

5 Testing style guide¹

5.0 Re-implementation of the testing tool.

The testing tool has been re-implemented. The style of the tests remain the same, and questions being in formatted text files and student responses being handled by a separate answer box, but the 'look and feel' of the answer box is being completely changed with the answer box for only one question being visible at a time (in default mode) The new version is functionally compatible with the old but the format of the control file has been changed However, a control file editor has been prepared that will automatically convert data from the old format to the new.

5.1 Changes

Changes between version 4 and version 4.2

1. The answer sheet and the control file syntax have been extended to include questions with numeric answers.
2. Multiple questions files are now supported. The number of files is specified on the command line together with the style of test.
 - If self-test (formative), then the user selects the question file
 - If assessed (summative), then a question file is randomly selected.

Changes between version 3 and version 4

1. No changes have been made to the specification of the Control (.txt) file. All existing tests will still work
2. The Microcosm linking has been extensively modified and is described below. The essential difference is that the RTF viewer is now launched 'automatically' by an 'autodispatch' message to Microcosm. There is no longer a need for a Generic link from the Test Tool to the RTF Question File. A small (optional, but a good idea) addition to the .MCS file allows the sttest.exe to be placed in mcm\bin and eliminated from the quiz directory.
3. The Answer Sheet has the rubric. The rubric can be removed from the top of the Question File.
4. There have been a number of small changes to the appearance of the Answer Sheet.
5. The feedback window is no longer the standard VB Message Box but more tailored to our needs. On its first appearance, the Feedback window is always positioned below the Answer Sheet. If the Feedback Window is moved, the new position is retained.
6. There have been changes to the Explain process.
7. When a question is correct, there is visible feedback that the student has been successful. The student can make several attempts to get a question correct.
8. The user is always prompted to verify an Exit from the test.

For all this to work it is essential to have the new version of **mcmmsg.dll** which re-implements Autodispatch of Microcosm documents. Use a version dated 6/13/94 or later. If you get a

¹ Version 4.0. October 1994. Dick Bacon, Steve Rake.

message about being unable to find the function QD_FileGot then you are not running with the latest version of **mcmmsg.dll**.

5.2 Student assessment in SToMP.

Where this style guide refers to the testing package, it is referring to the new version released with version 3.00. This can log student responses and handles four styles of multiple choice question, a free text and a free numeric response and can put random numeric values into the questions..

Three schemes for testing students are to be used, two providing feedback and the third for use in summative testing. In summative, the SToMP resource material will be available but there will be no direct support linking (i.e. linking to the SToMP/Microcosm resources) as in the other two versions.

5.3 In line testing

The first scheme is for questions to be put directly into a script, with some possible solutions being part of the question text (or immediately following). These possible solutions will be MCM buttons linking to support material as required. This will be called **in line testing**.

Both the second and third schemes will use the **testing package**, one with and one without active linking to the rest of the SToMP material.

5.4 The testing package.

The testing package has an Answer Window to handle student responses, and uses the rtf viewer to present questions (authored in Word). Each Question (**.rtf**) file has one or two associated text control files (**.qni** for all types of test, plus an **.ani** for assessed tests) that contain information to set up and control the Answer Sheet (e.g. the number of questions, the number of choices for each question, etc.) and the marking of the questions. Students will select their responses by mouse or keyboard action or by typing text into text boxes.

For *self tests* the **qni** file contains the set up information for the answer window and the answers with feedback messages.

For *assessed tests* the set up information is in the **qni** file but the answers are in the **ani** file. The **ani** files with the answers are not available to students.

Six types of response style are available:

- list (radio buttons) in which only one item may be selected from a list. The Answer Sheet displays radio buttons for this style. List items in the Question File should be labelled a), b) etc.
- list (check boxes) in which one or more items may be selected from a list. The Answer Sheet displays check boxes for this style. List items in the Question File should be labelled a), b) etc.
- rank ordering in which a list of items in the question must be ordered. The Answer Sheet shows a list of characters (a, b, c . .) and allows the user to drag them into the required order. List items in the Question File should be labelled a), b), c) etc. and the rubric should state clearly which direction the ranking is required.

