of the event sequence is terminated. The monitor comparison is done once per acquisition event, using the measured voltage immediately before the start of the event so the measured value will not be affected by droop or transients that occur during and immediately after the active transmit interval.

Starting with the 3.1 release, a similar monitoring function is now available for use with TPC Profile 5 (in all prior releases the monitoring feature was disabled for transmit events using Profile 5). The profile 5 monitor functionality is exactly the same as described above for profiles 1 - 4, with the exception that the fault detection thresholds are user programmable. A new mex function, "setProfile5VoltageLimits" provides control of this feature with the following syntax:

result = setProfile5VoltageLimits(P5HVmin, P5HVmax);

The arguments P5HVmin and P5HVmax are Matlab doubles that specify the lower and upper thresholds at which an error will be triggered, in units of Volts over the range from 0 to 100 Volts. The returned value "result" will be a Matlab string variable,

set to the value 'Success' if the command completed successfully, or some other string identifying the problem if there was an error. The profile 5 monitoring feature can be disabled by setting any pair of values such that P5HVmin is equal to or greater than P5HVmax. The profile 5 monitor function is also disabled by default, if you never invoke the setProfile5VoltageLimits command.

Note that you must manually update the monitor thresholds whenever you change the TPC Profile 5 HV setting (one approach would be to disable the monitor prior to changing HV, and then re-enable it with new limits after the voltage has actually settled at the new setting. An alternative would be to change the limits so they include both the old and new settings while the voltage is in transition, and then change again to new limits for the new setting after the transition is complete.)

The example script "SetUpL11_5vlashPushP5Mon" illustrates the use of this feature, with GUI controls on the min and max limits. This script is located in the folder *Examples_Biomedical / Specialty_Applications / RadiationForce_ShearWaveVisualization*.

8 Running the Verasonics System - A Summary

Background Overview

The Verasonics Software includes many example programs ("m-scripts") written in the Mathworks Matlab programming language, which specify, through a set of Matlab structures, all ultrasound parameters controlling the ultrasound acquisition as well as the data processing and image display.

By convention, Verasonics names the ultrasound parameter scripts "Setup_ProbeName_ScanType.m" (e.g. "SetUpL11_4vFlash"), and here such parameter initialization scripts will simply be referred to as "Setup" scripts.

When executed, these Setup scripts call several Verasonics Matlab helper functions and compiled C-language files, which Matlab calls ".mex" files. The Verasonics mex files use the Verasonics HAL to communicate with the Verasonics Hardware. The "Hardware Abstraction Layer" (HAL) is a low level software library that permits controlling the Verasonics Hardware with high level programming.

The Vantage project folder is the top level of the Vantage software hierarchy. Within this folder are m-scripts and mex files that the user sees in their Matlab file structure. The programs and functions use the Matlab environment to specify and run ultrasound imaging sequences on the Verasonics Hardware. The master Verasonics script is *VSX*, the "Verasonics Script eXecutor", which loads the user's specifications, sets up the graphics controls and display in Matlab, and calls the runtime program "runAcq" (a compiled mex file) that interacts with the HAL and the hardware. *VSX* usually runs in a continuous loop, until the user quits by closing the graphical user control panel.

The Vantage system should be placed in a location that is clean and well ventilated. Position the system so that it may be connected using the PCIe cable to the computer that you will use to control the system.

Launch Matlab and navigate to your Vantage project folder. Activate the Vantage project folder by issuing the *activate* command.

Execute one of the sample *Setup* scripts by navigating to the folder containing the desired script. Right-click on the script of interest and select the *Run* option. This will execute the script and create a similarly named *.mat* output file; these *.mat* files store the ultrasound parameters that were created by the *Setup* script.

Please note that *VSX* asks the user for the name of the *.mat* file that contains their desired settings.

The example *Setup* scripts are provided in the folders *Examples_Biomedical* and *Examples_NDE_NDT*. They should not be modified. Instead, we recommend that you begin development of a custom script by copying an example script into a new subfolder you have added and editing that copied file. This way, you will still have access to the unmodified original folder in its original location.

You run the Vantage software from within an activated Vantage project folder. When a Vantage project folder has been activated, it becomes the current working directory in Matlab. This is required in order for the Vantage software to operate properly.

You may add other directories to your Matlab path. They do not have to be located within your Vantage project folder. (Note however that directories outside the Vantage project folder will be removed from the path by the activate command. Any such directories must be added to the path after running activate.)

Starting an Imaging Script

At the Matlab prompt, enter *VSX*, which is case sensitive. *VSX* will then ask you for the name of the ".mat" file you want to use. For example, enter "L11-4vFlash" (do not include the ".mat" extension) to use the .mat file created by the SetUpL11_4vFlash.m" example script. After a brief pause, a graphical user interface ("GUI") control window and an image display window appear. If you have the Verasonics Hardware and Verasonics Software correctly connected, powered, and installed, and with an L11-4v probe connected, the image window will display live imaging data. If you do not have an L11-4v transducer, you can still run the script with no transducer connected to verify that the system is functioning (in this case don't expect to see anything other than some low-level thermal noise in the image display).

To build a script, use the Matlab "Current Folder" hierarchical viewer to find the script you want, such as "ExampleScripts/L11-4v/SetUpL11_4vFlash.m". Right click the file and choose "Run" from the contextual menu. This will create "L11-4vFlash.mat" and save it in the MatFiles sub-folder. You can then run *VSX* (case-sensitive) and type "L11-4vFlash" to load and run that script.

If the Verasonics Hardware is not detected by the HAL, the system drops into a simulation mode in which simulated backscattered data is generated using the script settings, and processed and displayed in real time at frames rates that are similar to those for live data.

In simulation mode most GUI processing parameter controls work as before, and this mode permits post-processing previously recorded data. The Vantage software can be used without the Vantage hardware unit attached, or with the Hardware attached but powered off. You can develop scripts in simulation mode and then attach the Vantage hardware unit to run the same script in live imaging mode.

In the example above, the "L11-4vFlash" file that was provided as input to *VSX* is a ".mat" file, which is a Matlab binary data file. The "SetUpL11_4vFlash.m" Matlab script, in the "ExampleScripts/L11-4v" sub folder, is the m-script file that generated the "L11-4vFlash.mat" output file. There are many other example scripts illustrating different acquisition geometries and sequences, for B-mode, Color Flow Doppler, Spectral Doppler, unfocused and focused beams, and other acquisition and processing approaches to help you learn how to write your own customized scripts. Note that an example script will not run if the probe plugged into the system does not match the probe specified in the script, but you can elect to run a script with no probe connected at all. There are exceptions and workarounds: see the "SequenceProgramming" user guide.

If you attempt to run an example script with a different transducer connected to the system, an error message will result and the script will not run. If no transducer is connected at all, you can elect to allow the script to run, for example, to verify that the script and system are functioning, or to examine the transmit output with an oscilloscope or for other test/ debug purposes.

Alternative Methods of Running a Script with VSX

As explained in the previous section, the most common way to run a script with VSX is to simply enter *VSX* at the Matlab command prompt. This will start VSX, and it will then prompt you for the name of the .mat file you want it to load and run. This must be the exact name of a .mat file that was previously created by running a SetUp script. There are several alternative ways you can use to run a script with VSX, however. You may use any of the alternatives listed below, depending on what you find most convenient:

Using SetUp Script name When invoking *VSX* from the Matlab command prompt as described above and then responding to the query from VSX for the name of the file to use, you can also enter the name of the SetUp script that was used to create desired .mat file. If you have entered a SetUp script name, VSX will open that file and search within it to find the name of the .mat file the script creates, and then use that .mat file if it exists (note you still must have run the SetUp script itself at some point in the past, to create the .mat file). This approach saves you the trouble of having to remember the exact spelling of the .mat file name.

