

**National Type Evaluation Technical Committee
Software Sector Annual Meeting
March 11-12, 2009 Reynoldsburg, OH**

Meeting Summary

Carry-over Items:	2
1. Issuing Certificates of Conformance for Software.....	2
2. Definitions for Software Based Devices (2009 Interim Agenda Item 310-2).....	3
3. Marking of Software Identification – G-S.1 (2009 Interim Agenda item 310-3).....	5
4. Identification of Certified Software	8
5. Software Protection/Security	10
6. Software Maintenance and Reconfiguration.....	12
7. Verification in the Field, By the W&M Inspector.....	13
8. NTEP Application for software requiring a separate CC	14
New Items:	15
9. Sealing requirements for Electronic Devices.....	15
10. Next Meeting	15
<i>Appendix A: Attendee List.....</i>	<i>16</i>
<i>Appendix B: Draft Presentation for Regional Meetings</i>	<i>18</i>
<i>Appendix C: Draft Article for NCWM Spring Newsletter.....</i>	<i>19</i>

**National Type Evaluation Technical Committee
Software Sector Meeting
March 11-12, 2009 Reynoldsburg, OH**

Meeting Summary

Carry-over Items:

1. Issuing Certificates of Conformance for Software

Source: NCWM Reports

Background: Excerpts of reports from the 1995-1998 Executive Committees were provided to NTETC Software Sector members at their April 2006 meeting. The chair asked the sector to review the following NTEP policy decision adopted by the NCWM in 1998 relative to the issuance of a separate Certificate of Conformance (CC) for software. During the 1998 NCWM Annual Meeting, the following recommendation was adopted as NTEP policy:

- “Software, regardless of its form, shall not be subject to evaluation for the purpose of receiving a separate, software Certificate of Conformance from the National Type Evaluation Program.”
- “Remove all of the software categories from the index of NCWM Publication 5, NTEP Index of Device Evaluations.”
- “Reclassify all existing software CCs according to their applicable device categories.”

Also relevant, from Section C of NCWM Publication 14: “In general, type evaluations will be conducted on all equipment that affect the measurement process or the validity of the transaction (e.g. electronic cash registers interfaced with scales and service station consoles interfaced with retail fuel dispensers); and all equipment to the point of the first indicated or recorded representation of the final quantity on which the transaction will be based.”

Recommendation: The Sector recommended the following language to be submitted to the NTEP Committee as a policy change, and requested that the NTEP Committee place this issue on their agenda:

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 3rd party. The request to add software could be made by the original CC holder on behalf of the 3rd 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 it was determined that this item be given voting status for the 2009 Annual Meeting Agenda.

Discussion: Ambler Thompson observed that in reality this type of software represents only a small portion of type evaluations; the vast majority of them are not standalone software. Cassie Eigenmann indicated that this item as

written might not clearly state the intention; which is to simply allow the labs to call standalone software packages that are type approved to be categorized as ‘software’. It is an administrative change, not really a regulatory change. The labs will not be doing anything differently at type approval time.

Dennis Beattie made the statement that if you follow the concept of ‘first final’, then you have to address every step of the process (and if that is done with software, then the requirement to address software is obvious). David Vande Berg explained that it is not always black/white, i.e. external software for tare/net calculations is sometimes not judged subject to type approval. It was suggested by Norm Ingram that we should clearly define what we mean by ‘software requiring a separate CC’; Cassie Eigenmann recommended using specific examples.

Steve Patoray listed some goals he felt it was important the Sector accomplish:

- Answer the question “What is this item that is up for vote going to change in practice?”
- Address Scale Manufacturers Association’s concerns on the S&T agenda items (310-2 and 310-3).

Ambler Thompson agreed, further suggesting that the Sector needs to ‘sell’ the concepts we’ve realized; and it was mentioned that the Regional meetings might be an opportunity to approach the states.

