# National Type Evaluation Technical Committee (NTETC)<br>Software Sector Meeting Agenda

March 19-20, 2013 / Columbus, Ohio

## INTRODUCTION

The charge of the National Type Evaluation Technical Committee (NTETC) Software Sector is important in providing appropriate type evaluation criteria for software-based weighing or measuring device based on specifications, tolerances and technical requirements of *NIST Handbook 44* Section 1.10 General Code, Section 2 for weighing devices, Section 3 for liquid and vapor measuring devices, and Section 5 for taximeters, grain analyzers, and multiple dimension measuring devices. The sector's recommendations are presented to the National Type Evaluation Program (NTEP) Committee each January for approval and inclusion in *NCWM Publication 14 Technical Policy, Checklists, and Test Procedures* for national type evaluation.

The sector is also called upon occasionally for technical expertise in addressing difficult *NIST Handbook 44* issues on the agenda of the National Conference on Weights and Measures (NCWM) Specifications and Tolerances (S&T) Committee. Sector membership includes industry, NTEP laboratory representatives, technical advisors and the NTEP Administrator. Meetings are held annually, or as needed and are open to all NCWM members and other registered parties.

Suggested revisions are shown in **bold face print** by **~~striking out~~** information to be deleted and **underlining** information to be added. Requirements that are proposed to be nonretroactive are printed in ***bold faced italics***.

**Table A**
**Table of Contents**

**Title of Content** **Page**

## Table B
## Glossary of Acronyms and Terms

| Acronym | Term | Acronym | Term |
|---|---|---|---|
| BIML | International Bureau of Legal Metrology | OIML | International Organization of Legal Metrology |
| CC | Certificate of Conformance | OWM | Office of Weights and Measures |
| EPO | Examination Procedure Outline | PDC | Professional Development Committee |
| GMMs | Grain Moisture Meters | PDC | Professional Development Committee |
| NCWM | National Conference on Weights and Measures | S&T | Specifications and Tolerances Committee |
| NTEP | National Type Evaluation Program | SMA | Scale Manufactures Association |
| NTETC | National Type Evaluation Technical Committee | WELMEC | European Cooperation in Legal Metrology |

## Details of All Items

*(In order by Reference Key)*

## SCHEDULE

*Note: topic times are approximate and merely included as a rough guideline to aid in maintaining meeting pace; some issues will invariably involve more detailed discussion than others.*

| **Tuesday, March 19, 2013** | | |
|---|---|---|
| **8:00 a.m.** | **Meeting Call to Order** | **Co-Chairs** |
| | Welcome / Introductions | |
| **8:30 a.m.** | **Status Reports** | |
| | 8. 2012 NCWM Interim Meeting Report | Interim Attendees |
| | 9. 2012 International Activity Report | A. Thompson, NIST, OWM |
| **9:00 a.m.** | **Work session – Carryover Items** | |
| | 1. Software Identification / Markings | |
| **10:00 a.m.** | **Break (15 minutes)** | |
| **10:15 a.m.** | **Carryover Items (continued)** | |
| | 1. Software Identification / Markings (continued) | |
| **12:00 p.m.** | **Lunch Break (1 hour)** | |
| **1:00 p.m.** | **Carryover Items (continued)** | |
| | 2. Identification of Certified Software | |
| **3:00 p.m.** | **Break (15 minutes)** | |
| **3:15 p.m.** | **Carryover Items (continued)** | |
| | 3. Software Protection / Security | |
| **5:00 p.m.** | **Adjourn** | |
| | | |
| **Wednesday March 20, 2013** | | |
| **8:00 a.m.** | **Continue Work Session – Carryover Items** | |
| | 4. Software Maintenance and Reconfiguration | |
| **10:00 a.m.** | **Break (15 minutes)** | |
| **10:15 a.m.** | **Carryover Items (continued)** | |

| | | |
|---|---|---|
| | 5. NTEP Application for Software and Software-Based Devices | |
| | 6. Training of Field Inspectors | |
| **12:00 p.m.** | **Lunch Break (1 hour)** | |
| **1:00 p.m.** | **Work Session – New Items** | |
| | 7. Next Meeting | |
| **3:00 p.m.** | **Break (15 minutes)** | |
| **3:15 p.m.** | **Work Session** | |
| | This time is reserved for revisiting items requiring additional attention and any unscheduled items brought to the sector for consideration. | |
| **5:00 p.m.** | **Adjourn** | |

