Good Laboratory Practice

GUIDELINES FOR THE DEVELOPMENT AND VALIDATION OF SPREADSHEETS

Release Date: 31.01.2018

Version: 2.0

TABLE OF CONTENTS

1 FOREWORD

The present document provides guidance, i.e. a basic strategy for GLP-compliant development and validation of spreadsheets. The guidance should aid test facilities and promote the use of a common standard, but should not be considered as legally binding. A test facility management may use different approaches, as long as they are in compliance with the OECD Principles of Good Laboratory Practice [1, 2].

The AGIT (**A**rbeits **G**ruppe **I**nformations**T**echnologie) is a working group consisting of representatives from Swiss GLP monitoring authorities and Swiss industry with the aim of proposing procedures, which are practical for use in test facilities fulfilling GLP regulatory requirements.

The Guideline for the development and validation of spreadsheets was issued in August 2011. This updated version (version 2.0) is in line with the OECD Advisory Document No. 17 (replacing OECD Consensus Document No. 10 [3].

2 INTRODUCTION

Spreadsheets are widely used for the storage, processing and reporting of data. As these spreadsheets are associated with the conduct of non-clinical safety studies intended for regulatory submission, it is essential that they should be developed, validated, operated, maintained, retired and archived in accordance with the OECD Principles of Good Laboratory Practice (GLP) [1, 2].

3 SCOPE

The aim of this document is to provide guidance on GLP-compliant development and validation of spreadsheets. It focuses on spreadsheets with data from a GLP study. This data is processed by calculations, transformations, Visual Basic for Applications (VBA) including recorded macros, and add-ins. The spreadsheets may also contain electronic raw data.

Out of scope:

- tabulating spreadsheets such as lists that do not contain raw data from GLP studies
- spreadsheets employed as single-use calculation sheets: in this case, a new file is created, data is entered and calculations are set up, e.g. mean and SD; the file is subsequently printed, dated and initialled. These spreadsheets need a Quality Control (QC) check on completion (the calculation should be repeated with a calculator). If this kind of spreadsheet is used on a routine basis, full validation is recommended.

4 REGULATORY REQUIREMENTS

The validation of computerised systems is required by the OECD Principles of Good Laboratory Practice [2]. A more detailed description of the application of the Principles of GLP to computerised systems was already published in the OECD Advisory Document No. 17 [3]. This document defines requirements for the operation of computerised systems in a GLP regulated environment.

The OECD GLP Principles and OECD Advisory Document No. 17 define validation as *"Action of proving that a process leads to the expected results. Validation of a*

computerised system requires ensuring and demonstrating the fitness for its purpose." The validation process provides a high degree of assurance that a computerised system meets its pre-determined specifications.

5 DEFINITIONS

A spreadsheet application can be defined as a computer program that represents information in a two-dimensional grid of data, along with formulas and logical functions that relate to the data.

In a GLP environment, the following types of spreadsheets are widely used:

- tabulating spreadsheets
- processing spreadsheets containing a copied set of raw data
- processing spreadsheets containing raw data.

Tabulating spreadsheets

These spreadsheets contain lists of information. Typical examples are a laboratory equipment inventory or a master schedule. Spreadsheets of this type do not per se contain raw data from a GLP study and therefore do not fall within the scope of these guidelines.

Processing spreadsheets containing a copied set of raw data

These spreadsheets are the most common. Data collected/obtained during a GLP study is processed within the spreadsheet (by calculations, transformations, VBA including recorded macros, add-ins) to obtain additional results.

Raw data is produced by a computerised system (equipment, software), and the data is entered manually, automatically or via copy-and-paste into the processing spreadsheet. In this case, the raw data resides in the computerised system or is paper-based. For example, haematology parameters may be measured on a system and the data exported as a spreadsheet file. The data is copied into the processing spreadsheet. In this case, the user should always check whether the raw data has been transferred completely and correctly.

Processing spreadsheets containing raw data

It is also possible to enter measured values directly into the spreadsheet for further processing. In this case, the spreadsheet contains raw data. A typical example is the import of data from an electronic balance via an interface or a communication port. Animal body weights are entered directly into the spreadsheet, and the group mean calculation and inter-group statistical analysis are performed on this sheet. The generation and subsequent import of data into the spreadsheet should be covered by the validation of the computerised system; therefore, these activities are out of scope in this document.

