NPL Report CMSC 49/04

Report to the National Measurement System Directorate, Department of Trade and Industry

Use of Internet by Calibration Services: Demonstration of Technology

David Ives, Graeme Parkin, Joan Smith,
Michael Stevens, Julie Taylor and
Martin Wicks

March 2004

Use of Internet by Calibration Services: Demonstration of Technology

Graeme Parkin and Michael Stevens,
Centre for Mathematics and Scientific Computing

David Ives, Joan Smith, Julie Taylor and Martin Wicks
Centre for Electrical and Time Metrology

March 2004

ABSTRACT

This report describes the experience of creating a demonstration of a new calibration and measurement service for calibrating Optical Time Domain Reflectometers (OTDRs), which are used to measure optical fibre, to run over the internet. The report covers a description of the calibration and measurement service, the generic software to support the development of internet services and some guidelines on how to develop such services.

ISSN 1471-0005

National Physical Laboratory
Queens Road, Teddington, Middlesex, TW11 0LW

Approved on behalf of the Managing Director, NPL
by Dave Rayner, Head of the Centre for Mathematics and Scientific Computing

Contents

1 Introduction

This report describes the experience of creating a demonstration of a new internet-enabled calibration and measurement service for calibrating Optical Time Domain Reflectometers (OTDRs), which are used to measure properties of optical fibres. The report covers a description of the calibration and measurement service, the generic software to support the development of internet-enabled services and some guidelines on how to develop such services.

The work was carried out under the Software Support for Metrology project, on the *Use of Internet by Calibration Services: Demonstration of Technology*.

2 Rationale

At present, the majority of top-level calibrations in the NMS are carried out by transporting reference standards from a remote calibration laboratory to a standards laboratory, such as NPL. The spread of the internet as a communication medium, and the availability of high precision transportable measurement systems, open up an opportunity to address the dissemination of calibration values from national standards laboratories such as NPL in a different way. It is already clear that there will be a strong trend towards providing remote access over the internet to more and more calibration and measurement services.

NPL has previously carried out two projects in the electrical calibration area, one related to Automatic Network Analysers (ANAs), now a commercial service, and the other in the area of voltage and resistance standards.

The next generation of internet-enabled services will require generic software that can solve more complex measurement requirements, requiring software that can be applied to any instrument, and that will provide a solution for mobile/field applications.

OTDR was chosen to demonstrate the concepts of the next generation of internet-enabled services because:

- the calibration process requires at least four instruments, employing both RS232 and GPIB communication links
- an NPL calibrated reference fibre can be used as the user’s calibrated artefact
- OTDRs are common calibration devices used extensively in laboratories and in the field
- the benefits to the user are similar to other internet calibration services already established, (see section 3)
- the OTDR is probably the most commonly used equipment for testing optical communication links. It is used for evaluating installed optical links, locating faults, and monitoring quality for optical fibre production

3 Benefits

Internet calibration brings the following benefits:

- greatly reduced elapsed time for the whole process;
- direct access for more customers to the primary standards laboratory;
- the measurement standards belonging to the customer laboratory do not need to be transported;
- the standards are calibrated under normal conditions of use in the customer laboratory;
- the "down time" for the customer laboratory is kept to a minimum;
- the results can be reviewed on-line before the calibration is completed;
- instructions and procedures can be conveyed over the internet;
- the calibration conditions can be recorded e.g. using a video or digital camera;

- the expertise of the higher-level laboratory can be transferred to the customer laboratories staff;
- the potential to provide direct traceability to national standards whilst cutting down the calibration chain to one link;
- appropriately high levels of accuracy can be transferred to the customer's laboratory;
- reduced costs;
- a much better cost to accuracy ratio than for traditional calibrations;
- calibration can be performed at any time – day or night;
- it is not about de-skilling – calibration skills at the customer laboratory are more important than ever – hence the customer's calibration staff may need some special training, which is likely to be of real benefit to their normal work;
- data warehousing - calibration results should be stored by the calibration laboratory in a database for possible recall by customers during subsequent calibrations and audits. Data warehousing can also provide long-term access to calibration history and provide robust, paperless audit trails;
- guidance for troubleshooting;
- increased confidence in measurements;
- the measurement cycle can be suspended and resumed at another time.

In addition the use of internet-enabled calibration need not mean any loss of the assurance that comes from using accredited services. NPL has been working with UKAS on guidance on how such services are to be accredited.