## WELCOME / INTRODUCTIONS

The Chair would like to welcome new individuals that have joined the NTETC Software Sector since the last meeting. Please welcome:

- Eric Morabito, New York Bureau of Weights & Measures
- Gary Benjamin, NCR Corporation

## STATUS REPORTS – RELATED NCWM AND INTERNATIONAL ACTIVITY

Attendees of the 2013 NCWM Interim Meeting will be asked to share any relevant comments or discussion that took place during the open hearings or NCWM Standards and Tolerances (S&T) committee working sessions.

Dr. Ambler Thompson, NIST, Office of Weights and Measures (OWM), will provide a synopsis of international activity that relates to the work of the sector.

## CARRY-OVER ITEMS

### 1. Software Identification / Markings

**Source:**
NTETC Software Sector

**Background / Discussion:**
Since its inception the sector has wrestled with the issue of software identification and marking requirements. *See the 2012 Software Sector Meeting Summary and the 2013 Interim Meeting S&T Agenda Item 360-2 for more background on this item.*

NIST, OWM had been adding items to the S&T Agendas that confused matters since the perception was that this sector had contributed to this input. Most of the confusion arose in the 1990’s, due to some items being approved, and others, such as the definitions for “Built for Purpose” and “Not Built for Purpose,” not being approved.

Mr. Truex, NTEP Administrator, discussed the difficulty there has been in coming to a consensus on these issues with a representative of the NTEP Committee. Suggestions from NTEP to come to some resolution has been to write an article for the newsletter (which Mr. Bliss, Mettler-Toledo, LLC, had already done, to no effect), sending a questionnaire to the NTEP community, asking what they'd like to see, and sending a representative from this sector to the S&T Committee.

Mr. Roach, California Division of Measurement Standards, is concerned that some people may want to interpret G-S.1.c as requiring a serial number for software. Mr. Lewis, Rice Lake Weighing Systems, Inc. pointed out that the computer that the software was running on could have the serial number, not the software itself. That shouldn't matter, regardless.

Mr. Bliss, Mettler-Toledo, LLC, pointed out that the terminology in G-S.1. "All equipment", could be interpreted to mean that it doesn't apply to software. It was proposed that G-S.1.c be amended to add "and software". Mr. Bliss suggested submitting a document explaining the reasoning behind the proposed changes, rather than assume that the text is self-explanatory. Making a presentation to the various committees on the subject in addition would be beneficial as well. If a document is written, perhaps the examples given in G-S.1.d.3.a can be eliminated. "Metrologically significant" isn't explicitly defined, but it's been used since time immemorial.

Attempts to modify G-S.1.1. have been controversial, both in this meeting and in other committees. Unfortunately, there has been little constructive feedback from the other committees. It would probably be easier to incorporate specific examples given in G-S.1.1.b.3 in *NCWM Publication 14*. After some discussion, the previously proposed language was modified slightly to address some of the concerns received via feedback from other sectors and interested parties:

*NIST Handbook 44 – Proposed changes:*

**G-S.1. Identification.** – All equipment, except weights and separate parts necessary to the measurement process but not having any metrological effect**,** 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 devices~~ **software**;
[Nonretroactive as of January 1, 1968]
(Amended 2003)

*(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 20XX)**

*(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 accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an exception, permanently marking the version or revision identifier shall be acceptable under the following conditions:***

***(a) The user interface does not have any control capability to activate the indication of the version or revision identifier on the display, or the display does not technically allow the version or revision identifier to be shown (analog indicating device or electromechanical counter) or***

***(b) the device does not have an interface to communicate the version or revision identifier.***

(e) an NTEP CC number or a corresponding CC Addendum Number for devices that have a CC.