Processing spreadsheets containing raw data should fulfil all the requirements specified for the handling of electronic raw data [4]. Additional spreadsheet add-ins or the use of a Document Management System (DMS) may be necessary to fulfil these requirements. For example, a GLP-compliant audit trail might only be feasible with the aid of a DMS.

6 VALIDATION APPROACH

The general approach for the validation of spreadsheets is comparable to other validation approaches, such as that described by the AGIT guidelines for the validation of computerised systems (CSV) [5]. It includes a common validation process (user requirements, risk assessments, specifications, planning, testing, reporting), the assignment of responsibilities, proper validation documentation, and rules for the archiving and change management processes.

However, the development and validation of spreadsheets often allows a simplified though still formal approach regarding many of the aspects of a validation project. During the validation process, for example, the user requirements and functional specifications (see Chapter 7 or the operational and performance qualification (see Chapter 8) may be summarised in a single step and documented in a single document.

With regard to assigning responsibilities, the spreadsheet developer is in many cases identical with the scientist who is also responsible for a given test method (see also Chapter 7). In this case, it has to be ensured that the developer is always aware of his current role, and that the phases of development and validation are clearly separated. If changes are necessary during the validation phase, they should be documented, evaluated and re-tested.

Change management procedures may focus on version control, and necessary changes may be implemented with less effort than for CSV (see Chapter 9).

Finally, with regard to the archiving of spreadsheet-processed data or spreadsheets containing raw data, reference is made to the recommendations given by the AGIT in an earlier document [6]. It has to be decided whether raw data is to be stored in an electronic form or as hard copy.

Overall, the extent of the validation depends on the risk analysis and the complexity of the spreadsheet, and in many cases the validation procedure may be quite short. However, it is highly recommended to adopt a systematic approach, deciding what is necessary on the basis of the risk analysis before omitting any testing procedures.

In the following chapters, individual aspects of the validation approach for spreadsheets will be described and covered only where they differ from earlier AGIT papers.

7 DEVELOPMENT PROCESS

7.1 User Requirements

Before the spreadsheet is developed, the users should define what they expect the spreadsheet to do, which calculations have to be performed, the layout, etc. This is defined in the User Requirement Specifications (URS), which would also include the Functional Specifications. For example, a (non-exhaustive) URS for the gravimetric pipette control spreadsheet may look like this:

URS-1. **The spreadsheet should be applicable for volumes from 0.01 to 10 mL**
URS-2. **The spreadsheet should only accept a sample number of 10**
URS-3. **The spreadsheet should calculate the mean, standard deviation (SD) and relative SD (RSD=(SD/average)x100%) of the 10 samples**
URS-4. **The spreadsheet should use the density of water between 20°C and 25°C in steps of 1°C**

URS-5. **The Inventory Number of the pipette should be visible on the screen and on the print-out.**

URS-6. **The spreadsheet should indicate whether the accuracy and RSD fulfil the acceptance criteria according to the SOP**

URS-7. **The spreadsheet input should only allow numerical values for the nominal volume of the pipette, sample weight and temperature**

...

Requirements concerning security, protection and GLP compliance have also to be taken into account:

URS-21. **The calculations in the spreadsheet should be protected against unauthorised modifications**

URS-22. **After completion of the measurement, the spreadsheet should be initialled and dated electronically by the user**

URS-23. **Making changes to the spreadsheet (modification/deletion) after signing off (initials/date) should not be possible**

URS-24. **The spreadsheet template and completed spreadsheets should be retained in a secure location in an electronic form with access control**

...

The requirements specified by the GLP Principles for the handling of electronic raw data are illustrated by cell and worksheet protection, but not thoroughly elaborated in the example file in Annex A. Usually, other software tools – such as document management systems – are used for that purpose.

7.2 Spreadsheet Design

The spreadsheet design corresponds to the system architecture of a software program, and a good design will:

- reduce the number of errors during programming,
- make it easier to maintain, adapt or further develop the spreadsheet and
- make it easier to verify the function during validation.

To illustrate a structured layout, the sample spreadsheet shown in Figure 1 is divided into an area where data input occurs (columns A and B: inventory number, nominal value, sample weight, water temperature), an area where the results are calculated and presented (columns D–F: mean, SD, RSD, etc.) and an area containing constants and criteria (columns H–L: water density at different temperatures, maximum Delta and RSD). The different areas may be highlighted by different colours or shadings.