Running VSX automatically from within the SetUp script A third approach is to add two lines of code within the SetUp script, so that when you run the script it will automatically run VSX to load and execute the resulting .mat file. This approach eliminates the need to manually run VSX, and also eliminates the need to respond to the query from VSX for the file to be run. To do this, insert the following two lines of code in the SetUp script immediately after the line where the script saves the .mat file with the "save" command. For example suppose I have a SetUp script that creates a file named "myScript.mat". This script would have a save command that looked like this:

```
save('MatFiles/myScript');
```

To automatically run VSX from within the SetUp script, I would add the following two lines immediately after the line with the save command:

```
filename = 'myScript';
VSX
```

When VSX is invoked, it first searches for a variable named *filename* in the Matlab workspace. If this variable is found, VSX will automatically load and run the file identified by *filename* instead of prompting for you to enter it.

Some of the example scripts provided by Verasonics already include those two lines, but have them commented out. In that case all you have to do to enable the automatic run of VSX is to uncomment the two lines and then save and run the script.

Creating Custom Imaging Scripts

The Verasonics Vantage system is a highly flexible and capable instrument for ultrasound research. You can create your own custom imaging scripts to instruct the Vantage hardware unit and Vantage software to perform a variety of acquisition sequences. Typically, custom processing algorithms are used with a custom acquisition sequence to produce novel ultrasound displays.

To learn how to write sequence scripts, and how to include custom processing functions, first see the *Vantage Sequence Programming Tutorial* document. You will also need to refer to the *Vantage Sequence Programming Manual* which describes all of the Application Programming Interfaces (APIs) and parameter definitions available to the user.

To protect your custom files, you should back up your Vantage project folders on a regular basis.

Debugging Scripts

The first step in debugging is to examine the contents of the structures created by the Setup script. These are variables in the Matlab workspace, and are examined by double clicking the variable name in the workspace window. In some cases, *VSX* will have created additional structures and modified others. In addition to normal Matlab debugging techniques, and those discussed in the Tutorial, see the *Vs_MATLAB and HAL Debugging Support* section in the Verasonics *HAL C User Guide* for some tips for debugging interactions of the Verasonics HAL with Verasonics Matlab scripts. Additionally, the *Version* Verasonics Matlab script can be useful to quickly test access to Verasonics HAL and VDAS hardware. The *vdasReset.m* (case-sensitive) Verasonics script can be used to correct Verasonics hardware/ software state issues, instead of resorting to a computer reboot. (If *vdasReset.m* fails, then it is suggested that you power off the computer and Verasonics VDAS to reset the VDAS hardware.)

Connecting Custom Transducers

It is possible to use custom transducers and arrays with the Verasonics System. See the "Vantage Sequence Programming Manual" user guide and tutorial documents, and the "Application Note on Verasonics ZIF Transducer Connector" document (available through Verasonics Customer Support), for more information about connector pinout, transducer identification schemes, and other features provided through the transducer connector interface (HV mix control, temperature sensors, etc.).

9 User Guidelines

Managing Multiple Vantage Software Installations

Starting with the Vantage 2.11 release up through the current 4.0.1-1903121200 release, you can install any number of copies of the Vantage project folder and locate them wherever you wish on your host computer. To create a duplicate installation you can either run the installer again, or simply make a copy of an existing Vantage project folder. You can chose any name you like for renaming the Vantage project folder directories- there is nothing in the Vantage SW itself that is keyed to the directory name. We would advise, however, to keep the 3-digit SW revision ID and build date code as part of the name, to avoid potential confusion if you have multiple installations at different release levels.

As explained in the "Final Setup" section of this document, when you want to switch from one Vantage project folder to another you should close Matlab, then restart it and select the new Vantage project folder as the Matlab working directory and then run activate. It is not necessary to restart the Host Computer or cycle power. Note that it is equally easy to switch from one version of Matlab to another, if you have multiple versions of Matlab at different release levels installed on your host computer. The procedure is the same: close Matlab, then start the new version of Matlab and then select the desired Vantage project folder as the Matlab working directory and then run activate.

When operating with a Vantage HW system connected, if you switch from one Vantage SW release to another and the two releases require different levels of FPGA code for the HW system (this will be the case when switching between any of 3.0, 3.1, or 3.2), you will be prompted to allow updates to the HW system and a system power cycle will then be required before using the system. This does not apply if you are running the SW in simulate-only mode, with the HW system either disconnected or turned off.

Adding other directories to the Matlab Path

Note that when using current Vantage SW releases, the desired Vantage project folder MUST be selected as the Matlab working directory. If you select a higher-level folder that contains the Vantage project folder, or if you select a sub-folder within the Vantage project folder, the Vantage SW will not function properly and may cause a Matlab crash or other problems. This is because some of the Vantage SW functions use relative path definitions, tied to the Vantage project folder (this constraint is a key element of the design approach to allow multiple SW installations on the host computer).

You can easily add new subdirectories of your own within a Vantage project folder, however. Since the activate function automatically adds all subfolders within the Vantage project folder to the Matlab path, any folders you have added will also be included.

If, however, you would like to add directories to the Matlab path that are not included within the selected Vantage project folder, you can easily do this as well using any of the Matlab utilities for adding directories to the path. The only constraint is that you must do this AFTER you have run activate for the current Vantage project folder. This is because the activate function will first remove everything from the Matlab path (other than Matlab's own directories), and then add to the path all directories within the newly selected Vantage project folder.

If you want to avoid having to manually add your other directories to the path every time you start Matlab and run activate, you can preferences feature included in current Matlab releases to specify files that will be included in the default Matlab path when Matlab starts up. Folders added in this way will not be removed by the activate function. If you use this approach, make sure the files you are adding do not have names that will shadow files of the same name within the Vantage project folder.

Overview of the folder Structure within the Vantage project folder

Starting with the Vantage 3.0 release, several subfolders were added to make it easier to find the files you are interested in and reduce the clutter in the top level Vantage project folder. Listed below are descriptions of these sub-folders and their intended purpose and content. In 4.0, the :

Utilities This folder contains many of the Matlab "helper" functions that might be used within a user's SetUp scriptdirectory, such as computeTXDelays, computeTrans, computeTGCWaveform, etc. These are all ".m" files in the customer release, that a user may want to have access to in order to understand system operation or in some cases even modify (for example, you may want to edit computeTrans to add support for a custom transducer you have developed).

System This folder contains Matlab ".p" files and compiled C and mex functions for which source code is not included in the customer release. These files must be present for the system to run, but since most users have no need to ever look at them we created this sub-folder to get them out of the way in the main folder.

Example Scripts In current Vantage software releases, the Example Scripts have been organized in a multi-level hierarchy of folders to make it easier for you to find the probes and associated scripts you are interested in. At the root level of the Vantage project folder there are two main example script folders, *Examples_NDE_NDT* and *Examples_Biomedical* to divide the scripts into these two broad categories of applications supported by the Vantage product line. At the next level down in *Examples_Biomedical*, there is a separate subfolder for each Vantage hardware system configuration: Vantage 32LE, 64, 64LE, and 128-256. Within each of these systems folders you

will find a subfolder for each UTA module type supported for that system, and within each UTA folder are folders for each probe supported by that UTA, and within each probe folder you will find the individual example scripts for that probe. A fifth subfolder named *Specialty_Applications* will also be found under *Examples_Biomedical*. This folder contains folders for a broad range of both specific imaging applications (such as *PhotoAcoustics*) and also examples of how to use specific features of the system (such as *MultiSystemCLockSynchronization*). In most of these specialty folders you will find application notes and other documentation as well as the associated example scripts. The organization within *Examples_NDE_NDT* is similar to *Examples_Biomedical* but somewhat simpler because of the smaller range of UTA modules and example scripts currently available.