*(1) 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 of Marking Information for ~~Not Built For Purpose~~ all Software-Based Devices.*** *—For ~~**not built for purpose,**~~ software-based devices, 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 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 20XX**)

The new language in G-S.1.1 reflects that the sector reached consensus on the following positions:

- The software version/revision should (with very few exceptions – see D-31 5.1.1) be accessible via the user interface.
- The means by which the software version is accessed must be described in the Certificate of Conformance (CC).

The sector promoted this item following the meeting via several means to try and address the concerns of other interested parties. A presentation was generated and shared with the S.M.A. at their meeting. The regions had access to this information, as it was posted on the NCWM website. Unfortunately, based on the comments in the 2013 Pub 15 item 360-2, some regions were not aware that this information had been provided.

During the 2013 NCWM Interim Meeting, no comments were received relative to this item during the Open Hearings. In considering the item, the Committee questioned whether or not the Software Sector was still actively working the item. It was reported that the Software Sector believed they had developed the item as much as possible, yet the different stakeholders affected by the proposal could not agree on the changes that the Sector had proposed. Based upon that update, the Committee agreed to add to its report a request that the Software Sector work with the Weighing Sector and Measuring Sector to identify which portions of the proposal need to be modified in order that they might be accepted by the entire community. The Committee acknowledges and appreciates the efforts of the Software Sector and looks forward to being able to consider a proposal that addresses both the identification of software and how it may be accessed.

**Conclusion:**
The item needs finalizing by the Sector. We may want to consider more direct methods, i.e. designating a representative to address the regional groups or other sectorsat their meetings. The annual meeting may be an appropriate venue for a presentation.

In addition, it was noted that it may be desirable to evaluate options that would lead to fully eliminating GS-1.1

## 2. 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 WELMEC and 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 inspector information that something has changed?**
Yes, the Category III Audit Trail or other means of sealing.

**How can the weights and measures 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 International Organization of Legal Metrology (OIML) requirement in *NIST Handbook 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?)*

*NIST 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.**

*From NCWM Publication 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 *NIST Handbook 44's* marking requirements.

In 2010, the sector recommended the following change to *NIST Handbook 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 20XX)**
>
> *(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 20XX)**

Also the sector recommends the following information be added to *NCWM Publication 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)

There was some additional discussion on this item regarding where this new requirement was best located. It was suggested that the first sentence of G-S.1.d.(3) could be added as a clause to the base paragraph G-S.1(d) text, e.g. "*the current software version or revision identifier for* ~~**not built for purpose**~~ ***software-based*** *devices, which shall be directly and inseparably linked to the software itself;*" .

It also was suggested that the second sentence in G-S.1.d. (3) might be more suitable for *NCWM Publication 14*, as it describes more ”how” than ”what” the requirement entails.

In addition, the sector considered the following information to be added to *NCWM Publication 14* as explanation/examples:

- The current software identifier must be displayable/printable on command during operation (or made evident by other means deemed acceptable by G-S.1.)
- At a minimum, the software identifier must include a version/revision indication (1.02.09, rev 3.0 a, etc). It could also consist of / contain checksum, etc (crc32, for example).
- 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.

Other questions that are still outstanding:

- 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?
- If a device is capable of doing so, does it still have to be able to display, print or communicate the identifier somehow, even if it is hard-marked?

**Conclusion:**

At the 2012 NTETC Software Sector Meeting, there was some discussion as to where the terminology regarding inextricably linking the software version or revision to the software itself belonged. At the moment, it is not incorporated in the proposed text for G-S.1. *NCWM Publication 14* may be a better option for the time being. This would be another item that would benefit from further explanation in a supplementary document.

One suggestion was this revision to G-S.1.d:

(d) ~~when metrologically significant software is employed,~~ the current software version or revision identifier, which shall be directly and inseparably linked to the software itself;~~, for not-built-for-purpose software-based electronic devices;~~

Alternatively, if the previously proposed new subsection G-S.1.d.3 from Item 1 is adopted, this concept could be inserted thus:

> *(3)* *The version or revision identifier shall be* directly and inseparably linked to the software itself and *accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an exception, permanently marking the version or revision identifier shall be acceptable under the following conditions:*

Several sector members were of the opinion that attempting to make this change at the same time as the earlier changes might be a difficult sell. Mr. Truex, NTEP Administrator, reiterated the necessity of baby steps.