Figure 1: Spreadsheet layout

The spreadsheet is password-protected, and data can only be entered in the marked cells (column B). Data entry is secured by error messages and ranges (see Annex A).

It is important for the design of a spreadsheet that:

- input and output areas are separated;
- calculations, constants and parameters cannot be changed by the user (locked cells);
- a blank template is used – the use of "master copies" with typical data should be avoided; and
- the template is password-protected (at sheet, workbook and VBA code level).

At this stage, it should be defined where the productive template is stored. The file paths for automatic opening or saving of files and the security settings should be adapted accordingly.

Such a design provides the basis for straightforward validation. All input data, constants and calculation results are shown on the spreadsheet, which allows stepwise verification, e.g. with a calculator. All the information required to verify the calculation is a printed hard copy or a corresponding screenshot.

7.3 Development

Based on the design, the spreadsheet can now be programmed. This phase should be carried out systematically. Many experienced spreadsheet users develop a spreadsheet and add further functions as needed. In order to fulfil the URS, however, a stepwise approach is recommended.

If VBA code is used, parts of the code can be written automatically by recording the manual operations. These parts of the code have to be checked in order to ensure the correct cell references.

7.4 Development Testing

Systematic testing of User Requirement Specifications
At this stage, the developer tests whether the spreadsheet fulfils the URS. A simple set of typical data for the input values is used and varied in order to simulate the situations that are needed to test all URS.
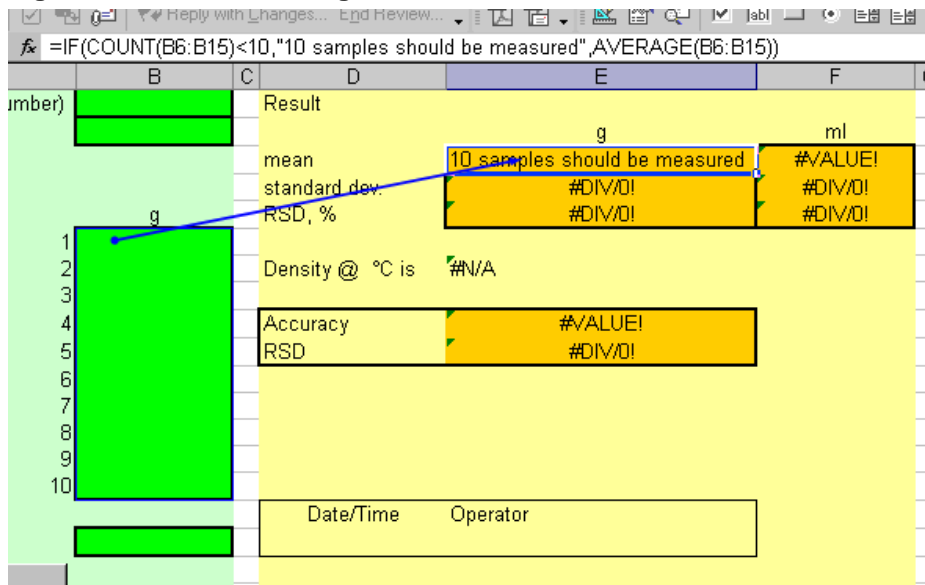
Code review

It is highly recommended that this review should be conducted by two persons – the developer and an experienced reviewer who is familiar with the intended use and the applied functions.

For spreadsheets, the code review consists of checking the content of cells, especially formulas, formula locations, cell numbers and ranges, syntax of additional functionality, and VBA code.

The cell contents can be reviewed by using the "view by formulas" functionality (see Figures 4 and 5 in Annex A). In particular, cell contents created by the automatic "fill-down" function have to be verified, since the use of this function can be error-prone, depending on the zoom factor and mouse sensitivity. The formula auditing tool can be useful for checking cell ranges and references (see Figure 2).

Figure. 2: Formula auditing tool

The VBA code (see Figures 9–11, Annex A) has to be evaluated line by line for consistency, program logic, correctness, etc.

The code review should be documented, for example, by the developer and the reviewer signing off the checked print-outs of the formula views, auditing tool views and VBA code.