MatFiles The .mat file generated when you run a SetUp script is now stored in this folder rather than in the main folder, again with the sole purpose of reducing clutter in the main folder. If you have your own SetUp scripts that you have developed, or created by modifying example scripts from earlier releases, you should modify them to store their results in the MatFiles subfolder. All that is required is to change the line with the *save* command to add the MatFiles path. For example suppose I had a SetUp script that created a file named "myScript.mat". This script would have had a save command that looked like this:

save('myScript');

To use the MatFIles subfolder, the command must be changed to:

save('MatFiles/myScript');

Note that no change is required to any function that reads or loads a .mat file, since the MatFiles folder will always be added to the Matlab path by activate.

A note of caution with regard to the "MatFiles" folder: You should take care that duplicate .mat files are not accidentally created in multiple subdirectories or in the main Vantage project folder. If this happens, Matlab will use the first one it finds on its search path, which may not be the one intended. For example, if there was a copy of "myScript.mat" in the main Vantage project folder but I then modified the script, and had the new mat file output stored in the MatFiles folder, VSX would use the old version in the Vantage project folder since that would be the first one found in the Matlab search path. The best way to avoid this dilemma is to delete any .mat files inadvertently created in the root level of the Vantage project folder.

Other Reminders

- If software installation is necessary, be sure to follow software installation instructions carefully and explicitly, especially with respect to power cycling (cold reboot) to ensure proper software configuration when FPGA updates have been made.
- Operate the system only at ambient temperatures between 50 to 95 ºF (10 to 35 ºC).
- Make sure the AC line power provided to the system is within the limits in the system specification for Voltage, frequency, available power level, and levels of surges, sags, or dropouts.
- Do not obstruct the airflow vents which protect the system from overheating. If for any reason the cooling fans do not operate when the system and computer are on, please shut the system down and contact Verasonics Support immediately.
- System Self-Test: If at any time there is a question about whether the system is still functioning properly, shut it and the host computer down completely. Upon restarting the computer and launching MATLAB, disconnect all transducers and initiate the Verasonics self-test script "VerasonicsVerificationTest", and confirm that the system passes without any errors. If errors or failure notices appear in the command window, please contact support@verasonics.com, and include the test output.
- If the user intends to calibrate the acoustic output of the system, VerasonicsVerificationTest should be run before data are taken to confirm that the system is performing within specifications. Acoustic output measurements or any other verification tests done on the system should be repeated after every SW or HW modification, upgrade, or repair.
- Keep the shipping carton and packing materials; they have been specially designed to protect the system should you ever need to ship it back to Verasonics or to another location. The original packaging should always be used to transport the system by vehicle.

10 Guide for use of the UTA System Configuration

UTA ("Universal Transducer Adapter") is the currently available configuration of the Vantage HW system that provides user-replaceable modules on the front of the system, to allow use with a variety of different transducer connector types and pinouts. This makes the Vantage system compatible with different families of commercially available ultrasound probes as well as Verasonics-provided probes or custom probes developed by a Vantage customer. By purchasing multiple UTA adapter modules, a single Vantage system can easily be reconfigured for use with more than one probe/ connector family.

The UTA configuration was introduced in September 2015; Vantage systems built prior to that date do not have the UTA feature and instead use a "captive SHI" on the front of the system that only provides the HDI-format 260 pin connector (equivalent to UTA 260-S or UTA 260-D).

All Vantage SW releases since 2.11 provide full support for the new features of the Vantage UTA HW configuration, while also maintaining full compatibility with the earlier pre-UTA "captive SHI" systems.

Adapter Modules Supported by the System

The Vantage 4.0.1-1903121200 software release supports the UTA adapter types listed below for use with a UTA-compatible HW system. The description of each adapter includes a summary of the system configurations that can be used with it.

UTA 260-S (HDI-format 128 channel single connector)

This adapter is designed primarily for use with Vantage-128 and Vantage-64LE system configurations. All example scripts for use with the Vantage 64 LE system and this adapter have the suffix "_64LE" added to the name of the SetUp script, to differentiate them from the full 128-channel scripts for the same probe. In addition to the 128 transducer element signals, this adapter supports the following additional features through the connector:

- Serial EEPROM interface for reading probe ID.
- Power and control signals for HV Mux switching, for probes that require that capability.
- DC element bias power supply output, for use with probes that require a DC bias voltage applied to the elements (this uses one of the same power supply outputs that is also used for HV Mux switching, and thus the DC element bias feature cannot be used with HV Mux probes).
- Two analog signal inputs that can be used for probe temperature sensing from thermistors, or other analog input signals.

Scripts for use with this adapter must define a Trans.connType value of 1.

Starting with the 3.4.0 release, the UTA 260-S can also be used with a Vantage 32 LE or Vantage 64 system, but only with 64-channel scripts that use connector element signals 1:64 and do not use element signals 65:128.

The UTA 260-S can also be used on a Vantage-256 UTA system; in that case the system will automatically configure itself to function as a Vantage 128.

An older pre-UTA Vantage 128 or 64 LE system with the single-connector captive SHI module will be functionally identical to the corresponding Vantage UTA system with the UTA 260-S adapter.

UTA 260-D (HDI-format 128 channel dual connectors)

This adapter is designed primarily for use with Vantage 256 systems, providing two 128-channel connectors for two separate probes, or for a 256-element probe using both connectors. The features available through either connector of the UTA 260-D are identical to those described above for the 260-S, and thus any probe and user script that works with a UTA 260-S can also be used with the UTA 260-D on a Vantage 256 system.

- Serial EEPROM interface for reading probe ID is provided independently for both connectors.
- HVMux switching, DC element bias, and analog signal inputs are provided in the exact same fashion as the UTA 260-S and can be used from either connector when it is used by itself. Note however that when both connectors are selected at the same time, these features will only be usable through the 'primary' connector, which is always connector 1 (left) with the current Vantage software.

Scripts for use with this adapter must define a Trans.connType value of 1.

Starting with the 4.0.0 release, the UTA 260-D can be also used with Vantage 64 LE an Vantage 128 systems, but use is restricted to element signals 1:64 on connector 1 and 65:128 on connector 2. The element signals from channels 65:128 on connector 1 and 1:64 on connector 2 can not be used at all; a script attempting to use any of these element signals will trigger an error condition. If connector 1 is selected by itself, the system will function as a Vantage 64, with the 64 system channels mapped to connector 1 element signals 1:64. If connector 2 is selected, connector 1 will also automatically be selected. System channels 1:64 will be mapped to connector 1 element signals 1:64, and system channels 65:128 will be mapped to connector 2 element signals 65:128.

The UTA 260-D cannot be used with Vantage 32 LE or Vantage 64 systems; attempting to do so will result in an error condition preventing use of the system.

An older pre-UTA Vantage 256 system with the dual-connector captive SHI module will be functionally identical to a Vantage 256 UTA system with the UTA 260-D adapter.

UTA 260-MUX (HDI-format 128 channel single connector)

This adapter provides the same probe connector interface functionality as the UTA 260-S, but adds 2:1 HV Mux switching within the adapter to allow the 64-channel Vantage 32 LE and Vantage 64 systems to have access to all 128 channels of any probe using a Trans.connType value of 1. Note however that HV Mux probes and probes requiring DC element bias cannot be used with this adapter, since the system's HV Mux control features are being used for the adapter itself and are not available through the connector.