**Conclusion:**

The sector recommended adding the following to *NCWM Publication 14* and forward to NTETC Weighing, Measuring, Grain Analyzer sectors for feedback:

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.

Also, it was decided to forward the two alternate options for adding requirements for uniquely identifying software to the individual sectors:

One suggestion was this revision to G-S.1.d:

(d) ~~when metrologically significant software is employed,~~ the current software version or revision identifier, which shall be directly and inseparably linked to the software itself;~~, for not-built-for-purpose software-based electronic devices;~~

Alternatively, if the previously proposed new subsection G-S.1.d.3 from Item 1 is adopted, this concept could be inserted thus:

*(3)* *The version or revision identifier shall be* **directly and inseparably linked to the software itself and** *accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an exception, permanently marking the version or revision identifier shall be acceptable under the following conditions:*

Both alternatives will be sent to the sectors for feedback.

## 3. Software Protection / Security

**Source:**
NTETC Software Sector

**Background / Discussion:**
The sector agreed that *NIST Handbook 44* already has audit trail and physical seal, but these may need to be enhanced.

*From the WELMEC Document:*

> **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 consideration of:
>
> 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 continued to develop a proposed checklist for *NCWM Publication 14*. The numbering will still need to be added. This is based roughly on R 76 – 2 checklist and discussions beginning as early as the October 2007 NTETC Software Sector Meeting. The information requested by this checklist is currently voluntary, however, it is recommended that applicants comply with these requests or provide specific information as to why they may not be able to comply. Based on this information, the checklist may be amended to better fit with NTEP's need for information and the applicant's ability to comply.

The California, Maryland and Ohio laboratories agreed to use this check list on one of the next devices they have in the lab and report back to the sector on what the problems may be. In February 2011, the North Carolina laboratory was also given a copy of the check list to try.

## 1. Devices with Embedded Software TYPE P (aka built-for-purpose)

1.1. Declaration of the manufacturer that the software is used in a fixed hardware and software environment. **AND** □ Yes □ No □ N/A

1.2. Cannot be modified or uploaded by any means after securing/verification. □ Yes □ No □ N/A

*Note: It is acceptable to break the "seal" and load new software, audit trail is also a sufficient seal.*

1.3. The software documentation contains:

1.3.1. Description of all functions, designating those that are considered metrologically significant. □ Yes □ No □ N/A

1.3.2. Description of the securing means (evidence of an intervention). □ Yes □ No □ N/A

1.3.3. Software Identification □ Yes □ No □ N/A

1.3.4. Description how to check the actual software identification. □ Yes □ No □ N/A

1.4. The software identification is:

1.4.1. Clearly assigned to the metrologically significant software and functions. □ Yes □ No □ N/A

1.4.2. Provided by the device as documented. □ Yes □ No □ N/A

## 2. 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)

2.1. The metrologically significant software is:

2.1.1. Documented with all relevant (see below for list of documents) information. □ Yes □ No □ N/A

2.1.2. Protected against accidental or intentional changes. □ Yes □ No □ N/A

2.2. 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 □ No □ N/A

## 3. Software with Closed Shell (no access to the operating system and/or programs possible for the user)

3.1. 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 □ No □ N/A

3.2. Check whether the manufacturer has submitted a written declaration of the completeness of the set of commands. □ Yes □ No □ N/A

## 4. Operating System and / or Program(s) Accessible for the User

4.1. 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 Weights and Measures jurisdiction and type-specific parameters). □ Yes □ No □ N/A

4.2. 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 □ No □ N/A

## 5. Software Interface(s)

5.1. Verify the manufacturer has documented:

5.1.1. The program modules of the metrologically significant software are defined and separated. □ Yes □ No □ N/A

5.1.2. The protective software interface itself is part of the metrologically significant software.

5.1.3. The functions of the metrologically significant software that can be accessed via the protective software interface. □ Yes □ No □ N/A