pair matching	in which some or all items from two lists must be matched. The Answer Sheet displays two lists, one with upper case and the other with lower case letters. The user must drag a line from each item in the top list (upper case) to items in the bottom list (lower case), to indicate pairings. The top list is labelled A), B),C) etc. and must be at least as long as the lower list which is labelled a), b), c) etc. The lower case list items may each pair with more than one upper case list items, but not vica versa. In the answer box the user can pair appropriate items by clicking first on a character in the top list and then on a character in the bottom list.
text	in which a text string must be typed in. The answer sheet displays one text box into which the number must be typed. The correct answer can be defined as alternative text strings containing wild characters.
numeric	in which the answer is one or more single values. This type is the same as the random numeric type, as described below, except that no random valued parameters may be used in the question.
random numeric	the answer may be a single value or multiple values (as in specifying a vector). If multiple values are required, then each is typed into a separate text box, which can be arranged in a row, a column, or in a matrix. Rubrics of the form 'x=' 'y=' 'z=' can be put before each row. The question file can contain values that will be randomised when the question is set, and the number of significant digits with which each value is to be displayed can be specified individually. Both the number of significant digits and the number of places of decimals can also be specified both as values or as ranges. The correct answer (and alternative answers) can be defined as a single value, a range of values or as an expression involving the randomised values.

In self test mode

In the answer window an **Explain** button will be displayed if the question keywords have been edited into the control file. These keywords are passed to Microcosm with a *show link* action when the **Explain** button is pressed. These keywords can either be carefully chosen to lead to existing background materials, or they may be unique (e.g. w1.2q3.4) to lead to a specific document for a given question.

When a student has finished a question, clicking the **Mark** button marks the question and provides feedback via a Feedback Window. as illustrated in the examples shown below. It is possible to create specific feedback messages for all possible wrong answers or for classes of wrong answers.

An **Explain** button only appears in the feedback window if the Control File has the appropriate keyword entries. Helpful messages in the Feedback window should be provided in *simple text*. A message containing symbols and/or Mathtype equation(s) must be placed in a separate file accessed from the explain button through a unique generic link.

Question 1		
<p>Yes, this is correct. The quality of m strained, particularly when there is p interference with the judicial process</p>		
<input type="button" value="Stop Marking"/>	<input type="button" value="Explain"/>	<input type="button" value="Next Question"/>

5.5 Question Files

The question files are conveniently authored within a unit's Word storyboard document. In this case the destination directory or filename must appear at the start of the first line for the *splitfile* macro to function correctly, e.g. for W2.1 unit question sheet 1

Directory c:\mcm\waves\quiz
 Filename wm2_1f0.rtf

Subsequent tests for the same unit *following on* in the storyboard do not require the directory specification. They can start with:

Filename wm2_1f1.rtf
 Filename wm2_2f2.rtf

Alternatively the files can be prepared in Word separately from the script (using the storyboard template), and then saved as rtf files when ready.

Each question sheet should start with a three line title (as described in chapter 2) as shown, with the copyright footnote added after a single space at the end of the second line. The number in braces after "Self test" is the number of the question sheet. Single (1) and multiple question sheets (1), (2), (3), for a unit should each be given a number. The question number is in heading2 style (i.e. in dark red).

<p>Measurement and Uncertainty Unit 2.1 ² Self test (3)</p> <p>Question 1</p>
--

There is no constraint on students to answer the questions in any given order, or even to answer all the questions.

5.5.1 Multiple choice or rank ordering questions

Where the question is in any of the rank ordering or multiple choice styles, care should be used in laying out the lists. Lower case letters should be used for the lists (to match the answer sheet), and the selection should be laid out to maximise visibility. For example, if each answer is short, the list can be put into two columns.

- | | |
|---------------------|---------------------|
| a) 22.30 ± 0.20 | d) 20.30 ± 0.02 |
| b) 35.05 ± 0.00 | e) 20.30 ± 0.01 |
| c) 18.00 ± 0.01 | f) 20.30 ± 0.03 |

²© 1994 University of St Andrews

or possibly on one line

- a) 0.032 b) 0.02 c) 0.04 d) 0.025

If each answer is a sentence, then use a hanging indent for each option:

- a) The flute is closed at the blown end and and open at the far end, so that the fundamental mode has one quarter wavelength in the instrument's length.
- b) The flute is open at the blown end and and open at the far end, so that the fundamental mode has one quarter wavelength in the instrument's length.
- c) The flute is closed at the blown end and and open at the far end, so that the fundamental mode has one half wavelength in the instrument's length.
- d) The flute is open at the blown end and and open at the far end, so that the fundamental mode has one half wavelength in the instrument's length.

