

## Appendix D

### National Type Evaluation Technical Committee (NTETC) Software Sector

March 2 - 3, 2010  
Sacramento, California

#### Annual Meeting Summary

##### Carry-over Items

1. NCWM/NTEP Policies – Issuing CCs for Software .....	D2
2. Definitions for Software Based Devices.....	D2
3. G-S.1. Identification (Software) .....	D4
4. Identification Of Certified Software .....	D8
5. Software Protection/Security .....	D11
6. Software Maintenance and Reconfiguration.....	D13
7. Verification In The Field by the Weights/Measures Inspector .....	D16
8. NTEP Application .....	D17

##### New Items

9. Training of Field Inspectors.....	D17
10. Next Meeting .....	D19

##### Appendices

Appendix A: Report on 2009 Interim Meeting .....	D20
Appendix B: Report on International W&M Activity .....	D21
Appendix C: 2010 Software Sector Meeting Attendee List .....	D22

## **Carry-over Items**

### **1. NCWM/NTEP Policies – Issuing CCs for Software**

*Source:* NCWM Reports

*Background:* For additional background on this item, see the 2009 Software Sector Meeting summary.

*Recommendation:* The Sector recommended the following language to be submitted to the National Type Evaluation Program (NTEP) Committee as a policy change.

**Software Requiring a Separate CC: Software, which is implemented as an add-on to other NTEP Certified main elements to create a weighing or measuring system and its metrological functions, are significant in determining the first indication of the final quantity. Such software is considered a main element of the system requiring traceability to an NTEP CC.**

**NOTE: OEM software may be added to an existing CC or have a stand-alone CC with applicable applications (e.g., a manufacturer adding a software upgrade to their ECR or point-of-sale system, vehicle scale weigh-in/weigh-out software added as a feature to an indicating element, automatic bulk weighing, liquid-measuring device loading racks, etc.) and minimum system requirements for “type P” devices (see proposed software definition below). It may be possible for a manufacturer to submit a single application for both hardware and software contained in the same device. A single CC would be issued.**

**In this instance, OEM refers to a 3<sup>rd</sup> party. The request to add software could be made by the original CC holder on behalf of the 3<sup>rd</sup> party. Alternatively, a new CC could be created that refers to the original CC and simply lists the new portions that were examined.**

The NTEP committee included this item in their agenda (NTEP Committee 2009 Interim Agenda Item 8); there was no discussion during the open hearing, and this became a Voting item for the 2009 Annual Meeting. At the 2009 National Conference on Weights and Measures (NCWM) Annual Meeting, this proposal was passed unanimously by the Conference.

*Discussion:* The NTEP Administrator was asked if there is to be any actual change in any document or is this strictly a procedural change? How do the labs know they can/should handle software items differently now? The answers to these questions were: there have not been any changes to Pub 14 this year. The Certificates of Conformance (CC) can now say “software.” The labs know this; NTEP policy is communicated to the labs. It was suggested that software could be a secondary classification on the certificates.

*Conclusions:* Our work is complete on this item; it will be removed from the agenda.

### **2. Definitions for Software Based Devices**

*Source:* 2009 Carryover Item 310-2. This item originated from the National Type Evaluation Technical Committee (NTETC) Software Sector and first appeared on the Committee’s 2007 Agenda as Developing Item Part 1, Item 2.

**From: NCWM Publication 15, 2010:**

**310-2 Appendix D – Definition of Electronic Devices, Software-Based and Built-for-Purpose Device**

*Item under Consideration:*

Delete the current definition of built-for-purpose device as follows:

~~***Built-for-purpose device. Any main device or element, which was manufactured with the intent that it be used as, or part of, a weighing or measuring device or system. [1.10] (Added 2003)***~~

and, add a new definition and a cross-reference to Appendix D in the National Institute of Standards and Technology (NIST) Handbook 44 (HB 44) for “Electronic devices, software-based” as follows to replace the current definition of “built-for-purpose device”:

**Electronic devices, software-based. – Weighing and measuring devices or systems that use metrological software to facilitate compliance with Handbook 44. This includes:**

**(a) Embedded software devices (Type P), aka built-for-purpose. – A device or element with software used in a fixed hardware and software environment that cannot be modified or uploaded via any interface without breaking a security seal or other approved means for providing security and will be called a “P,” or**

**(b) Programmable or loadable metrological software devices (Type U), aka not-built-for-purpose. – A personal computer or other device and/or element with PC components with programmable or loadable metrological software and will be called “U.” A “U” is assumed if the conditions for embedded software devices are not met.**

**Software-based devices – See Electronic devices, software-based.**

**Background:** For additional background information on this item, please reference the 2009 Software Sector Meeting Summary and the 2010 NCWM Interim Meeting Agenda (Pub 15)

At its 2009 Interim Meeting, the Central Weights and Measures Association (CWMA) received comments that the proposal is sufficiently developed and recommends moving this item forward as a Voting item on the Committee’s Agenda. At its 2009 Annual Technical Conference, the Western Weights and Measures Association (WWMA) received comments from Mr. Lou Straub, speaking on behalf of the Scales Manufacturers Association (SMA), indicating the SMA continues to oppose this item, noting that requirements should apply equally to the two different device types described. The WWMA received no other input on this item and recommends this item should remain Informational until the Software Sector has had an opportunity to review comments from the 2009 NCWM Annual Meeting and any comments made at subsequent regional weights and measures association meetings. At its 2009 Annual Meeting, the Southern Weights and Measures Association (SWMA) recommended keeping the status of this proposal to delete the current definition of built-for-purpose device and add a new definition and a cross-reference to Appendix D in HB 44 for “Electronic devices, software-based” to replace the current definition of “built-for-purpose device” as an Informational item. The SWMA agreed that the Software Sector should continue to work on the proposal until it arrives at some final language. During its 2009 Interim Meeting, the Northeastern Weights and Measures Association (NEWMA) stated that it supports the Committee’s decision to keep this item Informational to allow updated comments from the regional Weights and Measures associations and other interested parties based on information in the summary of the March 2009 meeting of the Software Sector. The item remains as an Informational item on 2010 Annual Meeting Agenda; the Specifications and Tolerances (S&T) Committee indicated that they look forward to additional work being done on this item by the Sector.

**Discussion:** Initially it was decided to table discussion on this item; as we worked on items further down the list, we would see if it was really necessary to include the ‘Type P’ and ‘Type U’ differentiation at this time; if so, we

would come back and work on the definitions. In particular, Agenda Item 3 (which contained references to the proposed definitions) would be examined in more detail to see if we couldn't satisfy the concerns of the SMA by avoiding differentiation of device types for identification purposes.