5.1.4. The parameters that may be exchanged via the protective software interface are defined. □ Yes □ No □ N/A

5.1.5. The description of the functions and parameters are conclusive and complete. □ Yes □ No □ N/A

5.1.6. There are software interface instructions for the third party (external) application programmer. □ Yes □ No □ N/A

The Maryland laboratory had particular questions regarding 3.1 and 5.1. The information for 3.1 could be acquired from an operator's manual, a training video, or in-person training. The items in 5.1 were confusing to the evaluators. The terminology is familiar to software developers, but not necessarily others. It was indicated that manufacturers were typically quick to return the filled out questionnaire, but he didn't know how his laboratory was supposed to verify that it was true. Generally, the laboratories wouldn't be expected to verify things to that level. For example, if the manufacturer states that a checksum is used to ensure integrity, the laboratories wouldn't be expected to evaluate the algorithm used.

The intent was to see whether the manufacturer had at least considered these issues, not for evaluators to become software engineers. Perhaps a glossary or descriptive paragraphs might be added to assist the evaluators for if the manufacturer has questions for the evaluators.

OIML makes use of supplementary documents to explain the checklist they use. Below are links:
http://www.oiml.org/publications/D/D031-e08.pdf
http://www.welmec.org/latest/guides/72.html
http://www.welmec.org/fileadmin/user_files/publications/2-3.pdf

WELMEC document 2.3 is the original source for our checklist, but it's been significantly revised and simplified. Mr. Payne, Maryland Department of Agriculture, is going to review the other documents and come up with some suggestions for the checklist. Mr. Roach, California Division of Measurement Standards, is going to begin using the checklist. The international viewpoint is that any device running an operating system is considered to be Type U. Mr. Roach mentioned that they're having lots of problems with "skimmers" stealing PIN's. Is there some way they can detect this?

Mr. Lewis, Rice Lake Weighing Systems, Inc., mentioned that he liked Measurement Canada's website. When answering similar questions, different pages would appear, based on answers to those questions: http://www.ic.gc.ca/eic/site/mc-mc.nsf/eng/lm00573.html

At the 2011 NTETC Software Sector Meeting, the laboratories were polled to obtain any feedback on the use of the checklist. Maryland attempted to use this checklist a few times. They had some difficulty obtaining answers from the manufacturers because the individual(s) interacting with the Maryland evaluator didn't always have the required information on hand. More experience in using the checklist will help determine what needs to be revised.

It was suggested that the checklist could be sent to manufacturers for their feedback as well, with the stipulation that it a completely voluntary exercise and purely informational at this point. The laboratories will coordinate with willing manufacturers to obtain feedback.

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

The checklist has been reviewed with an eye to making its terminology clearer to laboratories. Some examples and clarifications have been added as shown in the discussion section of this item. The revised checklist will be distributed to the laboratories for additional review. Maryland and California laboratories agreed to use the checklist on a trial basis.

## 4. Software Maintenance and Reconfiguration

**Source:**
NTETC Software Sector

**Background / Discussion:**
After the software is completed, what do the manufacturers use to secure their software? The following items were reviewed by the sector. *Note that agenda Item 3 also contains information on Verified and Traced updates and Software Log.*

1. Verify that the update process is documented (OK)
2. 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 **<u>or become inoperative</u>**.

   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 **<u>or become inoperative</u>**.

   Examples are not limiting or exclusive.

3. Verify that the sealing requirements are met

   The sector asked, What sealing requirements are we talking about?

   This item is **<u>only</u>** addressing the **<u>software update</u>**; 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

4. 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 US weights and measures 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.

*Note: It's possible that the Philosophy of Sealing section of NCWM Publication 14 may already address the above IF the definitions of Verified and Traced Updates (and the statement below) were to be added. The contrary argument was that it may be better to be explicit).*

> **Use of a Category 3 audit trail is required for a Traced Update. A log entry representing a traced software update shall include the software identification of the newly installed version.**

The sector recommended consolidating the definitions with the above statement thus:

> **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**~~ Category 3 audit trail. The audit trail entry shall include the software identification of the newly installed version.