All example scripts for use with the Vantage 64 system and this adapter have the suffix "_64" added to the name of the SetUp script, to differentiate them from the full 128-channel scripts for the same probe. Similarly, example scripts for the Vantage 32 LE will have the suffix "_32LE". Note that a script for use with the UTA 260-MUX adapter must be written to only use 64-element apertures in each acquisition event, with the HV Mux 'aperture' field added to each TX and Receive structure. See the Example Scripts folders for a full list of probes currently supported with example scripts for the UTA 260-Mux adapter module.

While intended primarily for use on the 64-channel Vantage 32 LE and Vantage 64 systems, the UTA 260-MUX adapter can also be used on Vantage 64 LE, Vantage 128, or Vantage 256 systems. In all of those cases the system will automatically configure itself as the functional equivalent of a Vantage 64 system, and provide the same functionality as the Vantage 64.

Scripts for use with this adapter must define a Trans.connType value of 1.

UTA 360 (MS-format 256 channel single connector)

This adapter is designed exclusively for use with Vantage 256 systems. This module has a Cannon DL-360 connector with 256 element signals and matches the pinout of the MS family of high frequency probes.

This connector interface supports the ability to ready an ID code from the probe. It does not provide any support for probe HV Mux control, element bias, or temperature sensing.

Scripts for use with this adapter must define a Trans.connType value of 3.

This adapter can not be used on Vantage 32 LE, Vantage 64, Vantage 64 LE, or Vantage 128 systems; attempting to do so will result in an error condition preventing use of the system.

UTA 408 (Verasonics 408 pin 256 channel single connector)

This adapter uses an ITT Cannon DLP408R 408 pin ZIF connector providing 256 element signals with a connector interface and pinout developed by Verasonics, for use only with probes provided by Verasonics (or custom probes designed by a customer to be compatible with this interface). The UTA 408 is designed primarily for use with the Vantage 256 system, but for a probe and script that only uses element signals in the range 1:128 it can also be used on the Vantage 128 system.

The UTA 408 does not support probes with HV Mux switches, but does include support for reading a probe ID serial eprom, providing a DC element bias power supply, and two probe temperature sensing thermistor analog input signals.

Scripts for use with this adapter must define a Trans.connType value of 4.

This adapter can not be used on Vantage 32 LE, Vantage 64, or Vantage 64 LE systems; attempting to do so will result in an error condition preventing use of the system.

UTA 408-GE (GE format 408 pin 256 channel single connector)

This adapter is specifically designed for use with the GE "D" series of ultrasound probes. See the example scripts folder for a list of GE probes and example scripts currently provided with the Vantage software. These GE probes can be purchased through Verasonics.

This adapter includes support for reading probe ID from the probe's serial eeprom as defined by GE, HV Mux power and control, and the ability to read one temperature sense thermistor for probes that include this feature. The connector interface for these features is proprietary to GE, and Verasonics cannot provide information on these interfaces to customers.

The UTA 408-GE provides support for 256 element signals, for use with Vantage 256 systems. Probes using 128 or fewer element signals (including HVMux-based probes with more than 128 elements) can also be used on Vantage 128 systems.

Scripts for use with this adapter must define a Trans.connType value of 7.

This adapter can not be used on Vantage 32 LE, Vantage 64, or Vantage 64 LE systems; attempting to do so will result in an error condition preventing use of the system.

UTA 160-DH/32 LEMO (two Hypertac 160 pin 128 channel connectors and 32 Lemo coax)

This adapter module provides two Hypertac 160 pin connectors with 128 element signals per connector using a pinout that is compatible with many commercially available ultrasound probes designed for NDT applications. The adapter also has 32 Lemo coax connectors for use with single-element transducers.

The Hypertac connector interface provides no mechanism for probe disconnect sensing, no probe ID, and no other features beyond the 128 element signals.

Scripts for use with this adapter must define a Trans.connType value of 12 (an earlier version of this adapter used a connType of 6, which is still supported by the system).

The UTA 160-DH/32 Lemo adapter can be used with any Vantage system configured with the UTA capability: Vantage 32 LE, Vantage 64, Vantage 64 LE, Vantage 128, and Vantage 256 Systems with any of the frequency ranges or other configuration options available on each system. Note however that while the Hypertac connectors provide support for transducers with up to 128 elements, not all of those elements can be used on all systems:

- Vantage 32 LE and Vantage 64 systems can only use elements 1:32 at each of the two Hypertac connectors.
- Vantage 64 LE and Vantage 128 systems can only use elements 1:64 at each connector.
- A Vantage 256 system can use all 128 elements at each connector.

On all Vantage systems you can select either one of the two Hypertac connectors by itself, or both of them together (with each connector routed to a separate set of channels in the system). Regardless of which combination of connectors is being used, the channel count available at each connector will be restricted as described above. In addition to the two Hypertac array connectors, the adapter provides 32 individual Lemo coax connectors for use with single-element transducers. These 32 connectors are accessible on all Vantage system configurations from Vantage 64 through Vantage 256, but they share the same 32 system channels that also support the first 32 element signals at the top Hypertac connector. Relays provided within the adapter can be programmed to select the Hypertac connector, or the array of Lemo connectors, or both the Hypertac and Lemo together (system channel signals connected in parallel to both a Lemo connector and the corresponding pin of the top Hypertac connector).

For more information on the use of this adapter module, refer to the document "AppNote_Adapter Module UTA 160-DH 32 LEMO", in the ExampleScripts / NDT scripts subfolder of the Vantage project folder.

UTA 160-SH/8 LEMO (one Hypertac 160 pin 128 channel connector plus 8 Lemo coax)

This adapter module provides the same Hypertac 160 pin connector interface as the UTA 160-DH/32 LEMO described above, with 128 element signals using a pinout that is compatible with many commercially available ultrasound probes designed for NDT applications. The UTA 160-SH/8 LEMO provides a single Hypertac connector and also has 8 Lemo coax connectors for use with single-element transducers.

The Hypertac connector interface provides no mechanism for probe disconnect sensing, no probe ID, and no other features beyond the 128 element signals.

Scripts for use with this adapter must define a Trans.connType value of 12.

The UTA 160-SH/8 Lemo adapter can be used with any Vantage system configured with the UTA capability: Vantage 32 LE, Vantage 64, Vantage 64 LE, Vantage 128, and Vantage 256 Systems with any of the frequency ranges or other configuration options available on each system. Note however that while the Hypertac connector provides support for transducers with up to 128 elements, not all of those elements can be used on all systems:

- Vantage 32 LE and Vantage 64 systems can only use elements 1:64 at the Hypertac connector.
- Vantage 64 LE, 128, and 256 systems can use all 128 elements at the connector.

In addition to the Hypertac array connector, the adapter provides 8 individual Lemo coax connectors for use with single-element transducers. These 8 connectors share some of the element signals used at the Hypertac connector. Relays provided within the adapter can be programmed to switch a pair of element signals from the Hypertac connector to a pair of Lemo connectors: element signals 1 and 128 at the Hypertac connector can be switched to the first two Lemo connectors, element signals 2 and 127 to the second pair of Lemo connector, and so on for all four pairs. There is no provision for selecting both the Hypertac element signal and the Lemo connector in parallel.

UTA 128 LEMO (128 single-element Lemo coax connectors)

This adapter provides an array of 128 individual Lemo single-element coax connectors, directly wired to the 128 channels of a Vantage 64 LE or Vantage 128 system.