**Conclusion:** When all other Agenda items had been discussed, it was determined that there was no real need to introduce this differentiation in device types at the current time. It was decided that we would recommend to S&T that this item be withdrawn for now (with the realization that work on future items may require we reintroduce the concept). The previously proposed language is recorded herein if future requirements would revive the need for the definitions to differentiate between device types.

### 3. G-S.1. Identification (Software)

**Source:** NTETC Software Sector

**Background:** During their October 2007 meeting, the Sector discussed the value and merits of required markings for software. This included the possible differences in some types of devices and marking requirements. After hearing several proposals, the Sector agreed to the following technical requirements applicable to the marking of software.

1. The NTEP CC Number must be continuously displayed or hard marked;
2. The version must be software-generated and shall not be hard marked;
3. The version is required for embedded (Type P) software;
4. Printing the required identification information can be an option;
5. Command or operator action can be considered as an option in lieu of a continuous display of the required information; and
6. Devices with Type P (embedded) software must display or hard mark make, model, S.N. to comply with G-S.1. Identification.

The Sector developed marking information requirements and submitted a proposal to the S&T Committee for considered inclusion in HB 44. There was much additional comment and various proposed versions of the table from NIST Weights and Measures Division (WMD), et al. (The complete background on this item can be seen in the 2010 Interim Meeting Agenda, NCWM Pub 15, 2010.)

The Sector noted that though currently it is allowable to display the CC number via a menu, there has been some challenges locating this information in the field due to the vagueness of the term "easily recognized." Hence, since it is left to the interpretation of the NTEP laboratory to ascertain whether a device's method for displaying the CC number meets the requirements, this vagueness has not been addressed in this new recommendation.

At the 2009 Software Sector Meeting, it was agreed that the proposed table had not accomplished the intended purpose of clarifying the requirements, indeed it seemed to have generated more confusion. Hence, this item was revisited from the beginning, and it was suggested that a simpler approach be taken, namely to modify the text of G-S.1. to match our intent. The proposal from our Sector was as follows:

**G-S.1. Identification.** – All equipment, except weights and separate parts necessary to the measurement process but not having any metrological effect and manufactured ~~prior to~~ after January 1, 201X, shall be clearly and permanently marked for the purposes of identification with the following information:

- (a) the name, initials, or trademark of the manufacturer or distributor;
- (b) a model identifier that positively identifies the pattern or design of the device;

*(1) The model identifier shall be prefaced by the word "Model," "Type," or "Pattern."  
These terms may be followed by the word "Number" or an abbreviation of that word.*

*The abbreviation for the word “Number” shall, as a minimum, begin with the letter “N” (e.g., No or No.). The abbreviation for the word “Model” shall be “Mod” or “Mod.” Prefix lettering may be initial capitals, all capitals, or all lowercase.*

*[Nonretroactive as of January 1, 2003]*

*(Added 2000) (Amended 2001)*

- (c) *a nonrepetitive serial number, except for equipment with no moving or electronic component parts and ~~not built for purpose software based~~ **software that is not part of a Type P (built-for-purpose) device.***

*[Nonretroactive as of January 1, 1968]*

*(Amended 2003 **and 201X**)*

- (1) *The serial number shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required serial number.*

*[Nonretroactive as of January 1, 1986]*

- (2) *Abbreviations for the word “Serial” shall, as a minimum, begin with the letter “S,” and abbreviations for the word “Number” shall, as a minimum, begin with the letter “N” (e.g., S/N, SN, Ser. No., and S. No.).*

*[Nonretroactive as of January 1, 2001]*

- (d) *the current software version or revision identifier for ~~not built for purpose software-based~~ **electronic devices;***

*[Nonretroactive as of January 1, 2004]*

*(Added 2003) (**Amended 201X**)*

- (1) *The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.*

*[Nonretroactive as of January 1, 2007]*

*(Added 2006)*

- (2) *Abbreviations for the word “Version” shall, as a minimum, begin with the letter “V” and may be followed by the word “Number.” Abbreviations for the word “Revision” shall, as a minimum, begin with the letter “R” and may be followed by the word “Number.” The abbreviation for the word “Number” shall, as a minimum, begin with the letter “N” (e.g., No or No.).*

*[Nonretroactive as of January 1, 2007]*

*(Added 2006)*

- (e) *an NTEP Certificate of Conformance (CC) number or a corresponding CC Addendum Number for devices that have a CC. The CC Number or a corresponding CC Addendum Number shall be prefaced by the terms “NTEP CC,” “CC,” or “Approval.” These terms may be followed by the word “Number” or an abbreviation of that word. The abbreviation for the word “Number” shall, as a minimum, begin with the letter “N” (e.g., No or No.)*

*[Nonretroactive as of January 1, 2003]*

The required information shall be so located that it is readily observable without the necessity of the disassembly of a part requiring the use of any means separate from the device.

*(Amended 1985, 1991, 1999, 2000, 2001, 2003, **and**, 2006 **and 201X**)*

*G-S.1.1. ~~Location~~ Method of Marking Information for ~~Not Built For Purpose~~ all Software-Based Devices. – ~~For not built for purpose, software based devices manufactured prior to~~ **after** January 1, 201X, **either**:*

- (a) *The required information in G-S.1. Identification. ~~(a), (b), (d), and (e)~~ shall be permanently marked or continuously displayed on the device; or*
- (b) *The Certificate of Conformance (CC) Number shall be:*
  - (1) *permanently marked on the device;*
  - (2) *continuously displayed; or*
  - (3) *accessible through an easily recognized menu and, if necessary, a submenu. Examples of menu and submenu identification include, but are not limited to, “Help,” “System Identification,” “G-S.1. Identification,” or “Weights and Measures Identification.”*

**Note:** *For (b), clear instructions for accessing the information required in G-S.1. (a), (b), and (d) shall be listed on the CC, including information necessary to identify that the software in the device is the same type that was evaluated.*

*[Nonretroactive as of January 1, 2004]*

*(Added 2003) (Amended 2006 **and 201X**)*

**Discussion:** As was noted in the review of what transpired at the Interim Meeting, there appears to be continued resistance, especially from the SMA, to differentiating between Type P and Type U software types. From their perspective it is ‘all software’ and they are concerned that marking requirements will be more complex if we delineate between two different types of software-based devices. Also, the inspectors want to standardize the method of locating the marking information when it is being displayed via menu, and insist that it should be very simple for field personnel to locate. Some additional work by the group resulted in this modified proposal that does not include the new definitions and does not specifically delineate any device types (in fact, it removes the existing mention of ‘built-for purpose’):