7.5 Release for Qualification

The documentation of development should now be complete. The sheet protection features are installed, and a formal, documented handover for the qualification process takes place. From this stage onwards, version control of the spreadsheet is highly recommended.

8 QUALIFICATION PROCESS

The purpose of the qualification process is to demonstrate that the developed spreadsheet is valid for its intended purpose. Within this process, it has to be demonstrated that the desired result – namely, a set of processed data, which was subject to operations such as calculations, rounding, formatting and re-grouping for

presentation as a table, chart or report – is achieved in a correct and reproducible manner.

8.1 Installation Qualification (IQ)

A spreadsheet template is a critical file since it usually contains the complete source code (VBA code and spreadsheet functions) and can be used to execute the program. In terms of software development, the source code and the compiled version are stored in the same file. Therefore, it is important to make sure that the spreadsheet template is stored in a secure, controlled environment. Servers with the appropriate read/write/modify security settings can provide a reasonable level of control and are therefore the recommended location for storage of templates. Alternatively, the use of a DMS comprising tools such as version control and backup function, as well as the possibility of access control can be considered.

It is self-evident that adequate backup measures have to be in place so as to avoid any damage to or loss of the template (e.g. if any of the above measures should fail).

In order to ensure that the spreadsheet can be utilised throughout the test facility, it may be necessary to check that

- the spreadsheet can be run on different computers (identical or different hardware configurations, user profiles, local settings) and
- the spreadsheet can be installed and run using different software versions.

Ability to run the spreadsheet on different computers
This is especially important when the spreadsheet is stored as a template on a server and run by different users on their desktop PCs. In many companies, the computers may have identical or similar hardware, but the configuration (properties or settings, e.g. operating system language) may be different depending on the use of the specific PC. For example, a computer used for instrument control may run with US versions of the software, whereas further data handling may be done in the office using a local version.

Special attention is required if normal use involves automatic access to raw data files (e.g. reading an export file created by an analytical instrument) and/or automatic or manual saving to certain folders on the network.

It is important to test whether a spreadsheet is capable of running on different computer hardware if that is part of its intended use. This would require a test of the influence of user profiles, local settings, etc.

Ability to install and run the spreadsheet using different software versions
It is important to test the spreadsheet on different software configurations if they are actually used in parallel. This does not need to be tested during installation qualification if it can be ensured that identical versions of the operating system and spreadsheet software will be used.

8.2 Operational Qualification (OQ)

Systematic tests of the initial user requirements have been performed during the development testing phase (usually by the software developer). It is not necessary to repeat every single test in this phase, but typical cases with all the basic functions should be included. It is recommended that all tests should be performed with different and independent data sets.

Test for ruggedness

Choose input values that lead to unacceptable or impossible results. Try to enter data in incorrect formats or in the wrong cells (text instead of numbers, values out of range, wrong number of decimal places, or real numbers instead of integers; see Table 1). In these cases, the user requirements should already specify what behaviour is required and expected from the system. As many of these tests have already been included in development testing, the focus should be on issues arising from the productive environment at that stage. This may include repeated tests with real data in order to discover malfunctions due to loops, or values that are not correctly re-set at the end of the execution. Testing of security aspects should also be included at this stage since the spreadsheet is now installed in the productive environment.

Table 1: Examples of tests to evaluate the ruggedness and security aspects of the spreadsheet

Test relating to URS:	Test	Test result/acceptance criteria
...		
URS-7	Enter "q" as weight in the sample 1 field.	The error message "Please enter a weight in grams!" appears.
URS-7	Enter "q" as temperature in the temperature field.	The error message "Please enter a temperature between 20 and 25°C!" appears.
...		
URS-21	Place the cursor in cell E4 and press "Delete" to erase the content of the cell.	The error message "The cell or chart you are trying to change is protected and therefore read-only" appears.
...		
URS-22	Enter "A1" into the Pipette (Inventory Number) field and press the "Enter/Save" button.	The error message "You have to fill-out all the green input cells!" appears.
...		