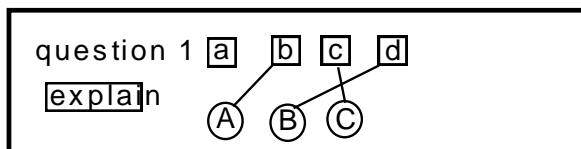
5.5.2 Pair matching questions

When using pair matching, the two lists are best arranged in two columns with the items in the list on the right being matched to the items in the list on the left. The left hand list will use lower case a), b), c), etc, and the right hand list will use upper case A), B), C), etc.

For example:

- | | |
|------------|---------------|
| a) flute | A) brass |
| b) trumpet | B) string |
| c) timpani | C) percussion |
| d) violin | |

In this case, a correct response could look like this:



An item from the lower case list may match more than one item in the upper case list, but not the other way round. Either list may be shorter or longer than the other, however.

Do not use the word 'Answer' before the lists and do not indent the list if more than one column is used or if lengthy text is used.

5.5.3 Textual and numeric questions

Question files styles require no specific considerations, except that it should be remembered that it is difficult to prepare questions that elicit a single unique text answer.

5.5.4 Random numeric questions

A random numeric question must be identified in the Word document by inserting the bitmap c:\docuverse\rtfvalq.bmp (a roundal) on the first line after the question number. N.B. this bitmap must be *linked to file*. The question files will be processed during final integration so that this bitmap will not be visible to the user taking the test. For each quantity to be randomised insert the bitmap c:\docuverse\rtfvalue.bmp at the position the value is to appear. This bitmap must also be *linked to file*. The rtf file displayed to the student will have specific values in place of the variables.

There are several parameters that can be specified for this style of question: the number of significant figures and the number of decimal places of the displayed values, and the number

of significant figures to be used in the answer. The answer can be specified as a single value, a column or a row vector, or a two dimensional vector. For each answer an expression involving the variables is required. Different precisions can be specified in alternative (or wrong) answers. A range can be defined as +- a percentage of the correct answer, and “any” value can be specified when answers are vectors.

5.6 The Associated Data file

Formative tests (self tests)

The actual test control (qni) file produced by the test editor has the following entries:

[Header]

- title=** This will appear in the title bar of the answer box.
- environment=stomp** This is to allow the test tool to have different functionality when used with other systems.
- mode=** The option is **linked** or **unlinked**, for use within or outside microcosm (for normal STOMP use the state is linked)
- test=** The test type is defined as **formative** (self test) or **summative** (assessed test). This can be overridden by a command line declaration in the launcher (mcs) file. which includes 3F for three formative tests for a unit or 1S for one summative tests (see section 5.10 below).
- question=** The number of questions n in the test, is specified as an integer.
- rubric=** This is one of three rubrics displayed in the answer control box., should be a general instruction such as **Press the 'mark' button to check your answer** for self tests, or **Answer all the questions you can before pressing the 'submit' button** for assessed tests.
- StyleRubric=text** This rubric has a default form for each style, containing a suitable instruction for the style of question. The defaults can be overridden.
- AlwaysOnTop=** The options are false(default) or true. It can be very inconvenient at some screen resolutions for the answer box to always obscure the question window.

For each question:

- [Question q]** q is an integer. The numbering must be sequential
- title=** This will usually be the question number which provides the title for the answer box. (an alternative entry is possible here, but it should not be wider than about 3 characters).
- style= n** The integer n corresponds to the following styles of question and thus answer box layout:
- | | |
|------------------------------|-------------------------|
| 1 list (radio button) | 5 numeric |
| 2 list (check boxes) | 6 text |
| 3 rank ordering | 9 random numeric |
| 4 pair matching | |
- Choice= $l m$** The first integer l defines the number of answer choices available for question styles **list** (1 & 2), **rank ordering** and **pair matching**.

The (space separated) second integer *m* is *only* used in the style **pair matching**. The number *l* is the length of the first list (i.e. the number of boxes with lower case letters) and the number *m* will be the length of the second list (upper case letters).

Keywords= This is a space separated list of words which need to be the sources of generic links to useful background material for the question.. Generic links to material within the contribution should be made by the author. For generic links to material outside the contribution let the editor at Surrey know what is required.

Rubric=text This is one of two rubrics that are specific to each question. This message should be specific to the subject or manner of the question, rather than the style being used.