**G-S.1. Identification.** – All equipment, except weights and separate parts necessary to the measurement process but not having any metrological effect ~~and manufactured after January 1, 201X~~, shall be clearly and permanently marked for the purposes of identification with the following information:

- (a) the name, initials, or trademark of the manufacturer or distributor;
- (b) a model identifier that positively identifies the pattern or design of the device;
  - (1) *The model identifier shall be prefaced by the word “Model,” “Type,” or “Pattern.” These terms may be followed by the word “Number” or an abbreviation of that word. The abbreviation for the word “Number” shall, as a minimum, begin with the letter “N” (e.g., No or No.). The abbreviation for the word “Model” shall be “Mod” or “Mod.” Prefix lettering may be initial capitals, all capitals, or all lowercase.*

*[Nonretroactive as of January 1, 2003]*

*(Added 2000) (Amended 2001)*

- (c) *a nonrepetitive serial number, except for equipment with no moving or electronic component parts ~~and not built for purpose software based software that is not part of a Type P (built for purpose) device.~~*

*[Nonretroactive as of January 1, 1968]*

*(Amended 2003 **and 201X**)*

- (1) *The serial number shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required serial number.*

*[Nonretroactive as of January 1, 1986]*

- (2) *Abbreviations for the word “Serial” shall, as a minimum, begin with the letter “S,” and abbreviations for the word “Number” shall, as a minimum, begin with the letter “N” (e.g., S/N, SN, Ser. No., and S. No.).*

*[Nonretroactive as of January 1, 2001]*

- (d) *the current software version or revision identifier for ~~not-built-for-purpose~~ software-based electronic devices;*

*[Nonretroactive as of January 1, 2004]*

(Added 2003) **(Amended 201X)**

- (1) *The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.*

*[Nonretroactive as of January 1, 2007]*

(Added 2006)

- (2) *Abbreviations for the word “Version” shall, as a minimum, begin with the letter “V” and may be followed by the word “Number.” Abbreviations for the word “Revision” shall, as a minimum, begin with the letter “R” and may be followed by the word “Number.” The abbreviation for the word “Number” shall, as a minimum, begin with the letter “N” (e.g., No or No.).*

*[Nonretroactive as of January 1, 2007]*

(Added 2006)

- (e) *an NTEP CC number or a corresponding CC Addendum Number for devices that have a CC. The CC Number or a corresponding CC Addendum Number shall be prefaced by the terms “NTEP CC,” “CC,” or “Approval.” These terms may be followed by the word “Number” or an abbreviation of that word. The abbreviation for the word “Number” shall, as a minimum, begin with the letter “N” (e.g., No or No.).*

*[Nonretroactive as of January 1, 2003]*

The required information shall be so located that it is readily observable without the necessity of the disassembly of a part requiring the use of any means separate from the device.

(Amended 1985, 1991, 1999, 2000, 2001, 2003, **and, 2006 and 201X**)

**Comments:** The thinking was that standalone software has no moving or electronic component parts and hence is not required to have a serial number. This was considered acceptable by the Sector; the Sector sees no value in requiring vendors submittals for NTEP approval that are software-only to print serial numbers on their distribution media (CD, DVD, etc). It was observed by California that if we continue with the concept of only examining ‘devices’ that typically off-the-shelf PC’s have their own serial number, generated by the manufacturer. This can and has been used by the inspectors as a means to meet G-S.1.(c) though the prefix/abbreviation is sometimes an issue since the PC manufacturer knows nothing about G-S.1.

It was also suggested that G-S.1.1.(b)(3) be modified to omit the term “easily recognized”; instead, a limited list of options would be available. A first pass at reworking G-S.1.1(b)(3) resulted in:

**G-S.1.1. Location Method of Marking Information for ~~Not-Built-For-Purpose~~ all Software-Based Electronic Devices. – For ~~not-built-for-purpose, software-based~~ devices ~~manufactured after January 1, 201X~~, either:**

- (a) *The required information in G-S.1. Identification. ~~(a), (b), (d), and (e)~~ shall be permanently marked or continuously displayed on the device; or*

- (b) *The CC Number shall be:*
- (1) *permanently marked on the device;*
  - (2) *continuously displayed; or*
  - (3) *accessible through one or, at most, two levels of access. ~~an easily recognized menu and, if necessary, a submenu. Examples of menu and submenu identification include, but are not limited to, “Help,” “System Identification,” “G-S.I. Identification,” or “Weights and Measures Identification.”~~*
    - (a) *For menu-based systems, “Metrology”, “System Identification”, or “Help”.*
    - (b) *For systems using icons, a metrology symbol (“M”), “SI”, or a help symbol (“?”), “i”, or an “i” within a magnifying glass).*

*Note: For (b), clear instructions for accessing the information required in G-S.1. (a), (b), (c), and (d) shall be listed on the CC, including information necessary to identify that the software in the device is the same type that was evaluated.*

[Nonretroactive as of January 1, 2004]  
(Added 2003) (Amended 2006 and 201X)

This new language for G-S.1.1.(3)(b) is in the early stages, and the Software Sector would like feedback regarding G-S.1.1(b)(3), particularly suggestions for specific allowed menu items/icons that should be included on the list.

**Conclusion:** The revised G-S.1. (and G-S.1.1.) above will be sent to NCWM S&T Committee as our updated recommendation.

*[Note: It was observed by WMD (after our meeting adjourned) that there have been several revisions, and revisions to revisions, to our G-S.1. proposals. The proofing (font, bold/italic, etc.) may no longer reflect the correct form with which changes are to be submitted, and they may not actually reflect the changes from what is currently in the 2010 Handbook. This needs to be addressed prior to submission to the S&T Committee; the Chair will compare the proposed language to the current HB 44 language and make sure the desired changes are marked properly in the forwarded proposal.]*

#### 4. Identification of Certified Software

**Source:** NTETC Software Sector

**Background/Discussion:** This item originated as an attempt to answer the question “How does the field inspector know that the software running in the device is the same software evaluated and approved by the lab?” In previous meetings it was shown that the international community has addressed this issue (both the European Cooperation in Legal Metrology (WELMEC) and the International Organization of Legal Metrology (OIML)). From WELMEC 7.2:

##### **Required Documentation:**

The documentation shall list the software identifications and describe how the software identification is created, how it is inextricably linked to the software itself, how it may be accessed for viewing and how it is structured in order to differentiate between version changes with and without requiring a type approval.



From OIML D 31:

The executable file “**tt100\_12.exe**” is protected against modification by a checksum. The value of checksum as determined by algorithm **XYZ** is **1A2B3C**.