- The UTA 128 LEMO can also be used on Vantage 32 LE and Vantage 64 systems, but in this case only 64 of the 128 Lemo connectors will be usable.
- The UTA 128 LEMO and also be used on a Vantage 256 system. In this case the Vantage 256 will configure itself to function like a Vantage 128.

Scripts for use with this adapter must define a Trans.connType value of 11.

UTA 64 LEMO (64 single-element Lemo coax connectors)

This adapter provides an array of 64 individual Lemo single-element coax connectors, directly wired to the 64 channels of a Vantage 32 LE or Vantage 64 system.

- The UTA 64 LEMO can also be used on Vantage 64 LE, Vantage 128, and Vantage 256 systems. In these cases the system will configure itself to function like a Vantage 64.

Scripts for use with this adapter must define a Trans.connType value of 11.

UTA 160-SI/8 LEMO (one Ipex 160 pin 128 channel plus 8 single-element Lemo coax)

This adapter module uses the Ipex 160 pin connector interface using a pinout that is compatible with commercially available ultrasound probes designed for NDT applications. The UTA 160-SI/8 LEMO provides a single Ipex connector and also has 8 Lemo coax connectors for use with single-element transducers.

The Ipex connector interface provides no mechanism for probe disconnect sensing, no probe ID, and no other features beyond the 128 element signals.

Scripts for use with this adapter must define a Trans.connType value of 10.

The UTA 160-SI/8 Lemo adapter can be used with any Vantage system configured with the UTA capability: Vantage 32 LE, Vantage 64, Vantage 64 LE, Vantage 128, and Vantage 256 Systems with any of the frequency ranges or other configuration options available on each system. Note however that while the Ipex connector provides support for transducers with up to 128 elements, not all of those elements can be used on all systems:

- Vantage 32 LE and Vantage 64 systems can only use elements 1:64 at the Ipex connector.
- Vantage 64 LE, 128, and 256 systems can use all 128 elements at the connector.

In addition to the Ipex array connector, the adapter provides 8 individual Lemo coax connectors for use with single-element transducers. These 8 connectors share some of the element signals used at the Ipex connector. Relays provided within the adapter can be programmed to switch a pair of element signals from the Ipex connector to a pair of Lemo connectors: element signals 1 and 128 at the Ipex connector can be switched to the first two Lemo connectors, element signals 2 and 127 to the second pair of Lemo connector, and so on for all four pairs. There is no provision for selecting both the Ipex element signal and the Lemo connector in parallel.

UTA 156-U (Ultrasonix format 156 pin 128 channel connector)

This module has a Cannon DL-156 connector with 128 element signals and matches the pinout of probes for the Ultrasonix system. It is intended for use with the Vantage 64 LE, Vantage 128, and Vantage 256 systems (when used on a Vantage 256, the system will configure itself to function like a Vantage 128 since only 128 element signals are available).

This connector interface supports the ability to ready an ID code from the probe. It does not provide any support for probe HV Mux control, element bias, or temperature sensing.

Scripts for use with this adapter must define a Trans.connType value of 9.

This adapter can not be used on Vantage 32 LE or Vantage 64 systems; attempting to do so will result in an error condition preventing use of the system.

UTA 256-Direct (256 element probe cable direct-connect interface)

This module is designed to support direct connection of a transducer cable to the adapter module, supporting up to 256 element signals. Instead of a probe connector, this module provides for direct connection of the probe cable assembly to connectors on the UTA module PCB, with an overall shield cover and strain relief attachments for the cable.

This adapter provides no support for probe ID, temperature sensing, or any other features within the probe itself beyond the 256 element signals.

Scripts for use with this adapter must define a Trans.connType value of 8.

The UTA 256-Direct can be used on all Vantage system configurations, but the number of element signals that can be used will depend on the system:

- With a Vantage 256, all 256 element signals can be used.
- With a Vantage 64 LE or Vantage 128, only the first 128 of the 256 element signals can be used.
- With a Vantage 32 LE or Vantage 64, only the first 64 of the 256 element signals can be used.

UTA 1024-Mux (1024 element probe cable direct-connect interface)

This module is designed to support 1024 element matrix array transducers (such as those that are available from Vermon; the example scripts provided with the system are for a Vermon 1024 element probe). Instead of a probe connector, this module provides for direct connection of the probe cable assemblies to connectors on the UTA module PCB, with an overall shield cover and strain relief attachments for the cables. The UTA module provides a 256 to 1024 array of HVMux switches, controlled through the user script similar to any other HVMux probe, to allow use with a Vantage 256 system.

This adapter provides no support for probe ID, temperature sensing, or any other features within the probe itself beyond the 1024 element signals.

Scripts for use with this adapter must define a Trans.connType value of 8.

This UTA module can not be used with Vantage 32 LE, 64, 64 LE, or 128 systems; attempting to do so will result in an error condition preventing use of the system.

UTA-related Terminology and Definitions

To make sense of Verasonics documentation and user instructions, we have defined new terminology and HW module names for the components of a UTA system, while remaining consistent with the terminology that was used in pre-UTA systems. In some contexts these names can be ambiguous and confusing; the definitions given below should help you resolve that confusion.

UTA This is an acronym for "Universal Transducer Adapter". This is our name for the overall system HW configuration with the UTA feature installed, such as "Vantage 256 UTA", while the earlier or pre-UTA HW configuration would simply be identified as "Vantage 256". The physical configuration of these two systems is as follows: A Vantage 256 system will have the SHI module installed at the front of the system chassis, while a Vantage 256 UTA system will have the UTA baseboard module installed in place of the SHI and then a UTA adapter module will be connected to the front of the UTA baseboard.

SHI module This is the physical HW module installed at the front of a pre-UTA system. SHI is an acronym for "ScanHead Interface". The SHI module includes the HDI-format probe connector accessible at the front of the system (or two HDI-format connectors on a Vantage 256 system). A Vantage 128 pre-UTA system is functionally identical to a Vantage 128 UTA system with the UTA 260-S adapter installed. Similarly, a Vantage 256 pre-UTA system is functionally identical to a Vantage 256 UTA system with the UTA 260-D adapter installed.

UTA Baseboard This is the functional equivalent of the SHI module for UTA systems. Instead of providing probe connectors, the UTA Baseboard provides the mechanical and electrical interface to support a UTA Module.

UTA Module This is the user-replaceable module that attaches to the front of a UTA system, and provides the physical probe connector(s).

Removing and Replacing UTA Adapters

The UTA adapters are intended to be very easily and quickly removed and replaced without the use of tools, but the following guidelines should be followed to ensure reliable performance and prevent damage to the system:

Turn off system power before removing or replacing a UTA adapter! If system power is present while an adapter is only partially connected, damage to the HW system could result. In addition, full system re-initialization is required for the system software to interact correctly with the newly installed UTA adapter. To help enforce this requirement, the HW system will go into a fault condition if it senses a UTA adapter being removed or replaced while power is on, or if the system is powered up with no adapter. This fault state is clearly identified by all of the LED's on the front of the system flashing continuously. To clear this fault condition you must shut the system down to a full power-off state, then finish installing the desired UTA adapter, and then restart the system.

To remove a UTA adapter First shut down the system to a power-off state, and then loosen all four thumbscrews on the front of the adapter. Then gently pull on the adapter handles simultaneously from both sides, while striving to keep the face of the adapter parallel to the front of the system. To prevent damage to the small and fragile connectors on the rear of the adapter, place it in a suitable container or flat on a table so that nothing comes in contact with the connectors.