4 Traceability to standards

When internet-enabled calibration is in operation, the user sends a stable calibration artefact to the calibration laboratory for characterisation. This calibrated artefact is then returned to the user and it will not have to be re-calibrated for a long period of time. The user then has controlled access to the laboratory's software algorithms and calibration procedure via the internet. These convert the artefact into a traceable standard against which equipment can be calibrated.

Measurement uncertainties are calculated in real-time by software running at the calibration laboratory.

5 Description of the OTDR service

A calibrated OTDR is used to measure the attenuation uniformity of an optical fibre. The OTDR is calibrated using the power reduction method. This method uses an attenuator to move a standard fibre through the dynamic range of the OTDR and compares attenuation values of 1 km sections of the fibre.

The calibration of an OTDR consists of the following aspects, described here mainly in terms of the equipment needed (for full details see NPL procedure [1]):

- **OTDR Operating Wavelength:-** The wavelength of the lasers in the OTDR is calibrated using a calibrated Optical Spectrum Analyser.
- **OTDR Distance scale:-** The distance scale of the OTDR is calibrated using the re-circulating loop technique. This requires a calibrated re-circulating loop.
- **OTDR linearity:-** The linearity of the OTDR is calibrated using an attenuator, a polarisation controller, a lead-in fibre and a long fibre.

- **OTDR Loss Scale Factor:-** The loss scale factor of the OTDR is calibrated using an attenuator, a polarisation controller, a short lead-in fibre, a long lead-in fibre and a calibrated fibre attenuation uniformity standard.
- **Thermometer:-** This is used to monitor the environmental temperature, and must be calibrated. The calibration can take up to twenty four hours to complete, and must be conducted within set temperature limits.

The OTDR can be calibrated with uncertainty budgets from the data collected using mathematical algorithms for straight line least squares fit, mean and standard deviation. Graphs are required to display some of the data.

All of the above must be capable of being controlled from a computer.

The internet service to be developed will:

1. Calibrate the OTDR in situ.
2. For a calibrated OTDR, correct any of its traces with respect to the calibration.
3. Store all data with respect to the OTDR in a database, including charging information.

6 Design and description of the software

6.1 Inputs to the design

6.1.1 NPL procedure and calibration software

The main input to the design of the internet service was the existing NPL procedure [1] covering the calibration process. This was supplemented by support from scientists who were experienced in carrying out calibrations and were familiar with the operation of the equipment.

Some software for performing the calculations was available, having been developed at NPL, and it was possible to adapt this for use. However, large parts of the procedure concerning the use of the instruments had only ever been run manually, so it was necessary to automate the control of the equipment in addition to driving the service remotely over the internet. Problems were resolved with the help of the relevant operating manuals and the experience of the scientists.

Hitherto, all the uncertainty budgets had only been calculated using spreadsheets, and it was necessary to determine the workings of the spreadsheets.

6.1.2 Instruments used in the calibration

Up to four instruments are required to calibrate the OTDR, with a maximum of four, including the OTDR, being connected to the computer at any one time. Two types of interface are required: RS 232 (the thermometer and the OTDR) and GPIB (all other instruments).

Difficulties were compounded because the particular OTDR and Optical Spectrum Analyser chosen for the project had not been used before at NPL, and there was no previous experience of driving them manually, let alone automatically. Furthermore, the commands used to control the instruments were complex, it being necessary to check their status before reading data.

The OTDR used was an Anritsu instrument, and the company provided support to the project team.

One requirement of the design was that the software should allow easy use of different varieties of instruments. In particular, it was desirable to write the software in such a way that different OTDRs could be added at a later date, allowing the user to select the model being used for a particular calibration.

6.1.3 Available software

Previous work by Adelard had resulted in some software called iCAL to control the iVR system (see [2] for some more details). This software consisted of a client and a server. Although much of the iCAL server was specific to the iVR calibration, it was developed to have a generic interface to the client software. However, it had only been tried on iVR, which required just one GPIB connection, one RS 232 connection, and used very simple commands to control the instruments. iCAL was developed with its own control language, which the server used to control the instruments by sending individual instructions to the client; this language had no control statements as they were not needed for iVR. The server software provided a useful starting point for the OTDR server software, but major changes were required to the generic parts, and the specific parts had to be replaced.

The iCAL client software was designed to be generic, and this proved to be the case, although it did need considerable development for this much larger project, to implement new and more complex interactions with the server, and to incorporate additional RS232 and GPIB interfaces.