Previous discussions have included a listing of some additional examples of possible valid methods (not limiting):

- CRC (cyclical redundancy check)
- Checksum
- Inextricably Linked version no.
- Encryption
- Digital Signature

Is there some method to give the Weights and Measures (WM) inspector information that something has changed? (Yes, the Category III audit trail or other means of sealing). How can the WM inspector identify an NTEP Certified version? (They can't, without adding additional requirements like what is described here, in conjunction with including the identifier on the CC).

The Sector believes that we should work towards language that would include a requirement similar to the OIML requirement in HB 44. It is also the opinion of the Sector that a specific method should not be defined; rather the manufacturer should utilize a method and demonstrate the selected identification mechanism is suitable for the purpose. It is not clear from the discussion where such proposed language might belong.

NTEP strongly recommends that metrological software be separated from non-metrological software for ease of identification and evaluation. From OIML:

Separation of software parts – All software modules (programmes, subroutines, objects etc.) that perform metrologically significant functions or that contain metrologically significant data domains form the metrologically significant software part of a measuring instrument (device or sub-assembly). The conformity requirement applies to all parts and parts shall be marked according to Section G-S-X.X.

If the separation of the software is not possible or needed, then the software is metrologically significant as a whole.

(Segregation of *parameters* is currently allowed - see table of sealable parameters)

Initial draft proposed language: (G-S.1.1.?)

Handbook 44 (This has been written into G-S.1.(d)(3):

**Identification of Certified Software:**

**Software-based electronic devices shall be designed such that the metrologically significant software is clearly identified by the version or revision number. The identification, and this identification of the software shall be inextricably directly and inseparably linked to the software itself. The version or revision number may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.**

Pub. 14:

**Identification of Certified Software:**

**Note:** Manufacturers may choose to separate metrologically significant software from non-metrologically significant software. Separation would allow the revision of the non-metrological portion without the need for further evaluation. In addition, non-metrologically significant

software may be updated on devices without breaking a seal, if so designed. Separation of software requires that all software modules (programs, subroutines, objects etc.) that perform metrologically significant functions or that contain metrologically significant data **domains** form the metrologically significant software part of a measuring instrument (device or sub-assembly). If the separation of the software is not possible or needed, then the software is metrologically significant as a whole. ~~The conformity requirement applies to all parts and parts shall be marked according to Section G-S-X.X.~~

The manufacturer must describe and possibly demonstrate how the version or revision identifier is directly and inseparably linked to the metrologically significant software. Where the version revision identifier is comprised of more than one part, the manufacturer shall describe which portion represents the metrological significant software and which does not.

From OIML D-31:

Legally relevant software of a measuring instrument / electronic device / sub-assembly shall be clearly identified with the software version or another token. The identification may consist of more than one part but at least one part shall be dedicated to the legal purpose.

The identification shall be inextricably linked to the software itself and shall be presented or printed on command or displayed during operation or at start up for a measuring instrument that can be turned off and on again. If a sub-assembly/an electronic device has neither display nor printer, the identification shall be sent via a communication interface in order to be displayed/printed on another sub-assembly/electronic device.

The first sentence of the first paragraph above is already addressed in HB 44's marking requirements.

**Recommendation:** Recommend the following change to HB 44, General Code: G-S.1.(d) to add a new subsection (3):

- (d) *The current software version or revision identifier for ~~not-built-for-purpose~~ software-based electronic devices;*  
[Nonretroactive as of January 1, 2004]  
(Added 2003) **(Amended 201X)**
- (1) *The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.*  
[Nonretroactive as of January 1, 2007]  
(Added 2006)
- (2) *Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).*  
[Nonretroactive as of January 1, 2007]  
(Added 2006)
- (3) **The version or revision identifier shall be directly and inseparably linked to the software itself. The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.**  
[Nonretroactive as of January 1, 201X]  
**(Added 201X)**

Also the Sector recommends the following information be added to Pub. 14 as explanation/examples:

- Unique identifier must be displayable/printable on command or during operation, etc.
- At a minimum, a version/revision indication (1.02.09, rev 3.0 a, etc). Could also consist of /contain checksum, etc (crc32, for example)

**Conclusions:** The item needs additional discussion and development by the sector. Outstanding questions: If we allow hard-marking of the software identifier (the Sector has wavered on this in the past), does the above wording then imply that some mechanical means is required (i.e., physical seal) to ‘inseparably link’ the identifier to the software? Do we still have to be able to display/print the identifier if it is hard-marked?

## 5. Software Protection / Security

**Source:** NTETC Software Sector

**Background:** The sector agreed that HB 44 already has audit trail and physical seal, but the question on the table is does the Handbook need to be enhanced to sufficiently discourage the facilitation of fraud, intentional or accidental, where software is concerned?

WELMEC and OIML again have addressed this issue specifically when dealing with software. From WELMEC:

### Protection against accidental or unintentional changes

Metrologically significant software and measurement data shall be protected against accidental or unintentional changes.

### Specifying Notes:

Possible reasons for accidental changes and faults are: unpredictable physical influences, effects caused by user functions and residual defects of the software even though state of the art of development techniques have been applied.

This requirement includes:

- a) Physical influences: Stored measurement data shall be protected against corruption or deletion when a fault occurs or, alternatively, the fault shall be detectable.
- b) User functions: Confirmation shall be demanded before deleting or changing data.
- c) Software defects: Appropriate measures shall be taken to protect data from unintentional changes that could occur through incorrect program design or programming errors, e.g. plausibility checks.

### Required Documentation:

The documentation should show the measures that have been taken to protect the software and data against unintentional changes.

### Example of an Acceptable Solution:

- The accidental modification of software and measurement data may be checked by calculating a checksum over the relevant parts, comparing it with the nominal value and stopping if anything has been modified.
- Measurement data are not deleted without prior authorization, e.g. a dialogue statement or window asking for confirmation of deletion.
- For fault detection see also Extension I.

The Sector derived a suitable checklist for Pub 14 from the OIML checklist, and asked the current NTEP labs to begin using this checklist on a trial basis for new type approval applications.