Right answer

Right=expression The *expression* specifies a correct response. The *expression* depends upon the style of a question as follows:

Style 1 for **radio button list**

expression = single letter **a..j**

e.g. right=b radio button b checked

'Other conditions' may use the **xor** function defined in style 2.

Style 2 for **list(check boxes)**:

expression = single letter **a..j**

expression = *function*(string of letters)

where *function* = **and, or, xor**

e.g. right=b box b only must be checked

right=and(df) both d and f boxes must be checked

right=or(abj) any combination of boxes a, b and j may be checked

right=xor(jdf) one only of boxes d, f and j may be checked.

Style 3 for **rank ordering** of choice *n*:

expression = *n* digits 1 to *n*

expression = *function*(strings of digits)

where *function* = **or**

e.g. right=32415 the digits must be in the order specified

right=or(12345 54321) either of these two rank orderings will be accepted

The meaning of the first example above, is that item 3 in the list (i.e. c) should come first, so that 32415 = cbdae

Style 4 for **pair matching** of choice *m n*:

expression = *n* upper case letters A B C etc. or the character _ signifying a blank box.

In **IfOther** expressions the character * is also allowed, and stands for any letter from the second list. This allows specific pairing cases to be identified without repetition. The number of characters in each list must equal the number of boxes requested - i.e. the first *choice* parameter.

expression = *function*(strings of upper case letters)

where *function* = **or**

e.g. right=DCBA the letters must be in the order specified
 right=or(BCDA DBAC ACBD)
 any of these three orderings will be accepted
 IfOther=or(**C* *C**) would correspond to C being paired with c or b, with
 any other combination of letters or blanks in the other positions.

Style 5 for **numeric** answers

This style the testing tool handles *numeric* answers by evaluating and comparing with single values or ranges of values. Numbers may be entered in conventional 'computer' formats, e.g. as integers (31), decimal fractions (3.142) or in exponent format (4.9e3).

expression = single value

expression = [minimum value..maximum value]

Style 6 for **text** answers

This style handles *text* answers (which may be alpha or numeric) by matching each character in the string.

expression = 'string'

expression = or('string1' 'string2')

The strings must be delimited by single quotes.

The strings may contain the character '*' which is treated as a 'wild card' that can stand for any single character.

Style 9 for **random numeric**

This style accepts numeric answers by evaluating them and comparing the value with expressions which can contain random variables. The variables can be defined by range or by an explicit set of values.

expression = a pascal style expression containing the variables within square brackets.
 e.g. right = [V1] * sin([V2] + 1.57078) The V may be upper or lower case.

The expression may be followed by "%n" where n may be integer or fractional. A value will then be correct if it falls within the inclusive range of (*expression* - n%) to (*expression* + n%).

"any - when the answer is a vector, the *expression* may be replaced by the word ANY (upper or lower case), so that any value will be accepted. This is designed to allow partially correct answers to be dealt with.

AnswerSig=*minsig* [*maxsig*]

The number of significant figures used in a student's response can be specified precisely, or as an inclusive range.

Vn= *rangemin rangemax* [*sigfig* [*decimals*]]

or

Vn= (*value1* [, *value2* [, *value3* [. .]]]) [*sigfig* [*decimals*]]

Variables are defined as ranges or as a list of values. The list of values must appear within braces and be comma separated. The number of variables starts from 1 for each question, and the numbering must be consecutive. The number of significant figures to be displayed and the number of decimal digits may also be specified.

If the **Explain** box is checked in the test editor and keywords specified, the answer sheet will display an *explain* button:

Rightkey=keywords Single space separated *keywords*.are entered here If the user presses the explain button, these keywords are passed to MCM show-links..

IfRight=message The *message* entered here (simple text only accepted) is put on the screen if the answer is correct.

Wrong answer:

If the **Explain** box is checked in the test editor and keywords specified the answer message box will display an *explain* button .

Wrongkey=keywords Single space separated *keywords* are entered here. If the user presses the explain button these keywords are passed to MCM show-links.

IfWrong=message The *message* entered here (simple text only accepted) is put on the screen if the answer .is wrong and does not match any of the specific wrong answers specified below.

Specific wrong answers

This allows specific wrong answers to be identified and given their own helpful message and explain button keywords.

Use the *ins* (insert) button to create a new special case and the *del* (delete) button to remove a case.. The cases are tested in the order shown, so the conditions can become successively more specific. The order can be changed by means of the *up* and *down* buttons.