8.3   Performance Qualification (PQ)

Table 2: Examples of tests with selected input data

Test relating to URS:	Test	Test result/acceptance criteria
...		
URS-3, URS-6	Enter "A1" in the Pipette (Inventory Number) field, "10" for the Nominal Value (ml) and "22" for the Temperature (°C). Enter the following values for samples 1–10: 9.98, 9.98, 9.99, 9.99, 10, 10, 10.01, 10.01, 10.02, 10.02	Results: mean 10 g, SD 0.015, RSD % 0.15, Accuracy ok, RSD ok
URS-3, URS-6	Now enter the following values for samples 1–10: 9.95, 9.95, 9.99, 9.99, 10, 10, 10.05, 10.05, 10.4, 10.4	Results: mean 10.078 g, SD 0.173, RSD % 1.717, Accuracy ok, RSD out of spec
URS-3, URS-6	Now enter the following values for samples 1–10: 10.3, 10.3, 10.3, 10.3, 10.3, 10.4, 10.4, 10.4, 10.4, 10.4	Results: mean 10.35 g, SD 0.053, RSD % 0.512, Accuracy out of spec, RSD ok
URS-3, URS-6	Now enter the following values for samples 1–10: 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0	Results: mean 10.00 g, SD 0.000, RSD % 0.000, Accuracy ok, RSD ok
...		

Test with selected input data
The goal of this step is to verify all possible and all special cases (including boundary values), even if they are very rare, pursuing a systematic approach (see Table 2). In contrast, the real data will typically be of relatively high consistency and cover only realistic cases, not necessarily the extremes. Therefore, the input data has to be selected to cover all functional requirements and combinations of theoretical cases for decision criteria (branching, conditional instructions), as well as all extreme values or special data points.

Test with real data
From the user's point of view, this is the principal part of the validation process – applying real measurement data as input data. Data input can be done manually or by importing a file, depending on the intended use of the spreadsheet. Irrespective of whether this is a manual or an automated step, real data from previous measurements is to be used. It is important to use a sufficient number of different real data sets in order to obtain a high level of confidence that the spreadsheet functions as intended.

8.4   Release

After the final Spreadsheet Validation Report has been approved by the validation director (following successful completion of qualification), the system is released for use by test facility management. This responsibility may also be delegated to the system owner [5]. Before the spreadsheet is released for GLP use, additional points should be considered, e.g. the availability of a Standard Operating Procedure (SOP) and the documentation of adequate user training.

Any modification of the spreadsheet after release should comply with change management procedures (see chapter 9). All released spreadsheet versions should be archived.

9 CHANGE MANAGEMENT

Any changes to the spreadsheet should be handled using systematic change management procedures. If a proposed change is to be implemented, the system owner should perform a risk assessment and decide on the amount of additional testing required. After successful testing, the spreadsheet should be formally released as described in chapter 8.4.

Changes in constants
Certain changes to a spreadsheet may not necessitate additional testing. For example, in the case considered above, the %Max Delta and %RSD may be changed in the SOP. The programmer decided during the design phase to have them as separate constants, and the calculations refer to the values given in these fields (J2 and J3; see Figure 1). Changing these constants does not pose any risk for the other calculations on the spreadsheet. The version number of the sheet should be changed and documented, and the new version released.

Changes in formulas, VBA code, etc.
Other types of changes may require additional testing. In our example, the temperature range may have to be extended to 20–30°C to make gravimetric control of pipettes also possible in the summer. The cell range of the temperature values I10–I15 is used for the lookup function, and an error in the extension of the range and the new values

is not likely to be detected during operation. Therefore, appropriate tests should be performed.

The correction of any error detected in the template during operation should be handled via change management. In addition, the influence of the detected error on historical data should be evaluated.

Whether a change was initiated by a change request or in order to correct an error does not alter the change management procedure. Depending on the risk assessment, the extent of testing will be adapted to the identified risk of errors and the probability of detection.

Unless otherwise specified, the system owner is responsible for the processes of change handling, testing, version control and distribution of a new version, together with SOP updates and training.