<b>Devices with embedded software TYPE P (aka built-for-purpose)</b>	
--	--

	Declaration of the manufacturer that the software is used in a fixed hardware and software environment, and	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	cannot be modified or uploaded by any means after securing/verification	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	<i>Note: It is acceptable to break the “seal” and load new software, audit trail is also a sufficient seal.</i>	
	The software documentation contains:	
	description of all <del>the metrologically significant</del> functions, <b>designating those that are considered metrologically significant</b> <i>OIML states that there shall be no undocumented functions</i>	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	description of the securing means (evidence of an intervention)	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	software identification	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	description how to check the actual software identification	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	The software identification is:	
	clearly assigned to the metrologically significant software and functions	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	provided by the device as documented	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
<b>Personal computers, instruments with PC components, and other instruments, devices, modules, and elements with programmable or loadable metrologically significant software TYPE U (aka not built-for-purpose)</b>		
	The <i>metrologically significant</i> software is:	
	documented with all relevant (see below for list of documents) information	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	protected against accidental or intentional changes	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	Evidence of intervention (such as, changes, uploads, circumvention) is available until the next verification / inspection (e.g., physical seal, Checksum, CRC, audit trail, etc. means of security)	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
<b>Software with closed shell (no access to the operating system and/or programs possible for the user)</b>		
	Check whether there is a complete set of commands (e.g. function keys or commands via external interfaces) supplied and accompanied by short descriptions	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	Check whether the manufacturer has submitted a written declaration of the completeness of the set of commands	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
<b>Operating system and / or program(s) accessible for the user:</b>		
	Check whether a checksum or equivalent signature is generated over the machine code of the metrologically significant software (program module(s) subject to <del>legal control</del> WM jurisdiction and type-specific parameters)	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	Check whether the metrologically significant software will detect and act upon any unauthorized alteration of the metrologically significant software using simple software tools (e.g., text editor)	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
<b>Software interface(s)</b>		
	Verify the manufacturer has documented:	
	the program modules of the metrologically significant software are defined and separated	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>

	the protective software interface itself is part of the metrologically significant software	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	the <i>functions</i> of the metrologically significant software that can be accessed via the protective software interface	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	the <i>parameters</i> that may be exchanged via the protective software interface are defined	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	the description of the functions and parameters are conclusive and complete	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	there are software interface instructions for the third party (external) application programmer.	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>

The Sector hopes to obtain feedback at this meeting from the NTEP labs regarding this checklist.

**Discussion:** The labs again indicated they had not had a chance to utilize the checklist. The list was reviewed and some minor modifications to the checklist text were incorporated as shown in this excerpt:

	The software documentation contains:	
	description of all <del>the metrologically significant</del> functions, <b>designating those that are considered metrologically significant</b> <i>OIML states that there shall be no undocumented functions</i>	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	description of the securing means (evidence of an intervention)	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>
	software identification	Yes <input type="checkbox"/> No <input type="checkbox"/> N/A <input type="checkbox"/>

**Conclusion:** Work is ongoing on this item with the intent that it eventually be incorporated as a checklist in Pub 14; again the labs are requested to try utilizing this checklist for any evaluations on software-based electronic devices.

## 6. Software Maintenance and Reconfiguration

**Source:** NTETC Software Sector

**Background:** After the software is completed, what do the manufacturers use to secure their software?

**Discussion:** The following items were reviewed by the Sector. Note that agenda item 3 also contains information on Verified and Traced updates and Software Log.

- a. Verify that the update process is documented (OK).
- b. For traced updates, Installed Software is authenticated and checked for integrity.

Technical means shall be employed to guarantee the authenticity of the loaded software (i.e., that it originates from the owner of the type approval certificate). This can be accomplished, for example, by cryptographic means like signing. The signature is checked during loading. If the loaded software fails this test, the instrument shall discard it and either use the previous version of the software or become inoperative.

Technical means shall be employed to guarantee the integrity of the loaded software i.e., that it has not been inadmissibly changed before loading. This can be accomplished for example, by adding a checksum or hash code of the loaded software and verifying it during the loading procedure. If the loaded software

fails this test, the instrument shall discard it and either use the previous version of the software or become inoperative.

Examples are not limiting or exclusive.

- c. Verify that the sealing requirements are met.

The Sector asked, what sealing requirements are we talking about?

This item is only addressing the software update, it can be either verified or traced. It is possible that there are two different security means, one for protecting software updates (software log) and one for protecting the other metrological parameters (Category I, II, or III method of sealing).

Some examples provided by the Sector members include, but are not limited to:

Physical Seal, software log

Category III method of sealing can contain both means of security

- d. Verify that if the upgrade process fails, the device is inoperable or the original software is restored.

The question before the group is can this be made mandatory?

The manufacturer shall ensure by appropriate technical means (e.g., an audit trail) that traced updates of metrologically significant software are adequately traceable within the instrument for subsequent verification and surveillance or inspection. *This requirement enables inspection authorities, which are responsible for the metrological surveillance of legally controlled instruments, to back-trace traced updates of metrologically significant software over an adequate period of time (that depends on national legislation).* The statement in italics will need to be reworded to comply with U.S. WM requirements.

The Sector **agreed** that the two definitions below for verified update and traced update were acceptable.

#### **Verified Update**

A verified update is the process of installing new software where the security is broken and the device must be re-verified. Checking for authenticity and integrity is the responsibility of the owner/user.

#### **Traced Update**

A traced update is the process of installing new software where the software is automatically checked for authenticity and integrity, and the update is recorded in a software update log or audit trail.

The Sector also worked towards language proposed for defining the requirements for a Traced Update (currently considered as relevant for Pub 14):

**For a Traced Update, an event logger is required. The logger shall be capable of storing a minimum of the 10 most recent updates. An entry shall be generated for each software update.**

**Use of a Category 3 audit trail is acceptable required for the software update logger Traced Update. In this case the existing requirement of 1,000 entries supersedes the 10 entry requirement. If software update is the only loggable event, then the Category 3 audit trail can be limited to only 10 entries. A software update log entry representing a software update shall include the following: the software identification of the newly installed version.**

- **An event counter;**

- the date and time of the change; and
- the event type/parameter ID, which indicates a software update event (if not using a dedicated update log);
- the new value of the parameter, which is the *software identification of the newly installed version.*

A Category III device may include the software update events in the Category III audit log in lieu of a separate software update log; the existing requirement for 1,000 entries supersedes the requirement for 10 entries.

The traceability means and records are part of the metrologically significant software and should be protected as such. If software separation is employed, the software used for displaying the audit trail belongs to the fixed metrologically significant software. (Note: This needs to be discussed further due to some manufacturer's concerns about where the software that displays the audit trail information is located and who has access if this feature is provided. Manufacturers did indicate that there are methods available to encrypt the audit trail information; however, it cannot be protected from being deleted.) (include flowchart from OIML D 31)

The Sector discussed how to best move this item forward, and there was also some discussion as to whether new language for the General Code was required. The following new text was proposed:

#### G-S.9. Metrologically Significant Software Updates

The updating of metrologically significant software shall be considered a sealable event. Metrologically significant software that does not conform to the approved type is not allowed for use.