Jim Truex (NTEP Administrator) felt that the upcoming vote will be a technical vote, requiring at least 27 states to vote in the affirmative to pass. He also indicated that this will not change the way the labs operate – it is merely the ability for the labs to label evaluated standalone software as such, and not be forced to categorize it as some type of device such as ‘weigh-in-weigh-out-system’. Steve Patoray also suggested that this is an important vote for the Sector; and asked that if the states continue to avoid dealing with software what is the future of the Sector?

Conclusions:

- **The Sector feels that this item is important and that there exists the possibility of misinterpretation of the scope/intent of this item by other interested parties, hence the Sector agreed to the following actions:**
 - Generate Problem Statement and specify benefits addressed by change (Done)
 - Feedback from labs/inspectors (Lucas, Frailer, Ingram?)
 - ‘Sales flyer’ / Newsletter article (Bliss et al)
 - Request added as Agenda item at CWMA/NEWMA? (Pettinato/Ingram)
 - Attend CWMA/NEWMA regional meetings? (?)

NCWM was contacted and the staff indicated that if it is desired to include an article in the newsletter, a final draft must be submitted by April 15th. The Sector work group should have a draft circulating by 4/3/09 so comments can be gathered by 4/10/09 for consideration prior to the final draft.

Doug Bliss provided a draft ‘slide show’ format presentation as a starting point for clearly presenting the ideas put forth by the Sector, and started on a draft article for the newsletter. Further work has progressed since the meeting (see Appendices B & C).

2. Definitions for Software Based Devices (2009 Interim Agenda Item 310-2)

Source: NTETC Software Sector

Background: Discussed was marking and G-S.1.1. It was initially suggested that "not built-for-purpose" be removed from the wording in NIST HB 44 G-S.1.1. However, after further discussion this may not be the correct or final decision. There is no definition for a ‘not built-for-purpose device’ in HB 44. The current HB 44 definition for a built-for-purpose device reads:

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)

The Sector recommended the following definitions be submitted to the S&T Committee as an item and be considered for inclusion in Appendix D of NIST Handbook 44 to replace the current definition of ‘build-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.

At the 2009 NCWM Interim Meeting, the Committee received comments from the Scale Manufacturers Association stating that it now opposes this item since there is no technological justification for making a distinction in software-based device types. Darrell Flocken added that the SMA can only provide limited responses; SMA continues to support the efforts of the Software Sector and the SMA response is based on the concern that the proposed definitions in this recommendation and the marking requirements proposed in agenda item 310-3 will require weighing devices be more complex than those currently produced.

The Meter Manufacturers Association indicated that it supports the item as written in the recommendation.

Will Wotthlie, Maryland, did not agree with the SMA position that there are no technological difference between the types of software-based devices. He added that Type P devices and separable elements have limited flexibility in changing software and indications and frequently include the sensing elements necessary for the measurement (e.g., load cells, meters, etc.). Whereas Type U devices and separable elements are typically devices that do not contain measuring elements; can be replaced with compatible equipment and display devices purchased from any number of sources; and only process metrological information received from measuring and other sensing elements.

Stephen Patoray, Consultants in Certification, agrees with the SMA that there are few differences between Type P and U software-based devices. However there are significant differences between Type P and U devices in that a Type P device is defined as an instrument that requires a security means since the instrument has fixed hardware (including sensing components) where the metrological software is *embedded* into the instrument. Type U devices do not include fixed components and metrological software can not be sealed using physical security seals or the minimum form of an audit trail (i.e., two event counters).

Software Sector Co-Chair Jim Pettinato (FMC Technologies) added that international recommendations recognize the differences between embedded software and programmable/loadable software. Additionally, the Software Sector recommends that this item remain informational to allow conference members to further study that proposed definitions.

The S&T Committee agreed with the comments received during the open hearing and the request from the co-chairman of the software sector and agreed that this item should remain an Informational item for further review.

Additional background information on this item can be reviewed in the 2009 Interim Agenda (NCWM Pub. 15).