The iCAL client and server software together with the related documentation were available to the project.

6.1.4 Connection to the internet through a mobile phone

For ease of use, and because mobile phone companies use fibre optics, it was required that the client PC be connected to the internet through a mobile phone. This in principle should have been easy, but in practice the transfer of data was error-prone and a simple protocol had to be implemented to overcome this by calling for data re-transmission where necessary.

6.2 Design

6.2.1 Overall architecture

The software is composed of a client, written in Microsoft Visual Basic [3], that runs on the computer located at the customer’s site, and a server program, written in PHP [4], that runs on a server computer at the calibration laboratory. The server controls the calibration process, while the client, using instructions received from the server (in the form of VB Script), controls the equipment and interacts with the operator. Data collected from the instruments is transmitted back to the server for processing, and the results are stored in a database.

Although the server software dictates how the calibration is conducted, the client software is in the driving seat throughout the process, and initiates all the communications. The client sends a message to the server and waits for a reply. On receipt of each message the server acts on the content and then issues a reply, see Figure 1.

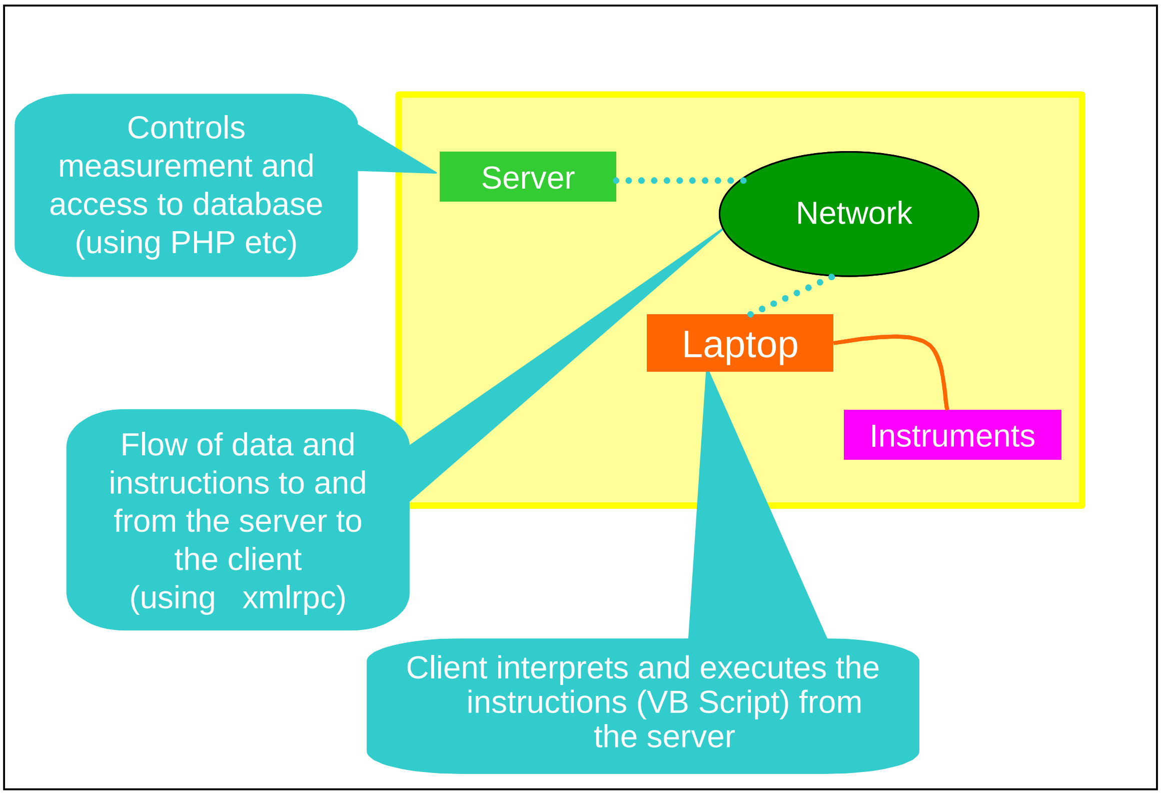
Controls measurement and access to database (using PHP etc)
Server
Network
Laptop
Instruments
Flow of data and instructions to and from the server to the client (using xmlrpc)
Client interprets and executes the instructions (VB Script) from the server

Figure 1. Overall archecture

6.2.2 Some implementation details

In the client, a Visual Basic control called INET is used to handle internet communications with the server. A function **execute** is called to send messages. It takes, as parameters, the address of the server, and, in XML format, the message to be sent. (A function for constructing XML was provided as part of the iCAL software.)