The NTEP Administrator indicated that the current requirements in G-S.8. already make the statement that any changes that affect metrological function are sealable, hence software updates may be covered and the proposed G-S.9. unnecessary. Mr. Todd Lucas suggested we go ahead and submit the proposed G-S.9. to the Committee and request a clarification/interpretation of G-S.8.

At the 2009 meeting, the Sector opined that the explicit language proposed for G-S.9. is clearer than any implied requirement in G-S.8. The Sector would like a clarification/interpretation of G-S.8. as it relates to software updates from the S&T Committee (with their response preferably to be included in Pub 16). The Sector will also continue to develop the proposed text (and flow chart) targeted for inclusion in Pub 14.

**Discussion:** The Sector reviewed the proposal and reconsidered allowing a separate 'update log'. It was decided that this would probably generate confusion and is not likely to be adopted by manufacturers anyway. Hence, the previously proposed text was modified to require a Category III audit trail for 'traced updates':

~~For a Traced Update, an event logger is required. The logger shall be capable of storing a minimum of the 10 most recent updates. An entry shall be generated for each software update.~~—Use of a Category 3 audit trail is acceptable required for the software update logger Traced-Update. ~~In this case the existing requirement of 1,000 entries supersedes the 10 entry requirement.~~ If software update is the only loggable event, then the Category 3 audit trail can be limited to only 10 entries. A ~~software update~~ log entry representing a software update shall include the following:—the software identification of the newly installed version.

- An event counter;
- the date and time of the change; and
- the event type/parameter ID, which indicates a software update event (if not using a dedicated update log);

- ~~the new value of the parameter, which is the *software identification* of the newly installed version.~~

~~**A Category III device may include the software update events in the Category III audit log in lieu of a separate software update log; the existing requirement for 1,000 entries supersedes the requirement for 10 entries.**~~

*Conclusions:* The general consensus of the group after considering feedback from external interested parties is that a new G-S.9. with explicit requirements is not necessary (nor likely to be adopted by the Conference) and that this requirement belongs in the Pub 14 lists of sealable parameters rather than in HB 44; i.e.

The updating of metrologically significant software shall be considered a sealable event.

Additional work is to be done to further develop the proposed text toward inclusion in Pub 14.

## **7. Verification in the Field, By the W&M Inspector**

*Source:* NTETC Software Sector

*Background Question:* What tools does the field inspector need?

Possible Answers:

- Have NTEP CC No. continuously displayed. (needs some type of protection) during the normal weighing or measuring operation
- Clear and simple instructions on NTEP CC to get to the other inspection information
- The CRC, checksum, version number, etc., needs to be easily accessible from operator console.
- Inspector needs to know how to access audit trail.
- System information is easily accessible (ram, OS, etc.)
- System parameters are easily accessible (AZT, motion, time outs, etc.).

Some discussion about system information requirements for the inspector took place... does the inspector really need to have access to OS, RAM information, etc.? (General opinion seems to be if there is a dependency, then the NTEP lab would specifically include that requirement in the CC.)

Audit trail info – the question was asked, does there need to be a specific requirement for providing access to this information?

Regarding the concept of First Final – There was some concern expressed as to how the inspectors are able to discern where the indication of first final be found for the system (as opposed to the devices in the system). What devices in the system are of concern to the inspector? The NTEP Administrator indicated that field inspectors need to follow the system all the way to receipt/bill generation.

Data transmission is an issue when considering systems as opposed to devices... how far does the inspector's jurisdiction extend? (Should we model future requirements on the WELMEC section concerning DTD/DSD?) Decision: data transmission/storage is not currently being addressed by the Sector at this time. Since part of the Sector's mission is education, do we want to assist in developing training aids for labs/inspectors related to evaluating/inspecting software-based devices? This will be a topic to be added to the Sector's Agenda for the next meeting.

At the 2009 meeting, the Sector decided to continue to develop this item, and initiate a new Agenda item specific to inspector training in relation to evaluating/validating software-based devices.

*Discussion:* A question from the floor requested opinion as to whether this Agenda item continued to serve a purpose. During discussion, it was stated that the goals of this item have all been addressed as part of all the other



agenda items save one (training), and inspector training will now be covered in a new item (Training of Field Inspectors), leaving this item without merit.

**Conclusion:** No argument was made for retaining this item as a separate item on the Agenda. This item will be removed from future Agendas.

## **8. NTEP Application for Software requiring a separate Certificate of Conformance—based Electronic Devices**

**Source:** NTETC Software Sector

**Background/Discussion:** The purpose of initiating this item was to identify issues, requirements and processes for type approving Type U device applications. It was suggested that it may be useful to the labs to devise a separate submission form for software for Type U devices. Question: What gets submitted? What requirements and mechanisms for submission should be available?

Validation in the lab - all required subsystems shall be included to be able to simulate the system as installed.

It was noted this Agenda item is irrelevant if the NTEP Committee does not approve the pending item up for vote.

Mr. John Roach, CA NTEP Lab, stated that if the software package being evaluated supports platforms/subsystems from multiple manufacturers, testing should be done using at least two platforms/subsystems. Scale labs and scale manufacturers indicated that this is not usually done for scale evaluations.

Conclusion of 2009 Sector Meeting: The Sector will continue to develop this item, contingent on the status of the related NTEP Committee Agenda item after the 2009 Annual Meeting.

**Discussion:** Since the NTEP committee passed the related item at the Annual we will continue to work on this. The NTEP director indicated that we can move in this direction, but felt that it was somewhat premature to develop this thoroughly now. At the point where the sector has developed checklist requirements, then we could move to perhaps add a subsection to current NTEP applications for applicable software. Refer to D-31.6.1. It was also agreed that there seems to be no reason for limiting the scope of this item to software-only applications, and hence all software/software-based devices could benefit from an enhanced application process. Hence the description of this agenda item was modified as shown in the marked up heading.

**Conclusion:** The item will be revisited at the 2011 Meeting and it will be decided whether to begin further development of this item at this time.

### **New Items**

## **9. Training of Field Inspectors**

**Source:** NTETC Software Sector

**Background:** During discussions at the 2009 meeting, the Sector concluded that a new Agenda item should be initiated specific to the training of field inspectors in relation to evaluating/validating software-based devices.

**Discussion:** CA has an Examination Procedure Outline (EPO) that begins to address this. Use Handbook 112 as a pattern template for how it could read.

Items to be addressed:

- Certificate of Conformance
- Terminology (as related to software) beyond what is in HB 44.
- Reference materials / information sources
- Safety

### System Verification Tests

**NOTE:** Item numbers 1 through 5 apply to both weighing and measuring devices. Numbers 6 and 7 are specific to weighing devices; while numbers 9 and 10 apply to measuring devices.