Discussion: It was reiterated by several individuals that again it seems that resistance to this item stems not from a disagreement with the intention, but from either a misunderstanding of the applicability or unrelated concerns over marking requirements.

Further discussion was related to how to best present the opinion/goals of the Sector to the interested external parties, such as the NCWM standing committees and the individual states. Some discussion on the wording of the definitions took place as well, with the slightly modified version being proposed:

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

- (a) Type ‘P’ (aka built-for-purpose) software-based electronic devices. 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; or
- (b) Type ‘U’ (aka not-built-for-purpose) software-based electronic devices. All metrological software-based devices not meeting the conditions of a Type ‘P’ device. Example: a personal computer or other device and/or element with PC components with programmable or loadable metrological software.

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

Conclusion: No consensus was reached on any language change. The Sector did agree that including the reason(s) for proposing these definitions as part of the effort to educate/promote external parties would be beneficial; and that we would attempt to explain the reasoning/intent of the proposed definitions together with/as part of the action items for Item 1.

3. Marking of Software Identification – G-S.1 (2009 Interim Agenda item 310-3)

Source: 2008 Carryover Item

Background: Starting at the October 2007 meeting, the Software Sector has discussed the value and merits of required markings for software. After several iterations, the Sector developed a table to reflect their positions:

Method	NTEP CC No.	Make/Model/Serial No.	Software Version/Revision ¹
TYPE P electronic devices shall meet at least one of the methods in each column:			
Hard-Marked	X	X	Not Acceptable ¹
Continuously Displayed	X	X	X
By command or operator action	Not Acceptable	Not Acceptable	X ²
¹ If the manufacturer declares that the primary sensing element “software” is integral, has no end user interface and no print capability, the element may be considered exempt from the marking requirement for version/revision. the version/revision shall be hard marked on the device. Example: Primary sensing element may be Positive Displacement (P.D.) meter with integral correction, digital load cell (only for reference, not limiting). ² Information on how to obtain the Version/Revision shall be included on the NTEP CC. Metrologically significant software shall be clearly identified with the software version. The identification may consist of more than one part but one part shall be only dedicated for the metrologically significant portion.			
Method	NTEP CC No.	Make/Model/Serial No.	Software Version/Revision
TYPE U electronic devices shall meet at least one of the methods in each column:			
Hard-Marked	X ³	X	Not Acceptable
Continuously Displayed	X	X	X
Via Menu (display) or Print Option	Not Acceptable	X ⁴	X ⁴
³ Only if no means of displaying this information is available. ⁴ Information on how to obtain Make/Model, Version/Revision shall be included on the NTEP CC. Metrologically significant software shall be clearly identified with the software version. The identification may consist of more than one part but one part shall be only dedicated for the metrologically significant portion.			

This table was submitted to NCWM S&T Committee and was assigned Developing status in 2008.

Prior to the 2009 Interim NIST Weights/Measures Division commented on this item, and presented an alternate proposal with significant modifications, which were included in the Interim Meeting Agenda background for the item (See 2009 Pub 15 for more details).

This item was assigned Informational status for the NCWM 2009 Annual Meeting.

Discussion: It was noted by several Sector members that the perceived scope of the original proposal has been extended by the modifications made by WMD and now appears to exceed both the purview and the intent of the Sector and it has become difficult to discern what our intentions were. Based on the fact that the table seems to have actually made the Sector’s intent less clear, it was proposed by the chair to revisit this item in relation to the current text of G-S.1 to clarify exactly what real changes to Handbook 44 would be required to achieve the intent of the Sector. It was also noted that there was some validity to the Scale Manufacturers Association argument that there is no justification for differentiation of marking requirements based on device type (P or U). After additional lengthy discussions, the following modified versions of G-S.1/G-S.1.1 were drafted:

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)

It was 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.

John Roach (CA NTEP Lab) indicated that if the proposed table (or some version thereof) is not eventually included as part of G-S.1 that it may be useful to incorporate a suitable table into Pub 14.