10 REFERENCES

[1] Ordinance on Good Laboratory Practice (GLP) of 18 May 2005 [RS 813.112.1] as last amended on 1 December 2012. (OGLP)

[2] OECD Series on Principles of Good Laboratory Practice and Compliance Monitoring No. 1: OECD Principles on Good Laboratory Practice (as revised in 1997). Environment Directorate, OECD, Paris, 1998. (OECD)

[3] OECD Series on Principles of Good Laboratory Practice and Compliance Monitoring No. 17: Advisory Document of the Working Group on Good Laboratory Practice. Application of GLP Principles to Computerised Systems. Environment Directorate, OECD, Paris, 2016. (OECD)

[4] Working Group on Information Technology (AGIT): Good Laboratory Practice (GLP); Guidelines for the Acquisition and Processing of Electronic Raw Data. (AGIT)

[5] Working Group on Information Technology (AGIT): Guidelines for the Validation of Computerised Systems (AGIT)

[6] Working Group on Information Technology (AGIT); Good Laboratory Practice (GLP); Guidelines for the Archiving of Electronic Raw Data in a GLP Environment. (AGIT)

11 WORKING GROUP ON INFORMATION TECHNOLOGY

The Working Group on Information Technology (AGIT) was founded on 27 March 1998 with the objective of discussing relevant topics of Good Laboratory Practice (GLP) in the field of information technology between industry and the monitoring authorities. The AGIT intends to set up guidelines based on legislative requirements and practical experience to support test facilities introducing information technology tools to computerised systems in practice. OECD GLP Advisory Document number 17 on the *Application of the Principles of GLP to Computerised Systems* is used as a basis for discussion.

The members of the AGIT are representatives of the Swiss GLP monitoring authorities (Olivier Depallens, Swiss Federal Office of Public Health; Elisabeth Klenke and Daniel Roth, Swissmedic, Swiss Agency for Therapeutic Products; Christoph Moor, Federal Office for the Environment), and invited experts from industry (Peter Esch, Novartis Pharma AG; Stephan Hassler, Innovative Environmental Sciences Ltd.; Silvio Albertini, F. Hoffmann-La Roche AG; Christine Wurz, Idorsia Pharmaceuticals Ltd.).

For the convenience of users, AGIT publications are available on the Swiss GLP website (see Good Laboratory Practice (GLP)). The Swiss GLP homepage also provides links and references to guidelines, laws and regulations, definitions etc.

AGIT Publications:

- Guidelines for the Validation of Computerised Systems

- Guidelines for the Management of Electronic SOPs in a GLP Environment

- Guidelines for the Archiving of Electronic Raw Data in a GLP Environment

- Guidelines for the Acquisition and Processing of Electronic Raw Data in a GLP Environment

- Guidelines for the Development and Validation of Spreadsheets

- Guidelines for Change Management and Risk Assessment of Validated Computerized Systems in a GLP Environment.

- Position Paper 1: Is it acceptable to destroy the paper originals of raw data and related study documentation, if an image of the paper is captured in an electronic form (e.g. scanned)?

- Guidelines for Collaboration with External IT Service Providers Supporting a GLP Environment

ANNEX A: PIPETTE CONTROL SPREADSHEET

This example is based on MS Excel® 2003 SP3 and was re-tested with MS Excel® 2013.

Figure 3: Spreadsheet layout

The spreadsheet is divided into an input area (green), an output area (yellow) and an area for verification of constants and specifications (Figure 3).

The yellow and blue areas contain formulas in cells (Figures 4 and 5). To view the formulas: *Tools -> Options -> Tab: View* and check "Formulas" under *Window options*.

Figure 4: Formulas in cells, output area

Figure 5: Formulas in cells, area for verification of constants and specifications

Security and protection

The bright green input cells (B1, B2, B6–B15, B17) have to be unlocked (*Format -> Cells...*, see right) before the sheet is protected. After protection, the user is only able to enter values in these cells. All other cells on the sheet are locked (by default).