The iOTDR server is based on the iCAL server from Adelard. It uses a MySQL [5] database for storage, and the xmlrpc [4] protocol to communicate over the internet with the client, and this enables structurally complex and binary data to be exchanged.

The iCAL client software was designed to be generically applicable to a variety of calibrations, but it lacked the flexibility required for this more complex calibration, and significant improvements had to be made to it. In particular, it has been extended to enable it to talk to up to four RS232 ports and as many GPIB addresses as the National Instruments card can handle. A Visual Basic control for running VBScript [6] has been added. VB script is a subset of Visual Basic, and it can be sent down from the server as an ASCII string to be run on the client. The features provided by VB Script that are important for this project are that it can include calls to Visual Basic functions on the client that read data from, or write instructions, to the hardware; and that these function calls can be embedded in loops. Because of using VB script, fewer interactions with the server are required, and it is possible to make the calibration more efficient in terms of run time and internet traffic.

Figure 2 shows some VB Script to read the temperature. Figure 3 also shows some VB Script to record some OTDR traces.

```
Sub gettemp()
   'Get temperature reading
   dim Reply
   dim temp
   rs232.openrs232 $THERMport, "9600,N,8,1", true,false
   rs232.writers232text $THERMport, "MEASURE" + vbcrlf, vbcrlf
   rs232.ReadRS232 cint($THERMport), Reply
   temp = Split(Reply, ",", -1, 1)
   GPIB.FetchString temp(1),"temperature"
   rs232.closers232($THERMport)
end sub
```

Figure 2. Example of a command script (VB Script) to read the temperature

```
sub Run_proc(Trace,Fibre_len,Attnstep,Attn,Attnstart,IOR)
   'SUBROUTINE TO TAKE A SERIES OF TRACES AT DIFFERENT LEVELS
   Dim Attenuation
   Dim At
   Dim Pulse
   GPIB.pbar(Trace*($Averaging+6) + 10)
   'Send the number of traces that will be sent so the server
   'knows how much to unpack
   GPIB.FetchString cstr(Trace),"traces"
   rs232.writers232text $OTDRport,"ATT " + cstr(Attn) + vbcrlf,
   vbcrlf
   Attenuation=Attnstart
   'There will be Trace copies of all following items sent to
   'the server
   for At=1 TO Trace
     GPIB.info("Collecting trace " + cstr(At))
     'SET EXTERNAL ATTENUATOR
     GPIB.WriteGPIB $GPIBatt, "ATT " +
     FormatNumber(Attenuation, 2) + "dB"+ vbCrLf
     'CHECK THAT THE LASER IS OFF BEFORE CONTINUING
     rs232.writers232text $OTDRport,"STR 1"+ vbcrlf, vbcrlf
     rs232.write_until $OTDRport,"LD?"+ vbcrlf, vbcrlf, "LD 0"
     CALL Savetrace(At, Attenuation, Fibre_len)
     Attenuation = Attenuation + Attnstep
   next
end sub
```

Figure 3. Example of a command script (VB Script) to record OTDR traces

A number of generic functions have been implemented in the client. These are controllable using VB script. They are listed in Table 1.

Generic function	Description
CheckGPIB	Finds listeners on the bus and initiates the GPIB
WriteGPIB	Writes a string to a GPIB address
FetchGPIB	Reads from a GPIB address and returns the result to the server
ReadGPIB	Reads from a GPIB address to a local variable
OpenRS232	Opens an RS232 port
CloseRS232	Closes an RS232 port
WriteRS232	Writes a string to an RS232 port
FetchRS232	Reads from an RS232 port and returns the result to the server
ReadRS232	Reads from an RS232 port to a local variable
Fetchstring	Sends a string from a local variable to the server
Checkerror	Sends an error code, generated by any GPIB or RS232 function, to the server
Delay	Waits for a specified number of seconds
Info	Displays a message as a label at the top of the screen
pbar	Displays and starts a progress bar at the top of the screen
clearevents	Calls DoEvents, which cannot be called directly in VB scripts

Table 1. Client functions called by VB Script

6.2.3 Software development difficulties

Debugging proved difficult and time-consuming, both for PHP and for VB Script. PHP has no development environment. It is simply written as a text file and left in place for the server to execute the code.