Conclusion: **The Sector wishes to address concerns related specifically to software and does not wish to debate the merits of general marking requirements beyond that related to software identification. We feel the above proposed changes better reflect the Sector position. If WMD and NCWM S&T feel a table outlining general marking requirements would clarify the intent of G-S.1 then the Sector suggests that following simplified version may better suit the purpose.**

Table G-S.1.a Identification for Devices Manufactured on or after January 1, 201X		
<u>Required Marking</u>	<u>Full Mechanical Devices and Separable Mechanical Elements</u>	<u>Electronic Devices, Software Based</u>
<u>Manufacturer or CC holder ID</u>	<u>Hard Marked</u>	<u>Hard Marked, Continuously Displayed, or Via Menu (display) or by command or operator action</u>
<u>Model identification</u>	<u>Hard Marked</u>	<u>Hard Marked, Continuously Displayed, or Via Menu (display) or by command (operator action)</u>
<u>Serial number</u>	<u>Hard Marked</u>	<u>Hard Marked, Continuously Displayed ¹</u>
<u>Metrologically Significant Software version</u>	<u>Not Applicable</u>	<u>Continuously Displayed, Via Menu (display) or by command (operator action) ²</u>
<u>Certificate of Conformance number</u>	<u>Hard Marked</u>	<u>Hard Marked or Continuously Displayed, or Via Menu (display) or by command (operator action) ³</u>
<u>¹Type ‘U’ devices need not have a non-repetitive serial number.</u>		
<u>²If the manufacturer declares that the primary sensing element “software” is integral, has no end user interface and no print capability, the version/revision shall be hard marked on the device. Example: Primary sensing element may be Positive Displacement (P.D.) meter with integral correction, digital load cell (only for reference, not limiting).</u>		
<u>³If the Certificate of Conformance number is to be displayed via menu and/or submenu, the means of access must be easily recognizable. In addition, instructions on how to obtain the remaining required information not hard-marked or continuously displayed shall be included on the NTEP CC.</u>		

(Added 201X)

Note that this new version of the table reflects the aforementioned changes proposed for the G-S.1 text as well, homogenizing Type P and Type U requirements, with the exception of the serial number requirement being waived for standalone software. It was also noted that much of the information previously included in the separate proposed Table G-S.1b was redundant as it is already stated verbatim in the text of G-S.1; hence the Sector questions the benefit of the WMD - proposed separate Table G-S.1b.

4. Identification of Certified Software

Source: NTETC Software Sector

Background: 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 WELMEC and OIML). From WELMEC:

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:

Example from DSW 2CD (now 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 W&M inspector information that something has changed? (Yes, the Category III audit trail or other means of sealing). How can the W&M 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 CoC).

Recommendation: The Sector believes that we should work towards language that would include a requirement similar to the OIML requirement in HB44. 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?)

Identification of Certified Software:

Software-based electronic devices shall be designed such that the metrologically significant software is clearly identified. The identification of the software shall be inextricably linked to the software itself.

- **Unique identifier must be displayable/printable on command or during operation, etc. (marking req't in addition)**
- **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)**

Discussion: Discussion on this item was brief; it was the general consensus that those in attendance understood the goals of this item and were in agreement of those goals; however the conceptual language was not far enough along to warrant detailed discussion specific to a draft proposal and more work offline should be done.

Conclusion: A work group will be designated by the Sector Co-Chairs prior to the NCWM Annual Meeting to further promote the state of this item, to be discussed at the next Sector meeting.

5. Software Protection/Security

Source: NTETC Software Sector

Background: The sector agreed that Handbook 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.

Recommendation: 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.