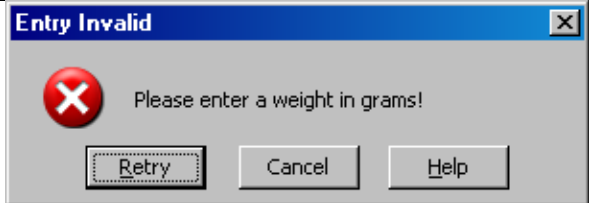The worksheet is protected (*Tools -> Protection -> Protect Sheet...*) with the password "agit". The workbook is protected (*Tools -> Protection -> Protect Workbook...*) with the same password.
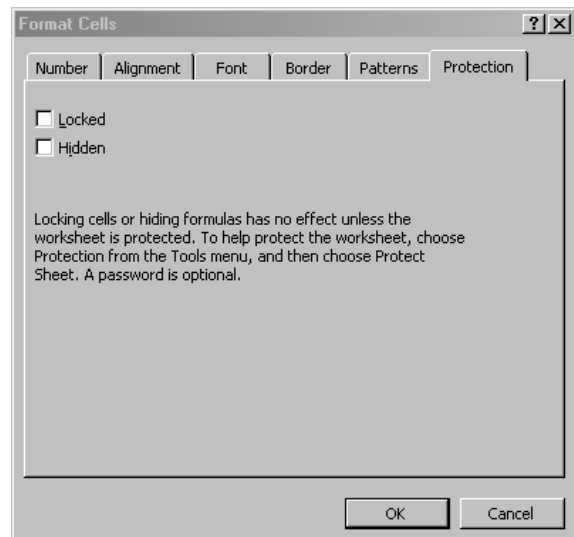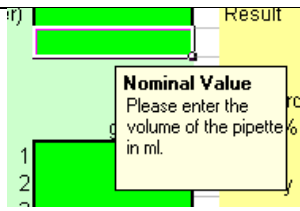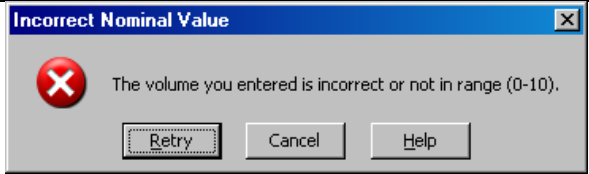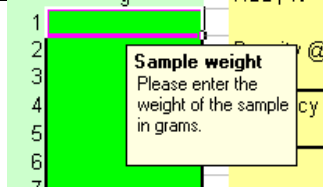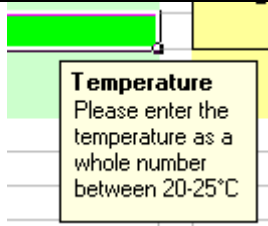
Data input

The user has to enter the data for the Inventory Number (B1), Nominal Value (B2), the ten sample weights (B6–B15) and the temperature of the water used (B17).

The cells use the data validation functionality in MS Excel® (*Data -> Validation...*). With this functionality, the input of invalid data (e.g. alphanumerical data in numerical fields, out-of-range values) can be intercepted.

Cell(s)	Data allowed	Input message	Error message
B1	No restriction (can be any combination of letters and numerals)		No error message

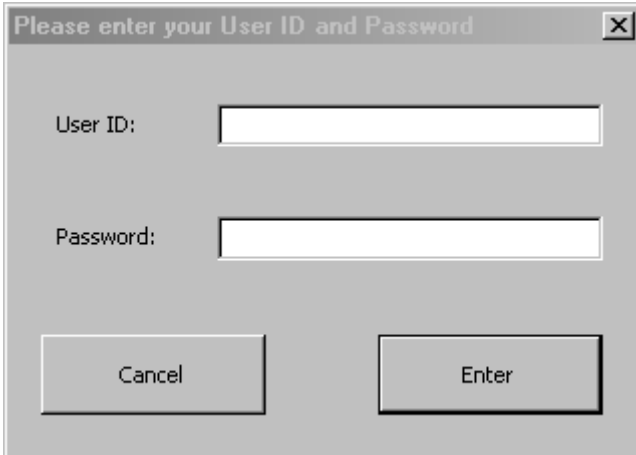| Cell(s)	Data allowed	Input message	Error message
B2	Decimal between 0 and 10		
B6–B15	Decimal >0		
B17	Whole number between 20 and 25		

Enter/Save

When the measurement has been performed, the user has to terminate the sheet by pressing the "Enter/Save" button.

If the "Enter/Save" button is pressed and not all green input cells have been filled out, the following error message appears:
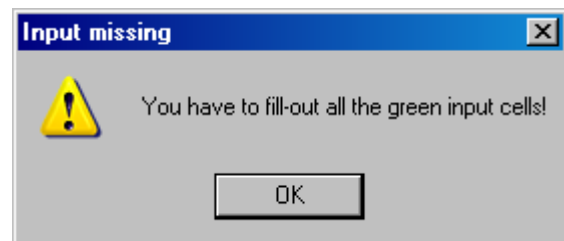
When the button is pressed, the following dialog box appears:

Figure 6: User Access Form

The "Users" sheet (which is hidden; *Format -> Sheet -> Hide* or *Unhide*) contains the UserID, password and user name of, in this case, three users.