When there are bugs in the PHP, the usual effect is for no response to be returned when the client sends a message. If this happens as soon as the client is started, it generally means there is a syntax error. The procedure here is to look at the PHP using a web browser, and the syntax error will be reported, with information that is of varying use in locating the fault. It pays to make only small changes to the PHP before checking it with the browser.

If the browser shows no fault, but, at some point during the calibration, there is no response from the server, then some other problem has occurred. Typical examples would be attempting to write to a MySQL table without first opening the database, or attempting to write to a file that is not open.

Sometimes strange effects were observed. Inadvertently putting a comment in the PHP text file, but outside the PHP begin and end brackets, had the effect of increasing the server response time roughly tenfold. The cause of this problem was uncovered when the XML returned to the client was examined.

By default PHP scripts have a 30 second timeout. This needs to be switched off because iOTDR scripts can run for considerably longer.

Errors in VB Script are not trappable. Syntax errors are detected as soon as an attempt is made to run the script, usually with an unhelpful message, such as “statement expected”, or “faulty script”, being reported. Run time errors have the effect of terminating the script immediately. It is not possible to detect if a script has been interrupted in this way or if it has terminated normally. It is recommended to build in a mechanism for determining that a script has run to its proper conclusion.

6.2.4 Demonstration

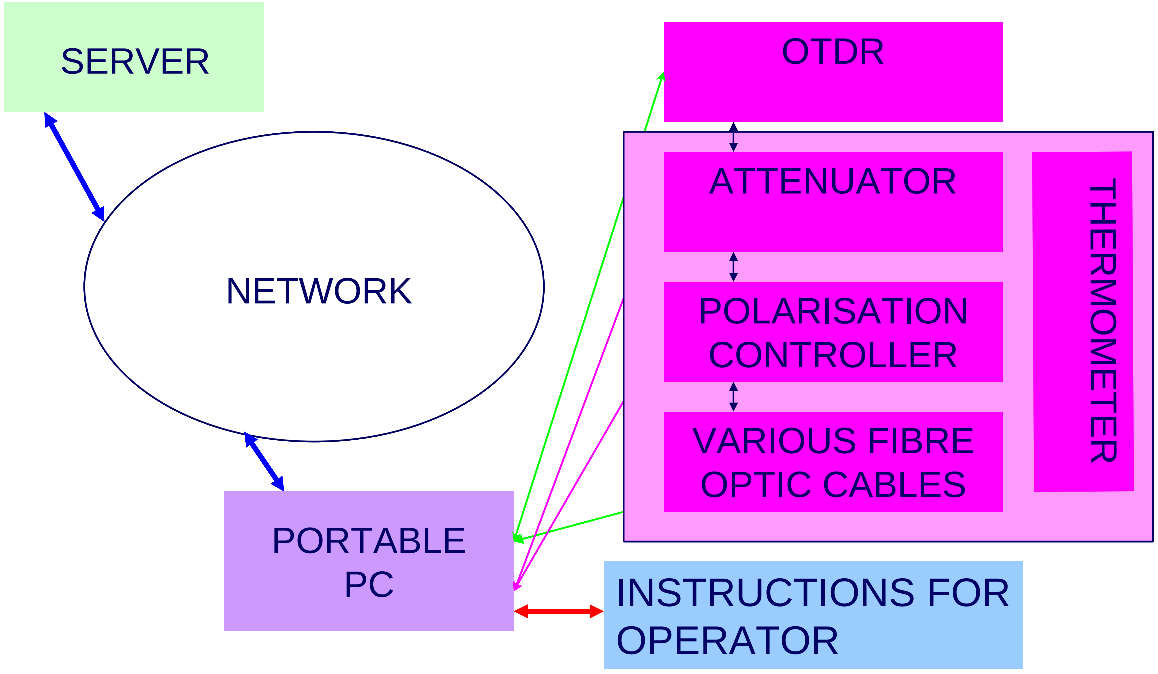

Figure 4. Set up required for one aspect of the calibration of an OTDR

Figure 4 shows the set up required to do measurement of OTDR linearity and loss scale factors, see section 4.

The green lines to the OTDR and Thermometer represent RS 232 connections. The purple lines to the Attenuator and Polarisation Controller represent GPIB connections.

A user of a service for OTDR calibration starts by establishing a link from a portable PC to the internet, using a mobile phone if required. Next the iOTDR client software must be started. The client automatically dials up the server over the internet, and establishes a dialogue. The user is required to log in, and a session identifier is allocated which will allow the user to return to the session later.