Devices with embedded software TYPE P (aka built-for-purpose)		
	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 the (all) metrologically significant functions OIML states that there shall be no undocumented functions	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"/>
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)			
	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"/>
Software with closed shell (no access to the operating system and/or programs possible for the user)			
	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"/>
Operating system and / or program(s) accessible for the user:			
	Check whether a checksum or equivalent signature is generated over the machine code of the metrologically significant software (program module(s) subject to legal control W&M 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"/>
Software interface(s)			
	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"/>

Discussion: The Chair requested feedback from the NTEP Labs as to whether they had the opportunity to utilize the checklist; each lab reported either they have not had any applications for devices where the checklist could be used, or were unaware of the request to try the checklist. The labs were again asked to try to use the checklist should the opportunity present itself.

Conclusion: The Sector will again wait for laboratory feedback on this item; discussion on this item will continue as part of the next agenda item since the two are closely related.

6. Software Maintenance and Reconfiguration

Source: NTETC Software Sector

Background: The Following Items were reviewed by the Sector in previous meetings.

- 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 e.g. 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 e.g. 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 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 US W&M requirements.

Recommendation: 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 for the software update logger. In this case the existing requirement of 1,000 entries supersedes the 10 entry requirement. A software update log entry shall include the following:

- 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)

Discussion: 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.

Jim Truex 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. 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

Conclusion: The Sector feels 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.

(Note to S & T – this item assumes additional requirements in individual codes will be eventually added to address this requirement; e.g. L.M.D. code has philosophy of sealing section that could be enhanced to include processes described.)

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

Source: NTETC Software Sector

Background: What tools does the field inspector need as relates to software-based electronic devices? Some possible answers:

NTEP CC – hard marked, continuously displayed, via menu command or operator action

Clear and simple instructions on NTEP CC to get to the other Inspection Information

The metrologically significant software identifier needs to be easily accessible from operator console

Clear and simple instructions on NTEP CC to access audit trail(s)

Recommendation: The Sector needs to continue to develop this item.

Discussion: 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 CoC.)

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?) 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.

Conclusion: The Sector will continue to develop this item, and initiate a new agenda item specific to inspector training in relation to evaluating/validating software-based devices.

8. NTEP Application for software requiring a separate CC

Source: NTETC Software Sector

Background: This item had been on the agenda of previous meetings, but was not discussed due to time limitations.

Recommendation: Identify issues, requirements and processes for type approving type U device applications.

Discussion: 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/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.

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

New Items:

9. Sealing requirements for Electronic Devices

Source: Weighing Sector Tech Advisor

Background: Steve Cook of NIST has been involved in attempting to address some concerns with the current wording of G-S.8 as it relates to the sealing of electronic devices and configuration modes. Since this is related in some respects to other items within the purview of the Software Sector, it was suggested that it may be beneficial for the Sector to review and comment on the proposed language.

Discussion: The Sector discussed the relevance of this item, and though it is related somewhat to our discussions on software security and maintenance/reconfiguration, it is broader in scope and hence it was decided that the item was not wholly relevant to the Sector's mission.

Conclusion: **The Software Sector takes no position on these proposed changes.**

10. Next Meeting

Recommendation: The Sector was asked to develop a proposed date and location for the next meeting.

Discussion: The Sector discussed two options for the next meeting; continuing to meet in Ohio or alternating to a Western location to maintain equity in travel for the various participating labs. There appeared to be a preference (after an informal polling) to alternate the meeting location from year to year.

Conclusion: **The Sector recommends that the next meeting be held in Sacramento in or around March 2010. Sector Co-Chair Norm Ingram will investigate suitable hotels and meeting facilities and report back to NCWM. Details need to be firmed up by December of this year.**

Appendix A – 2009 Software Sector Attendees



2009 Software Sector Meeting Attendee List March 11-12, 2009 / Reynoldsburg, Ohio

William Arce

USDA/Packers & Stockyards
210 Walnut Street, Suite 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 Ave
Winnipeg, Manitoba R3C 4K5
P. (204)983-8910 F. (204)983-5511
E. dennis.beattie@ic.gc.ca