Figure 7 UserIDs, passwords and user names on the "Users" sheet

When the green input cells are filled out but the user enters an incorrect UserID, the following error message appears:

When the user enters a correct UserID but an incorrect password, the following error message appears:

If the UserID and password are correct, the green input cells are locked and the "Enter/Save" button is disabled. The user name is entered in the field E17 and the date and time in field D17 (see Figure 8). The data can no longer be changed on the sheet. The MS Excel® workbook file can now be saved in a secure environment.

Figure 8: Screen after entering UserID and password.

VBA Code

The green texts in Figures 9–11 are comments to illustrate the VBA code.

Figure 9 shows the VBA code which is executed when the "Enter/Save" button is pressed on the worksheet. It is checked whether all the green input cells have been filled out. If this is the case, the user access form (Figure 6) is shown. If not all green input cells have been filled out, the error message "Input missing" (see above) is shown.

Figure 9: VBA code for the "Enter/Save" button

```
Private Sub btn_EnterSave_Click()
'definition variables
Dim MyCell As Range, CellisEmpty As Boolean

'initialisation
CellisEmpty = False

'checks if one of the cells is empty, if one or more cells
'are empty, a message is depicted
For Each MyCell In Range("B1:B2,B6:B15,B17")
    If MyCell = "" Then
        CellisEmpty = True
        Exit For
    End If
Next MyCell

If CellisEmpty = True Then
    MsgBox "You have to fill-out all the green input cells!", _
    vbExclamation, "Input Missing"
    Exit Sub
End If

usfAccess.Show

End Sub
```

Figure 10 shows the VBA code for the buttons on the user access form. When the user presses "Cancel", the form is hidden. If the "Enter" button is pressed, the subroutine "UserIDPasswordCheck" is executed.

Figure 10: VBA code for the buttons on the user access form

```
Private Sub btnCancel_Click()
usfAccess.Hide
End Sub


Private Sub btnEnter_Click()
usfAccess.Hide
UserIDPasswordCheck
End Sub
```

Figure 11 shows the VBA code for the UserID and password check. If the UserID or password is incorrect, the "Invalid UserID" and "Invalid Password" error messages appear (see above) and the sheet will not be protected; instead, the user access form is shown so that the UserID or password can be corrected. If both the UserID and the password are correct, all the cells on the worksheet are locked.

Figure 11: VBA code for the UserID/password check

```vba
Private Sub UserIDPasswordCheck()
Dim UserID_entered, UserName, Password_entered As String

'The UserID and Password are taken from the
'UserID/Password Dialog (usfAccess)
UserID_entered = usfAccess.tbUser_ID.Value
Password_entered = usfAccess.tbPassword.Value

With Worksheets("Users").Range("A:A")
    'Looks on the "Users sheet" if the user is defined (valid UserID)
    Set user = .Find(UserID_entered, LookAt:=xlWhole)
    If user Is Nothing Then
        MsgBox "You are not a valid user!", vbExclamation, _
        "Invalid UserID"
        'User can make corrections if UserID is incorrect
        usfAccess.Show
        Exit Sub
    End If
End With

'The password and user name of the correctly entered UserID
'are retrieved
User_password = user.Offset(0, 1).Value
UserName = user.Offset(0, 2).Value

'Checks if password is correct
If Password_entered = User_password Then
    'unprotect sheet to make changes
    Worksheets("Pipette_control").Unprotect Password:="agit"
    'Date and Time are entered into the cell D17
    Worksheets("Pipette_control").Range("D17").Value = Now
    'The user name is entered into cell E17
    Worksheets("Pipette_control").Range("E17").Value = UserName
    'The input cells (bright green) are locked to prevent from
    'changes after entering the data.
    Worksheets("Pipette_control").Range("B1:B2,B6:B15,B17").Locked = True
    'The button "Enter/Save" is disabled to prevent from changes
    'after pressing the button the first time.
    Worksheets("Pipette_control").btn_EnterSave.Enabled = False
    'protect sheet
    Worksheets("Pipette_control").protect Password:="agit", Contents:=True
    Else
    'If the password is incorrect the user is sent back to the
    'UserID/Password dialog
    MsgBox "Your password is incorrect!", vbExclamation, _
    "Invalid Password"
    usfAccess.Show
    Exit Sub
End If
End Sub
```