1. Identification. The identification (ID) tag may be on the back room computer server and could be viewed on an identification screen on the computer monitor. The ID information may be displayed on a menu or identification screen. Though currently discouraged, some systems may be designed so the system must be shut down and reset to view the ID information. **G-S.1. (1.10.)**
  - 1.1. Manufacturer.
  - 1.2. Model designation.
2. Provisions for sealing. **G-S.8. [1.10.]; S.1.11. [2.20]; S.2.2. [3.30.]**
  - 2.1. Verify sealing category of device (refer to Certificate of Approval for that system).
  - 2.2. Verify compliance with certificate.
3. Units of measure.
  - 3.1. A computer and printer interfaced to a digital indicator shall print all metrological values, intended to be the same, identically. **G-S.5.2.2.(a); G-S.5.1. [1.10.]**
  - 3.2. The unit of measure, such as lb, kg, oz, gal, qts, liters, or whatever is used, must agree.
4. Operational controls, indications and features (buttons and switches). Verify that application criteria and performance criteria are met (refer to Certificate of Approval).
  - 4.1. Any indication, operation, function or condition must not be represented in a manner that interferes with the interpretation of the indicated or printed values.
5. Indications and displays.
  - 5.1. Attempt to print a ticket. The recorded information must be accurate or the software must not process and print a ticket with erroneous data interpreted as a measured amount.

### Weighing Devices

6. Motion detection.
  - 6.1. For railway track, livestock, and vehicle scales apply or remove a test load of at least 15 dwhile simultaneously operating a print button, push-button tare or push-button zero. A good way to do this is to try to print a ticket while pulling the weight truck or another vehicle onto the scale. Recorded values shall not differ from the static display by morethan 3d. Perform the test at 10 %, 50 % and 100 % of the maximum applied test load. **S.2.5.1.(a) [2.20.]; EPO NO. 2-3, 2.4.**
  - 6.2. For all other scales, apply or remove at least 5d. Printed weight values must agree with the static weight within 1d and must exactly agree with other indications. **S.2.5.4.(b) [2.20.]; EPO NO. 2-3, 2.4.**
7. Behind zero indication.
  - 7.1 Apply a load in excess of the automatic zero setting mechanism (AZSM) and zero the scale. **S.2.1.3. [2.20.]; EPO NO. 2-3, 2.4., 2.5.2.**

**Example:** On a vehicle scale have someone stand on the scale, then zero them off (AZSM is 3d). Remove the weight (person) and note the behind zero display (usually a minus weight value) or error condition.

    - 7.2. Attempt to print a ticket. With a behind zero condition, (manually or mechanically operated) a negative number must not be printed as a positive value.
8. Over capacity.
  - 8.1. Manually enter a gross weight if permissible or apply a test load in excess of 105 % of the scale’s capacity. **S.1.7. [2.20.]; S.1.12., UR.3.9. [2.20.]**
  - 8.2. Attempt to print a weight ticket. A system must not print a ticket if the manually entered weight or load exceeds 105 % of the scale capacity.

### **Measuring Devices**

10. Motion detection.
  - 10.1. Initiate flow through the measuring element. Attempt to print a ticket while the product is flowing through the measuring chamber. The device must not print while the indication is not stable. **S.2.4.1. (3.30.)**
11. Over capacity.
  - 11.1. Attempt to print a ticket in excess of the indicated capacity. A system must not print a ticket if the device is manually or mechanically operated in excess of the indicated value.

**NOTE:** Be aware of error codes on the indicator which may be interrupted as measured values.

**Conclusion:** This item is in the early stages; work will continue on the item working toward materials to aid in the training of field inspectors. It was indicated that working in conjunction with the Professional Development Committee (PDC) to develop training materials, etc. would be a logical path of progress once we have developed the information content to include.

## **10. Next meeting**

**Background:** The Sector is on a yearly schedule for Sector meetings. The NTEP Administrator determines when the next meeting is possible.

**Discussion:** The NTEP Administrator indicated that the NTETC meetings are to be scheduled where the Conference gets the most ‘bang for the buck’, so that implies (considering our spring schedule) one of the states with an NTEP lab. Hence we’ve been rotating among Annapolis, Maryland, Columbus, Ohio, and Sacramento, California. It was also mentioned by the Technical Advisor that this rotating of the location has been quite beneficial to the group, considering the variety of input from individuals not typically able to make the trip to attend distant meetings.

**Conclusion:** Given the above, it was suggested that it would be Maryland’s turn in 2011. In keeping with the March timeframe and trying to avoid the last blast of winter, the group decided to return to Annapolis, Maryland, preferably March 15-16, 2011. Second choice would be the following week (March 22-23<sup>d</sup>). The Maryland lab personnel will assist the NCWM staff in suggesting one or more suitable host facilities for the meeting.

***Appendix A: Report on 2009 Interim Meeting***

There were two items on the NCWM S&T Committee agenda related to our mission – Item 310-2 (definitions of software based devices) and Item 310-3 (marking requirements). The consensus was that they still need work, and they remain informational.

It seemed from the comments made during the open hearings that the membership didn't see a clear benefit to the field inspectors, and the scale manufacturers were also resistant to the change, fearing distinction between different types of devices would complicate marking, and additionally the SMA didn't see a difference between built-for-purpose and non-built-for-purpose.

In general, the feedback at the Interim gave the impression to Sector members that attended that we need to back up a little.

## ***Appendix B: Report on International WM Activity***

There's a new project regarding field verification, but there likely won't be activity this year.

There weren't too many changes to WELMEC 7.2. They are mainly clarifications. The current methodologies are now considered a bit too restrictive, so they're being reconsidered.

There has been an update to one of our referenced WELMEC documents since our last Software Sector meeting:

*Software Guide (Measurement Instruments Directive 2004/22/EC)* is now at Issue 4.

You can download an updated copy of this document at <http://www.welmec.org/publications/7-2.asp>

The changes are minor, including:

- Removal of the requirement that the NB maintain a file of the documentation and (if necessary) the software supplied for Type P & Type U submissions.
- Software Download extension has two additions, listed below in **blue** (underscored and bolded) below.

### **9 Extension D: Download of Legally Relevant Software**

This extension shall be used for the download of legally relevant software **as long as the metrological characteristics remain unchanged and the declaration of conformity is still valid**, (e.g., bug-fixes). These requirements are to be considered in addition to the basic requirements for Types P and Type U described in Chapters 4 and 5 in the guide.