Conditions, messages, and keywords can be edited by selecting and using the *edit* button and appear as the following entries in the qni file:

OtherCondition1=Specific wrong answer entered using syntax outlined above in the 'Right answer' section.for each style.

OtherMessage1=message as above

OtherKey1=keywords as above

For the random numeric style the significant figures are specified using

OtherSig1= with the specification as above

OtherCondition2=

etc.

Summative tests (assessed tests)

The test and question information entered using the test editor is split between the qni and ani file. The test and answer window control data goes into a qni file and the actual answer information into the ani file. The ani files will be removed at Surrey and distributed separately from STOMP. Marks can be awarded for correct answers. There is an option for alternative answers to attract marks if entered under the 'Wrong answer' or 'Specific wrong answers' conditions.

5.7 Help (physics).

There are five different ways in which physics theory help can be obtained when using the testing tool in **linked** mode. In unlinked mode only the first of these is available.

1. Question files are viewed using the rtf viewer and conventional Microcosm linking is available as normal.

2. The **Explain** button against each question in the **answer sheet** does a 'show links' on the **keywords** provided for that question.
3. The **Right answer** feedback window displays:
 - A simple text **message** entered using the test editor (qni file **IfRight=message**).
 - An **explain** button which passes on the **Right answer keywords** (qni file **Rightkey = keywords**) to Microcosm for 'show links' . The keywords must be sources of generic links to helpful material.
4. The **Wrong answer message** (qni file **IfWrong=message**) and **Explain** button with **Wrong answer keywords** (qni file **Wrongkey = keywords**) is treated the same way.as the **Right answer** in 3.
5. The **Specific wrong answer messages** (e.g. qni file **OtherMessage1=message**) and the **Explain** button with the **Specific wrong answer keywords** (qni file **OtherKey1,2 etc**) are also treated the same as the **Right answer** in 3..

Ideally, the question **Explain** button keywords should provide general background material for the question. The **Right answer (Rightkey=)** keywords should provide links to material that would be of further interest. The **Wrong answer (Wrongkey=)** keywords should provide links to other helpful information that is more specific than the question **Explain** button material. The **Specific wrong answer** (e.g. **Otherkey1, 2 etc**) keywords should provide links to specific material for the identified error. Clearly these four should link to different material. It is appreciated that this can be very difficult to achieve and in many cases sufficient material will just not exist to be able to make sufficient discrimination.

5.8 Handling responses

Formative mode

The two buttons available to the user are **Mark** for individual questions and **Mark all** for the test. When the **Mark all** button is clicked answers are checked to see a) if any have not even been attempted, and b) whether those that have been attempted are logically wrong (i.e. not finished or containing duplicate responses). In all cases the student is asked to choose between continuing anyway and going back and correcting/finishing the questions.

If in **formative** mode each question's response will be judged right or wrong. At least two messages are available for feedback, and other messages can be displayed for special answer conditions. As stated above, when a message is displayed the user can be invited to ask for further information using the Explain button (if provided), which passes the keywords to Microcosm for 'show Links'.

Summative mode

In **summative** mode only the **Submit** button is available. when this is clicked the student's responses are saved to a data-base and the testing tool terminates. The student is warned if any questions have not been attempted, and they are given the chance of going back.

5.9 Working in the Answer Sheet.

In the answer sheet the different question styles have the following facilities for entering the answers:

list (style 1 and 2) - radio buttons (for a single answer) or check boxes (for one or more answers)(

rank (style 3) - lower case letters must be dragged (mouse) into numbered boxes.

pair matching (style 4) - Pairs are selected from a set of lower case letters a, b, etc (in boxes) and upper case letters A, B, etc (in circles). Clicking on a box and then on a circle makes the pairing, which is shown by a line.

numeric and text (style 5 and 6) - a single text entry box is provided. Text entry is standard, and numeric entry is as described below for random numeric.

random numeric a single entry box, or an array of entry boxes can be provided. Numeric input can be entered as integers, or the two standard floating formats (123, 123.456, 12.34e4). If the format is not valid, then an error message is displayed and the user can enter the number again. The text box, or the left hand column (if an array of boxes is specified) can be preceded by 'a=', or 'x=', 'y=', etc. Numeric and text answers can only be typed in via the keyboard.

Assessed tests may only be marked once, and there is no immediate feedback to the students about whether questions were right or wrong.