To install a UTA adapter make sure system power is still off. Pick up the adapter from both side handles, and make sure you do not have it upside down- the Verasonics label printed on the adapter should be at the top (there is also mechanical keying in the corners, that will prevent an adapter from being fully installed if it is upside down). Align the adapter as closely as possible with the opening on the front of the system and parallel with the face of the system and gently push it into place until you can engage all four thumbscrews. Tighten all four screws until you feel some resistance, and then continue to tighten a bit at a time on each screw to preserve the parallel orientation. Continue until you can see on all edges that the adapter is fully seated onto the system- the gap between the adapter and the face of the system should be very small. Do not tighten the thumbscrews any more than you can easily do with your fingers.

Do not use the adapter handles to lift or move the system! This could damage the adapter. The handles are only to be used for removing and replacing adapters. To lift or move the system always grasp it from the bottom.

Writing User Setup Scripts for use with the UTA System

All requirements and guidelines for writing SetUp scripts for use with a UTA system configuration are identical to the pre-UTA configuration, with one exception: the script must identify the connector type that is required, by specifying a new field in the Trans structure- *Trans.connType*. In all earlier Vantage SW releases for use on pre-UTA hardware systems there was no need to identify the connector type since only one connector type existed that was identical on all system configurations.

When a SetUp script is run with the 4.0.1-1903121200 SW and a hardware system is present, VSX will determine which connector type is present on the hardware system based on the type of UTA adapter module that is installed. If the system connector type does not match the connector type required by the script, the SW will exit with an error condition (or can be directed to revert to simulation-only operation).

Trans.connType Definition

This is a new field that has been added to the Trans structure to support the UTA feature. *Trans.connType* is a required field for the Vantage system SW; it is a Matlab double set to an integer value used as an index to identify one of the connector types supported by the UTA system. The following connector type index values are recognized by the 4.0.1-1903121200 SW:

Trans.connType = 1 identifies the HDI-format connector, identical to the probe interface used on pre-UTA Vantage systems. For backward compatibility to scripts developed with earlier SW releases, VSX will create Trans.connType with a default value of 1 if it was not defined in the user SetUp script (but only if the value of Resource.Parameters.numTransmit matches the requirements of the HDI connector, i.e. either 128, or 256 with Resource.Parameters.Connector = [1 2]). This connector type is used on the UTA 260-S, UTA 260-D, and UTA 260-MUX adapter modules.

Trans.connType = 2 identifies the interface from either a “break-out board” UTA adapter (providing direct access to the element signals, with no specific probe connector at all), or a custom UTA adapter built by a user.

Trans.connType = 3 identifies the MS family connector interface and pinout (Cannon DL-360 connector with 256 element signals). This connector type is used on the UTA-360 adapter module.

Trans.connType = 4 identifies a connector interface developed by Verasonics (256 element signals using an ITT Cannon DLP408R 408 pin ZIF connector). This connector type is used on the UTA 408 adapter module.

Trans.connType = 6 identifies the Hypertac 160 pin, 128 channel connector used by commercially available probes for NDT applications. This connector type is used on early versions of the UTA 160-DH/32 LEMO adapter module where disconnect sensing is provided for the connector.

Trans.connType = 7 identifies the GE "D" series probe family connector interface. This connector type is used on the UTA 408-GE adapter module.

Trans.connType = 8 identifies the "direct connect" interface to the probe, designed to mate directly with cable termination "paddle boards" at the end of the probe cable assembly. This connector type is used on the UTA 256-Direct and UTA 1024-MUX adapter modules.

Trans.connType = 9 identifies the Ultrasonix format 156 pin, 128 channel connector This connector type is used on the UTA 156-U adapter module.

Trans.connType = 10 identifies the Ipex 160 pin, 128 channel connector This connector type is used on the UTA 160-SI/8 LEMO adapter module.

Trans.connType = 11 identifies the single-element Lemo coax connector This connector type is used on the UTA 64-LEMO and UTA 128-LEMO adapter modules, and also on the other NDE/NDT adapter modules using the Hypertac or Ipex connectors.

Trans.connType = 12 identifies the Hypertac 160 pin, 128 channel connector used by commercially available probes for NDT applications. This connector type is used on current production versions of the UTA 160-DH/32 LEMO and UTA 160-SH/8 LEMO adapter modules, with no disconnect sensing for the Hyertac connector.

There are two other ‘special’ Trans.connType values that can be defined in a user’s SetUp script:

Trans.connType = 0 identifies a simulation-only script that is not intended to be used on a HW system at all, and thus there is no need to identify a probe connector interface. When this value is present in the script, the system will operate in simulation mode only (and will ignore the configuration of the hardware system if one is

present). None of the system HW compatibility constraints will be applied (for example, the number of transmit and receive channels can be set to any desired value in the range from 1 to 256). Setting Trans.connType to 0 has the same effect as setting Resource.Parameters.simulateMode to 1 in earlier SW releases (if a user script has set Trans.connType to 0 and Resource.Parameters.simulateMode is undefined, VSX will set a default value of 1 for simulateMode). VSX will assign a default Trans.connType value of zero for any script that has not defined it but has set Resource.Parameters.numTransmit to a value incompatible with the HDI connector; the result will be that the script will only run in simulation mode.

Trans.connType = -1 is for use in test scripts that are intended to be compatible with all UTA adapters. When VSX initializes a script with this value for use on a HW system, the -1 will be replaced with the index value representing the connector type that is actually present on the UTA adapter attached to the system. This allows test scripts or other utilities to be written in a 'generic' format that will automatically adapt to the actual system HW configuration when they are run.

Element to Channel Mapping

As was the case in all earlier Vantage releases, the *Trans.Connector* array (or the selected column of *Trans.HVMux.Aperture* for HV Mux probes) defines the mapping from individual transducer elements to the associated channel in the Vantage HW system for the selected connector or connectors. These channels are identified in the *Sequence Programming* manual as "I/O channels", to distinguish them from physical hardware channels (in some cases, such as when using a single HDI connector on a Vantage 256 system, the number of I/O channels may be less than the number of physical HW channels). For all Vantage software releases since 2.11 there will always be an exact 1-to-1 mapping between I/O channels and columns of the Receive Data buffer in the Matlab workspace; the number of columns in the Receive buffer definition (*Resource.RcvBuffer.colsPerFrame*) will always be equal to the number of I/O channels for the selected connector(s).

In summary, the length of the Trans.Connector array will always be equal to the number of transducer elements identified in the Trans structure. The Nth entry in that array will define the connection for element N of the transducer definition, and the value of that entry will identify the I/O channel number the element is connected to. A value of zero means that element is not connected at all, and cannot be used by the system. All of this remains exactly the same for the UTA system configuration, for all UTA adapter modules.

Defining the Trans.Connector array for a user-supplied transducer

If you are defining a script for a probe recognized by the Vantage software and supported through the *computeTrans* function, all of the element-to-channel mapping is done for you and most of the time you don't even have to think about it. But if you need to connect test equipment to a particular individual element signal at the probe connector, you will need to find the associated connector pin number (the Trans.Connector array defines the element-to-channel mapping but does not specify which physical connector pin is associated with each element). Even worse, if you are connecting a transducer that is not supported by *computeTrans* to the system, you will have to figure out how to define the Trans.Connector array. These issues were complicated enough on the original system; in the UTA system they get even more confusing since there are multiple UTA adapters supporting multiple connector types to choose from, and each UTA adapter and connector has a unique pin assignment at the connector and a unique mapping of those pins to system channels.