The server presents the user with a list of available services, and once a service has been selected, the server guides the user through it by sending instructions to the client software, which passes them on to the user, displaying them on the PC screen. Typical instructions would explain how to connect the instruments and cables. Once the equipment is ready and the server has been informed of this by the user, the server sends down commands (in the form of VBScript) to the client to control the measurements.

Before any part of the calibration can be performed, the user must identify the equipment in use. The specific items known by the server are present in drop-down lists for the user to make a choice. The user also specifies the RS232 and GPIB port numbers for the equipment. These numbers are stored on the database and are used by default for future calibrations.

Much of the calibration is temperature-critical, the allowed temperature range being stored by the server on the database. Before a calibration commences, a command is sent to the client to read the temperature from the thermometer. The reading is returned to the server, where it is compared with the stored maximum and minimum values. In the case where the reading lies outside the valid range, a message is sent to the client and the user is informed that the calibration cannot proceed.

All data collected is returned to the server and stored in the database. All analysis of the data is done on the server.

Faulty connections to equipment are detected. Error codes are sent from the client to the server, and meaningful error messages are displayed to the user on the screen. Care has been taken to deal with transmission faults, which are prone to arise when using the mobile phone link. Often a recovery can be effected without having to abort the calibration.

7 Achievements

Two pieces of software have been developed that enable a calibration of OTDRs to be conducted over the internet. Some of this software is generic, whilst some is specific to the calibration of OTDRs.

The client software, written in Visual Basic, is generic and can be used where connections to multiple instruments are required using GPIB or RS 232 interfaces, and where the commands to control those instruments use complex interactions. There is nothing specific about the OTDR calibration built into the client software, and it could readily be used for other calibrations. OTDR-specific code is not stored in the client, but is sent down from the server in the form of scripts for execution by the client.

The server software, written in PHP, contains much that is specific to the calibration of OTDRs, but the mechanism for communicating with the client software is generic, and the specific parts although not directly re-usable, provide a useful model for other calibrations.

A demonstration of this technology is available for OTDR’s and has been shown to various NPL staff and government and industry contacts.

The demonstration employs a generic network connection and has been successfully tested using a mobile phone link. This link is more error-prone than a traditional modem connection, but a recovery mechanism has been built into the software to enable the calibration to proceed in most circumstances.

8 Lessons for setting up an internet-enabled measurement service

The internet adds in several layers of complexity to a calibration, all of which can have problems and can be difficult to identify. The generic software uses four different languages, PHP, VB Script, Visual Basic and HTML (used to display instructions and results to the user). It also uses another protocol, xmlrpc. Currently there are no easy ways to debug in this mixed programming language environment.

In view of this complexity, the recommendation is to develop the control procedures that interact with the required instruments first, to develop a standalone automated calibration system next, and finally to adapt this for use over the internet.

In the future, consideration will have to be given to replacing the client software, as Microsoft have replaced Visual Basic with VB.NET. VB.NET could be used, but the rewrite would not be trivial.

9 Conclusions

This work has produced software, much of it generic, that can be used to develop internet-enabled metrology services. The software can be used as the basis for the control and calibration of any instrument or device thought to be suitable for such a purpose. NPL is pursuing ways to use this technology.

Demonstrations and presentations have been given to explain and show the practical use of the software for calibrating OTDRs. iOTDR could be launched as a service if a sufficient requirement is identified within the user community.

It is recommended not to develop an internet-enabled service until a standalone service has been shown to work.

The internet has proved to be a useful tool for improving accessibility to traceable measurements, and for encouraging measurement best practice to be used more widely in industry.

References

1. Ives, D.J., *Procedure QPCETM/B/403 - Fibre attenuation coefficient uniformity calibration*. 2001, National Physical Laboratory, Centre Electromagnetic and Time MetrologyQPCETM/B/403,

2. Rayner, D., *Survey of International Activities in Internet-enabled Metrology*. 2003, NPL,NPL Report CMSC 21/03, www.npl.co.uk/ssfm/download/#cmsc21_03.

3. MicroSoft, *Visual Basic*,http://msdn.microsoft.com/vbasic/

4. Lerdoff, R. and et al., *PHP: Hypertext Preprocessor*,http://www.php.net/

5. MySQL AB, *MySQL*,http://www.mysql.com/

6. MicroSoft, *VBScript*,http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/vtoriVBScript.asp