5.10 Using the testing tool within Microcosm.

As far as Microcosm is concerned, the testing tool is just another programmed activity. This means that for each test an **.mcs** file is required with a command line that specifies the tool and the root name of a set of question documents with the number of documents and style of the test - **F** for formative or **S** for summative. Each question document requires two files, an **rtf** that is displayed as the question File and a **.qni** that is used by the testing tool as the control file. For assessed tests, the **.ani** file is only used by the marking tool. The same base filename must be used for these two or three files. The **.rtf** and **.qni** files for each test must be in the same directory as the **.mcs** file . The command line in the **.mcs** must specify the base filename without extension. e.g., in a situation where there is only one formative (self test) question file the command line entry in mm1_3ef.mcs would be:

```
[Microcosm Script]
```

```
Command=%system/stomp/stbin/%sttest.exe mm1_3ef $Filename$ 1F
```

The single question file and control file in this case is mm1_3ef0.rtf and mm1_3ef0.qni.

Where there is a set of assessed question files the command line might be

```
[Microcosm Script]
```

```
Command=%system/stomp/stbin/%sttest.exe wm1_5cf $Filename$ 5S
```

This example would mean that there were five sets of files wm1_5cf0.rtf, wm1_5cf0.qni, wm1_5cf0.ani, wm1_5cf1.rtf, wm1_5cf1.qni, wm1_5cf1.ani . . . wm1_5cf4.rtf, wm1_5cf4.qni, wm1_5cf4.ani, and that the test is summative so that the user would be presented with a randomly selected test from the five possibilities. In a formative test the user would be asked to select a file.

Note that the last specifier on the command line (e.g. 1F, 5S) specifies the style of the test - **Summative** or **Formative**, and this overrides any specification in the qni file (see section 5.6 above).

The **.mcs** file(s) must be imported into Microcosm in the normal way, as a Launcher document type.

The **.rtf** document (the question file) must be imported in to Microcosm as an ordinary rtf document. It is no longer necessary to link the Test Tool with the **.rtf** document by a Generic Link. The file is automatically displayed by the Test Tool

All question (**.rtf**) files and associated control (**.qni** and **.ani**) files must be put into your quiz directories..

If n is the number of question sheets in the test.

For each **formative** test there are $2*n+1$ files (*base.mcs, base0.rtf, base0.qni, base1.rtf, base1.qni, base2.rtf, base2.qni, etc.*)

For each **summative** test there are $3*n+1$ files (*base.mcs, base0.rtf, base0.qni, base0.ani, base1.rtf, base1.qni, base1.ani, base2.rtf, base2.qni, base2.ani, etc.*)

The **.ani** files will be removed during module integration and distributed separately.

The Launcher replaces \$FileName\$ with the full path file name of the **.mcs** file. The Test Tool uses this information to identify the full path names of the current directory and, hence, the control (**.qni** & **.ani**) and question (**.rtf**) files.

5.11 Filenames

All problem and test files go into the quiz directory for the contribution. They are named as shown in the following examples:

Problem file for unit 2.1 in Waves is **pw2_1.rtf**, Measure **pm2_1.rtf** and Optics **pop2_1.rtf**.

Formative (self test) question sheets for Waves unit 2.1 are:

sheet 1 is **wm2_1f0.rtf**, sheet 2 is **wm2_1f1.rtf**, sheet 3 is **wm2_1f2.rtf**
etc

The associated launcher file is **wm2_1f.mcs**.

The command in this mcs file will be

```
command=%/system/stomp/stbin%sttest.exe wm2_1f $FileName$ 3F
```

The data files for these three test sheets are:

wm2_1f0.qni, **wm2_1f1.qni**, **wm2_1f2.qni**

Summative (assessed test) question sheets for this unit are:

sheet 1 is **wm2_1s0.rtf**, sheet 2 is **wm2_1s1.rtf**, sheet 3 is **wm2_1s2.rtf**
etc

The associated launcher file is **wm2_1s.mcs**.

The command in this mcs file will be

```
command=%/system/stomp/stbin%sttest.exe wm2_1s $FileName$ 3S
```

The data files for these three test sheets are:

wm2_1s0.qni, **wm2_1s1.qni**, **wm2_1s2.qni**

wm2_1s0.ani, **wm2_1s1.ani**, **wm2_1s2.ani**

For tests in the Measure and Optics modules the **w** should be replaced with **m** or **op**.

The question files for assessed tests are encrypted for security, the decryption can only take place when the testing tool is used to look at the question files. For additional security the ani files (which contain the answers) are also encrypted, even though these files should not be placed on any machine available to students.