#### **D2: Authentication of downloaded software**

*Means shall be employed to guarantee that the downloaded software is authentic, and to indicate that the downloaded software has been approved by an NB.*

#### **Specifying Notes:**

1. Before the downloaded software is used for the first time, the measuring instrument shall automatically check that:
  - a. The software is authentic (not a fraudulent simulation).
  - b. The software is approved for that type of measuring instrument.
2. The means by which the software identifies its NB approval status shall be made secure to prevent counterfeiting of the NB status.
3. If downloaded software fails any of the above tests, see D1.
4. **If a manufacturer intends to change or update the legally relevant software he shall announce the intended changes to the responsible notified body. The notified body decides whether an addition to the existing TEC is necessary or not. For software download it is indispensable that there is a software identification which is unambiguously assigned to the approved software version.**

***Appendix C: 2010 Software Sector Meeting Attendee List***

**William Arce**

USDA, GIPSA, PSP  
210 Walnut Street, Room 317  
Des Moines, IA 50309  
P. (515) 323-2510 F. (515) 323-2590  
E. william.arce-arana@usda.gov

**Dennis Beattie**

Measurement Canada  
400 St. Mary Avenue  
Winnipeg, Manitoba R3C 4K5  
Canada  
P. (204) 983-8910 F. (204) 983-5511  
E. dennis.beattie@ic.gc.ca

**Doug Bliss**

Mettler-Toledo, Inc.  
1150 Dearborn Drive  
Worthington, OH 43085  
P. (614) 438-4307 F. (614) 438-4355  
E. doug.bliss@mt.com

**Kevin Detert**

Avery Weigh-Tronix  
1000 Armstrong Drive  
Fairmont, MN 56031-1439  
P. (507) 238-4461 F. (507) 238-8255  
E. kdetert@awtglobal.com

**Dick Dirksen**

Vande Berg Scales  
770 7<sup>th</sup> St. NW  
Sioux Center, IA 51250  
P. (712) 722-1181

**Cassie Eigenmann**

DICKEY-john Corporation  
5200 Dickey-john Road  
Auburn, IL 62615  
P. (217) 438-2294 F. (217) 438-2635  
E. ceigenmann@dickey-john.com

**Michael Frailer**

Maryland Department of Agriculture  
50 Harry S. Truman Parkway  
Annapolis, MD 21224  
P. (410) 841-5790 F. (410) 841-2765  
E. fraileml@mda.state.md.us

**Andrew Gell**

FOSS North America, Inc.  
8091 Wallace Road  
Eden Prairie, MN 55344  
P. (952) 974-9892  
E. agell@fossnorthamerica.com

**Teri Gulke**

Liquid Controls  
105 Albrecht Drive  
Lake Bluff, IL 60044-2242  
P. (847) 283-8346 F. (847) 295-1170  
E. tgulke@idexcorp.com

**Norman Ingram**

CDFA / DMS  
6790 Florin Perkins Road, Suite 100  
Sacramento, CA 95828-1812  
P. (916) 229-3016 F. (916) 229-3015  
E. ningram@cdfa.ca.gov

**Kenneth Jones**

CDFA / DMS  
6790 Florin Perkins Road, Suite 100  
Sacramento, CA 95828-1812  
P. (916) 229-3052 F. (916) 229-3015  
E. kjones@cdfa.ca.gov

**Michael Kelley**

Ohio Department of Agriculture  
Division of Weights and Measures  
8995 East Main Street  
Reynoldsburg, OH 43068  
P. (614) 728-6290 F. (614) 728-6424  
E. mkelley@agri.ohio.gov

**Dan Parks**

California Division of Measurements  
Standards  
6790 Florin Perkins Road, Suite 100  
Sacramento, CA 95828  
P. (916) 229-3000 F. (916) 229-3015  
E. dparks@cdfa.ca.gov

**Paul A. Lewis, Sr.**

Rice Lake Weighing Systems, Inc.  
230 West Coleman Street  
Rice Lake, WI 54868-2404  
P. (715) 434-5322 F. (715) 234-6967  
E. plewis@ricelake.com

**James Pettinato**

FMC Technologies Measurement Solutions,  
Inc.  
1602 Wagner Avenue  
Erie, PA 16510  
P. (814) 898-5250 F. (814) 899-3414  
E. jim.pettinato@fmcti.com

**Joe Raspino**

California Division of Measurement  
Standards  
6790 Florin Perkins Road, Suite 100  
Sacramento, CA 95828  
P. (916) 229-3070 F. (916) 229-3015  
E. jraspino@cdfa.ca.gov

**Dan Reiswig**

California Division of Measurement  
Standards  
6790 Florin Perkins Road, Suite 100  
Sacramento, CA 95828  
P. (916) 229-3023 F. (916) 229-3015  
E. dreiswig@cdfa.ca.gov

**John Roach**

California Division of Measurement  
Standards  
Type Evaluation Program  
6790 Florin Perkins Road, Suite 100  
Sacramento, CA 95828  
P. (916) 229-3014 F. (916) 229-3015  
E. jroach@cdfa.ca.gov

**Mike Roach**

VeriFone  
4011 Barwood C  
Tampa, FL 33624  
P. (727) 953-4245 F. (727) 953-4001  
E. mike\_r4@verifone.com

**Brett Saum**

San Luis Obispo County Weights and  
Measures  
2156 Sierra Way, Suite A  
San Luis Obispo, CA 93401  
P. (805) 781-5922 F. (805) 781-1035  
E. bsaum@co.slo.ca.us

**Matthew Stevens**

California Division of Measurement  
Standards  
Type Evaluation Program  
6790 Florin Perkins Road, Suite 100  
Sacramento, CA 95828  
P. (916) 229-3018 F. (916) 229-3015  
E. mstevens@cdfa.ca.gov

**Scott Szurek**

Emerson Process Management  
1612 South 17th Avenue  
Marshalltown, IA 50158  
P. (641) 754-3425 F. (641) 754-3630  
E. scott.szurek@emerson.com

**Ambler Thompson**

NIST, Weights & Measures Division  
100 Bureau Drive, MS 2600  
Gaithersburg, MD 21701  
P. (301) 975-2333 F. (301) 975-8091  
E. ambler@nist.gov

**Van Thompson**

California Division of Measurement  
Standards  
6790 Florin Perkins Road, Suite 100  
Sacramento, CA 95828  
P. (916) 229-3025 F. (916) 229-3016  
E. vthompson@cdfa.ca.gov

**James Truex**

National Conference on Weights and  
Measures  
88 Carryback Drive  
Pataskala, OH 43062  
P. (740) 919-4350 F. (740) 919-4348  
E. jim.truex@ncwm.net

THIS PAGE INTENTIONALLY LEFT BLANK