Doug Bliss

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

Cathy Brenner

USDA GIPSA FGIS
10383 North Ambassador Drive
Kansas City, MO 64153-1394
P. (816)891-0486 F. (816)891-8070
E. cathleen.a.brenner@usda.gov

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

Darrell Flocken

Mettler-Toledo, Inc.
1150 Dearborn Drive
Worthington, OH 43085
P. (614)438-4393 F. (614)438-4355
E. darrell.flocken@mt.com

Michael Frailer

Maryland Department of Agriculture
50 Harry S. Truman Parkway
Annapolis, MD 21401
P. (410)841-5790 F. (410)841-2765
E. fraileml@mda.state.md.us

Travis Gibson

Rice Lake Weighing Systems, Inc.
230 West Coleman Street
Rice Lake, WI 54868
P. (715)234-3494 F. (715)234-6967
E. tragib@rlws.com

Bob Helwig

Vande Berg Scales
770 7th Street NW
Sioux Center, IA 51250
P. (712)722-1181 F. (712)722-0900
E. bhelwig@vbssys.com

Norman Ingram

California Division of Measurement Standards
6790 Florin Perkins Road, Suite 100
Sacramento, CA 95828
P. (916)229-3016 F. (916)229-3015
E. ningram@cdfa.ca.gov

Paul Lewis

Rice Lake Weighing Systems, Inc.
230 West Coleman Street
Rice Lake, WI 54868
P. (715)234-9171 F. (715)234-6967
E. plewis@ricelake.com

Todd Lucas

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

Ken Mahaney

Vande Berg Scales
770 7th Street NW
Sioux Center, IA 51250
P. (712)722-1181 F. (712)722-0900
E. KMahaney@vbssys.com

Steve Patoray

Consultants on Certification, LLC
1239 Carolina Drive
Tryon, NC 28782
P. (828)859-6178 F. (828)859-6180
E. steve@consultoncert.com



2009 Software Sector Meeting Attendee List (cont.) March 11-12, 2009 / Reynoldsburg, Ohio

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

John Roach

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

Mark Schwartz

Accu-Sort Systems, Inc.
511 School House Road
Telford, PA 18969
P. (215)721-5053 F. (215)721-5557
E. mark.schwartz@accusort.com

Ambler Thompson

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

James Truex

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

David Vande Berg

Vande Berg Scales
770 7th St NW
Sioux Center, IA 51250-1918
P. (712)722-1181 F. (712)722-0900
E. davevb@vbssys.com

Appendix B – Slide Show (Draft) for Presentation at Regional Meetings

<h3 style="text-align: center;">Software COC</h3> <p style="text-align: center;">What is it and why do we need it?</p>	<h3 style="text-align: center;">Why? What's Broken?</h3> <ul style="list-style-type: none"> • Software that runs on a PC may execute metrological functions <ul style="list-style-type: none"> – Display indication – Tare manipulation – Price computation – Receipt printing • PC based software is often difficult to <ul style="list-style-type: none"> – Identify – Verify – Protect
<h3 style="text-align: center;">First Final</h3> <ul style="list-style-type: none"> • Refer to first final requirement here (Pub 14 admin policy) <div style="text-align: center; margin-top: 20px;"> </div>	<h3 style="text-align: center;">PC-based Software Examples</h3> <ul style="list-style-type: none"> • Point of Sale Cash Register • Gas Station Pump Control • Vehicle Scale In-Out
<h3 style="text-align: center;">Point of Conflict?</h3> <ul style="list-style-type: none"> • Current NTEP policy states that software shall not be separately evaluated and given a CoC • It could be interpreted that Type Evaluation of the example systems is in conflict with the above rule. <ul style="list-style-type: none"> – No hardware was evaluated in these 	<h3 style="text-align: center;">What Software is NOT Affected?</h3> <ul style="list-style-type: none"> • Software that executes confined within purpose-built hardware is generally not an issue <ul style="list-style-type: none"> – Hardware provides a ready place to mark for identification – Software is not easily modified (by design) – Physical seal is often an option