The Vantage SW includes some utilities to make the element-to-channel-to-connector pin mapping easier to understand:

showEL After running any script with VSX, and then exiting VSX, the Matlab workspace will still contain all of the structures that were used while the script was running. Type the command *showEL* at the command prompt, and a listing of all transducer elements will be displayed in the Matlab Command Window, showing the system I/O channel number, physical connector pin number, and Apod array entry number (applies to both Receive.Apod and TX.Apod) associated with that element. Enter the command with an argument, i.e. *showEL(7)*, and the mapping only for the identified element will be displayed. While working with a script, this utility makes it quick and easy to find the connector pin number and I/O channel number (which is also the column number in the Receive Buffer for the associated receive data) for that element.

showCH and **showAPOD** These utilities are identical to showEL as described above and will display the same information, except that the listing will be sorted by I/O channel numbers or Apod array entries instead of by elements.

To define the Trans.Connector array for a probe that is not recognized by *computeTrans*, you must first determine from the probe design documentation which physical connector pin each element connects to. Then you can use one of the utilities described above to find the mapping from each connector pin to system I/O channel, so you can put the correct channel number in each entry of the Trans.Connector array. Refer to the *Sequence Programming* manual for additional information on integrating custom transducers on the system.

11 Software Licensing

The Verasonics Software now contains a licensing mechanism. Software licensing was introduced in the Vantage 2.8 software release and has been incrementally enhanced in subsequent software releases.

Capabilities That May Be Licensed

As of this release, the following capabilities may be licensed:

1. Hardware

   A Hardware license enables the use of the Vantage software with one or more Verasonics Vantage hardware unit systems.

2. Simulation

   A Simulation license enables the use of the Vantage software when it is disconnected from a Verasonics Hardware system.

3. ArbwaveTk

   An ArbwaveTk license enables the use of the Arbitrary Waveform Toolkit, a subsystem of the Vantage software.

   With an ArbwaveTk license, the user may run the "Gui_Arb.m" utility within the ArbwaveToolbox software package provided with Vantage software releases. This is a stand-alone, self-contained software utility used for generation and analysis of arbitrary waveforms.

4. ArbwaveTx

   An ArbwaveTx license enables the generation of custom designed waveforms when the Vantage software is used in conjunction with a Vantage hardware unit system.

   When this capability is enabled, the Vantage hardware will transmit any waveform it is capable of generating. When this capability is disabled, the Vantage hardware will only transmit waveforms that can be produced with the parametric TW.type.

5. ExtendedTx

   An ExtendedTx license enables the generation of extended transmit waveforms when the Vantage software is used in conjunction with a Vantage hardware unit system.

   This license permits generating "push" transmit waveforms using TPC Profile 5 with the internal auxiliary TPC power supply on the Vantage hardware. This capability has no counterpart in simulation mode, so the presence/absence of this capability has no effect on the Vantage software simulation functionality.

6. Trigger

   A Trigger license enables the use of the Trigger In and Trigger Out capabilities of the Vantage hardware unit, when that hardware is connected to the Verasonics Vantage software. Trigger capability is an optional feature on Vantage 64 and Vantage 64 LE systems, but is automatically licensed as a standard feature on Vantage 128 and Vantage 256.

7. Recon

   The Recon license enables the use of the Reconstruction processing function within the Vantage software. Recon capability is an optional feature on the Vantage 64 system, but is automatically licensed as a standard feature on all other Vantage systems.

8. zMeas

   A zMeas license enables the use of the Impedance Measurement utility, a subsystem of the Vantage software.

   With a zMeas license, the user may run the "showImpedance" utility within the Tools / Impedance Measurement software package provided with Vantage software releases. This is a stand-alone, self-contained software utility used for measuring the impedance of transducer elements as seen through the probe connectors on the front of the system.

About Your Software Licenses

The software license delivered with your system enables every capability you have licensed from Verasonics.

All of the licenses for a single organization are contained within one software license file. Accordingly, you should always update all of your systems when you receive a new license file from us.

For example, if you have two Vantage systems, two host controllers, and three workstations using the Vantage software in simulation mode, the licenses for all of these systems are contained within a single license file, and that single file should be replicated to all of your host controllers and simulation workstations.

Your current software license is bundled with the software release built for your organization and stored in your download folder.

Updating Your Software License

From time to time you may need to update your software license. When this happens, please do the following:

1. On the system that is to be licensed, please install this software release if it is not already installed.
2. On the system that is to be licensed, run the *systemInfo* command from the Matlab command prompt.
3. The *systemInfo* command will generate a file, *systemInfo.txt* in the current directory.
4. Attach the *systemInfo.txt* file to an email message. In the email message, describe the situation briefly and send the email to licensing-request@verasonics.com.
5. Your email will received, processed and an updated license will be then sent to you by email in a timely manner.

> **Note**
>
> Please do not request a license without attaching an appropriate *systemInfo.txt* file. We cannot generate a license for you without this information. Requesting a license without attaching this information will only increase the time required to get you an updated license.

Temporary Software Licenses

In the case of an emergency, you can contact your local Verasonics representative or Verasonics Technical Support (support@verasonics.com) for a temporary software license.

Grace Period

All capabilities are licensed for a specific period of time. This period of time is extended by a *grace period* of thirty calendar days duration.

If you know that you will be using your Vantage system for a critical task, it is prudent to check ahead of time to make sure that the capabililties you intend to use have sufficient time remaining on them such that they will continue to be available to you when you need them.

Additional Licenses

Your Service Level Agreement with Verasonics entitles you to additional licenses. Please contact Verasonics for additional information on the number of licenses you are entitled to receive.

Once you know the number of additional licenses you are entitled to receive, please refer to *Updating Your Software License*, above.

Locating Your Software License File

Your software license file is named *license.enc*. It is expected to be found at the root level in your Vantage project folder.

> **Note**
> Software license files are binary, encrypted files. Tampering with the Verasonics software licensing mechanism may breach your contractual agreement(s) with Verasonics.

Installing a New Software License File

If you receive an updated software license file from Verasonics, you install it by copying it, unmodified, into your Vantage project folder. Once this has been done, please perform a graceful shut-down and restart of your Verasonics Vantage system.

Reviewing Your Licensed Capabilities

You can review the contents of your software license file at any time.

To do this, run the *systemInfo* command from the Matlab command prompt.

Then review the file *systemInfo.txt* in the Matlab text editor, or another text editor of your choice.

12 Maintenance

Fuse Replacement

There are two fuses in the AC Inlet on the back of the Vantage system. Replace with 10A 250V, Slo Blo, Bussman, MDA Series Fuses.

Contacting Verasonics Support

If you need support for any reason, please send an email to support@verasonics.com.

13 Sécurité et Conseils Réglementaires

Explication des symboles

Symbol	Meaning	Explanation
	"Voir le Manuel de l'Utilisateur"	(Ce Document)
	"Attention"	Ce symbole indique une situation potentiellement dangereuse qui peut entraîner des blessures mineures ou modérées. Le Manuel de l'Utilisateur (ce document), et en particulier les consignes de securité suivants, doit être consulté quand ce symbole est en évidence.
	"Tensions Dangereuses"	Ce symbole indique la présence de tensions dangereuses à l'intérieur du couvercle, et qu'il n'y a aucun composant réparable par l'utilisateur. Ne pas enlever les panneaux latéraux!

Mesures de Sécurité

Opérateur: Il est important que vous suiviez ces exigences de sécurité pour vous protéger et pour la protection du système Verasonics. Veuillez exiger a tout nouvel utilisateur du système de se familiariser avec les consignes de sécurité dans ce manuel.

	NOTE: Le système Verasonics n'est pas certifié pour l'utilisation avec des sujets humains. Pour plus de détails sur les risques et sur les prévisions de sécurité du système, et pour autres informations sur les précautions et préparations à prendre afin d'utiliser le système dans le cadre de la recherche avec des sujets humains, veuillez contacter Verasonics: Support@Verasonics.com.