**Conclusion:**
The sector recommended that as a first step, the following be added to *NCWM Publication 14*:

> **The updating of metrologically significant software, including software that checks the authenticity and integrity of the updates, shall be considered a sealable event.**

Mr. Truex, NTEP Administrator, believes the above sentence is unnecessary since it's self-evident. It was agreed to ask the other sectors for feedback on the value of this addition.

Though the sector is currently recommending only the single sentence be incorporated into *NCWM Publication 14* for the time being, ultimately, the sector may wish to advance the remaining language of the original item submission.

## 5. NTEP Application for Software and Software-based 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. What gets submitted? What requirements and mechanisms for submission should be available? Validation in the laboratories - all required subsystems shall be included to be able to simulate the system as installed.

Mr. Roach, California Division of Measurement Standards, 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 laboratories and scale manufacturers indicated that this is not usually done for scale evaluations.

Since the NTEP Committee passed the related item at NCWM Annual Meeting we will continue to work on this. Mr. Truex, NTEP Administrator, 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.

Comments given at the meeting indicate that current practice does not require anything different for software / software based devices compared to any other type approval. It was also noted that for international applications, OIML D-31.6.5 states, “The approval applicant is responsible for the provision of all the required equipment and components.” This would likely also be the policy of NTEP.

Since the checklist is still being tried out by some of the laboratories, the sector is not quite ready to develop this fully. Some documentation that eventually might be required by applicants could include (from WELMEC doc. 7-2 Issue 4):

- A description of the software functions that are metrologically significant, meaning of the data, etc.
- A description of the accuracy of the measuring algorithms (e.g. price calculation and rounding algorithms).
- A description of the user interface, menus, and dialogs.
- The software identification (version, revision, etc.) and how to view it.
- An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc, if not described in the operating manual.
- An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.
- The operating manual.

**Conclusion:**
These documentation requirements will be considered as input for requirements that will eventually appear in *NCWM Publication 14* and the application paperwork. Further work by the sector to develop the *NCWM Publication 14* requirements is needed, after more input from the labs is gathered.

## 6. Training of Field Inspectors

**Source:**
NTETC Software Sector

**Background / Discussion:**
During discussions at the 2009 NTETC Software Sector 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.

California has an Examination Procedure Outline (EPO) that begins to address this. Use *California 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 *NIST Handbook 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 15d while 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 more than 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**

9. Motion detection.
9.1. 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)
10. Over capacity.
10.1. 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:**
Mr. Jordan, California Division of Measurement Standards, is already doing something similar, and he may be able to assist. Mr. Roach, California Division of Measurement Standards, will talk to him to see whether they're available. In addition Mr. Parks, California Division of Measurement Standards, is based in Sacramento and a potential resource. If the meeting is held in Sacramento next year, they may be able to attend.

Mr. Truex, NTEP Administrator, pointed out that the PDC would also be a valuable resource on this subject. Mr. Pettinato, Co-Chair, will contact them.

**NIST Handbook 112- Examination Procedure Outline for Commercial Weighing and Measuring Devices.*

## NEW ITEMS

## 7. Next Meeting

**Background:**
The sector is on a yearly schedule for NTETC Software Sector Meetings. Mr. Truex, NTEP Administrator, will determine when the next meeting is possible. This year was California's turn in the rotation to host the meeting, but due to the uncertainly of New York's status as potential host, the meeting ended up being back in Ohio. Hence, New York and California again are possible locations for the 2014 meeting.

## 8. 2013 NCWM Interim Meeting Report

There was one item on the NCWM S&T Committee Agenda for the 2012 NCWM Interim Meeting related to work done by the NTETC Software Sector. *2012 Publication 15* S&T Item 360-2 relates to the 2012 NTETC Software Sector Agenda Item 1: Marking Requirements.

## 9. 2013 International Report

Dr. Ambler Thompson, NIST, Office of Weights and Measures (OWM), will provide a synopsis of international activity that relates to the work of the sector. Software Sector Co-Chair Mr. Jim Pettinato will summarize the discussion that took place at the European Cooperation in Legal Metrology (WELMEC) WG7 meeting in Dec. 2011.

Highlights of interest to the NTETC Software Sector:

- New WELMEC 7.2 draft document circulated for comment by WG7
- R-117 working group