- **Après l'installation initiale du système** (et après toute installation de logiciel), lancez la routine d'auto-test (à la commande Matlab, tapez "VerasonicsVerificationTest"). Assurez-vous d'abord qu'aucune sonde ne soit branchée. Veuillez prendre contact avec "Verasonics Support" si le test ne fonctionne pas, ou si le test indique un problème. Veuillez répéter le test après transport du système, après avoir débranché / rebranché le câblage du système ou autre démontage partiel, et après installation de logiciel, etc.
- **Ne jamais faire fonctionner le système sans panneaux en place**. Les panneaux sont nécessaires pour le refroidissement du système, le blindage CEM, et la protection de l'utilisateur et du système. Des tensions potentiellement dangereuses existent à l'intérieur du boîtier. Aucune pièce interieure n'est réparable par l'utilisateur. En cas de panne, indiquée par l'auto-test, ou par analyze de résultats, le système doit être livré a Verasonics pour mise à niveau ou tout autre entretien. Preuve de l'enlèvement non autorisé des couvercles latéraux peut entraîner l'annulement de la garantie du système.
- **Ne jamais stocker les liquides sur le système ou à proximité**, et surtout ne jamais renverser de liquide sur ou dans l'enceinte pour protéger l'utilisateur et le système d'une défaillance électrique. Si un déversement se produit, éteigner immédiatement le système (ordinateur et VDAS inclus) et contacter Support@Verasonics.com. Assurez-vous que des mesures prises pour protéger le système contre les déversements ne restreignent pas aussi le flux d'air pour le refroidissement.
- **Certaines sondes prototypes peuvent être mal blindées** et peuvent interférer avec d'autres instruments, et en présence

de dispositifs médicaux sensibles aux rayonnements RF, des soins supplémentaires doivent être pris. Si le système sera utilisé dans un environnement où cela pourrait gêner, le client doit effectuer des tests CEM de la configuration complète du système (sonde, cablage, VDAS, ordinateur, et tous les autres appareils connectés). Les essais CEM des composantes individuelles ne sont pas suffisants pour estimer la performance EMC du système global.

- **Sécurité et alimentation du système complet**: En usage normal, l'appareil VDAS doit être interconnecté avec un ordinateur, un écran, et potentiellement d'autres dispositifs qui peuvent chacun avoir leur propre connexion d'alimentation. L'utilisateur doit être conscient du fait que le courant de fuite du à l'ensemble de ces dispositifs interconnectés peut dépasser les limites de sécurité. Autres problèmes avec les connexions à l'alimentation, comme les connexions à terre inadéquates, peuvent entraîner des risques de sécurité potentiellement graves à l'opérateur ou aux sujets de l'essai. Il est recommandé que l'utilisateur effectu des mesures de sécurité de ligne, par exemple pour l'isolation, courant de fuite, et impédance de la connexion à terre. Ces vérifications de sécurité de l'alimenation doivent être répétées chaque fois que le système est démonté et remonté, ou soumis à des transports rugueux.
- **Rappel**: Verasonics recommande a l'utilisateur de lancer l'auto-test (VerasonicsVerificationTest) après toutes modifications, comme par exemple, mis à jour (logiciel ou matériel), transporté, ou démonté et remonté. Pour exécuter le test, démarrez le système, lancez Matlab, tapez "VerasonicsVerificationTest" et répondez aux questions eventuelles. Le test devrait terminer avec un message de success (Pass) ou erreur (Fail). En cas d'erreur, veuillez transmettre le fichier creé par le test à support@verasonics.com.

Dégâts de transports et livraison

Après toutes livraisons veuillez examiner soigneusement le carton d'expédition pour tout signe d'endommagement. Examinez également les capteurs de choc attachés à l'exterieur du carton. Veuillez nous communiquer au plus tôt si l'un d'entre eux soit déclenché, ou s'il y a d'autres indications de dommages ou de mauvais traitement en cours d'envoi. Après avoir déballé le système, inspecter soigneusement le boitier du VDAS et ses connecteurs afin de vérifier que la livraison n'aurait pas abimée le système.

Immunité CEM (Sensibilité au bruit électromagnetique)

Si l'image ultrasonore apparaît contaminé par du bruit, il est possible que le système ne soit pas suffisamment blindé contre le rayonnement ou la conduction d'énergie provenant de l'environment électromagnetique. Dans ce cas, veuillez contacter Verasonics pour des conseils supplémentaires: support@verasonics.com.

Mise à la terre électrique

Lors de l'installation, assurez-vous que le circuit d'alimentation soit fourni d'une connexion à la terre fiable. Assurez-vous aussi d'utiliser le cable d'alimentation comprenant un connecteur a trois broches (avec terre) fourni par Verasonics.

Transport du Système

Prenez soin en soulevant le système Vantage, qui peut peser jusqu'à 20 kg (44 lb). Si le modèle est capable d'émission HIFU, le boitier peut peser jusqu'à 26 kg (57 lb). Lors de déplacement, veuiller toujours utilizer le carton d'emballage fourni par Verasonics, conçu pour la protection du système contre les chocs et abus de transport. Le carton est muni de poignées pour faciliter le soulèvement. Rappelez-vous de plier les genoux!

14 Conseils a l'Utilisateur

- Si l'installation du logiciel est nécessaire, suivez les instructions d'installation soigneusement et précisement, en faisant particulièrement attention au redémarrage automatique pour assurer la configuration correcte du logiciel. L'installation du logiciel sera corrompue si vous relancez manuellement avant que les redemarrages automatiques puissent se faire.
- Utilisez le système uniquement à des températures ambiantes comprises entre 50 à 95ºF (10 à 35 ºC)

- Assurez-vous que la performance de la ligne d'alimentation soit entre les limites spécifiées pour: la tension, la fréquence, le niveau de puissance disponible, ainsi que les niveaux de surtensions, les baisses de tension, et les décrochements.
- Ne bouchez pas les grilles de ventilation qui protègent le système contre la surchauffe. Si les ventilateurs ne fonctionnent pas lorsque le système et l'ordinateur sont en marche, veuillez éteindre le système et prendre contact avec Verasonics au plus tôt.
- Auto-test du système: S'il y a jamais question de savoir si le système fonctionne toujours correctement, éteignez l'ordinateur et aussi l'interrupteur de l'alimentation du VDAS. Apres quelques secondes, d'abord allumez l'interrupteur du VDAS, et ensuite redemarrez l'orinateur. Lors du redémarrage et le relancement de MATLAB, débranchez tous les transducteurs et lancez le programme d'auto-test Verasonics "VerasonicsVerificationTest", et vérifiez que le système passe sans aucune erreur. En cas d' erreur, ou d'avis d'échec apparaissant dans la fenêtre de commande, contactez support@verasonics.com, et prennez soin d'envoyer les résultats du test.
- Si l'utilisateur a l'intention de controller la sortie acoustique du système, le programme VerasonicsVerificationTest doit être exécuté a chaque démarrage afin d'assurer que le système fonctionne selon ses spécifications. Les mesures de sortie acoustiques, ou d'autres tests de vérification effectuées sur le système, doivent être verifiées après chaque modification du logiciel (software) ou du materiel (hardware), de mise à niveau, ou de reparation ou maintenance.
- Veuillez conserver le carton d'expédition et les matériaux d'emballage; ils ont été spécialement conçus pour protéger le système en cas de transport. Si jamais vous avez besoin d'expédier le système de nouveau à Verasonics ou à un autre endroit, l'emballage d'origine doit être utilisé.

Fin de la Version Française — End of French Document