

**NPL REPORT
DEM-ES 008**

XML for Data Curation

Robin Barker

Not restricted

April 2006

XML for Data Curation

Robin Barker
Mathematics and Scientific Computing Group

April 2006

ABSTRACT

This is the report from the SSfM project on Data Curation, which investigated techniques for storage and manipulation of measurement data. The project concentrated on the use of XML to represent data in the measurement and calibration process, showing how the various forms of data could be presented as XML and how the XML data could be processed to produce other XML and finally a calibration certificate.

© Crown Copyright 2006
Reproduced with Permission of the Controller of HMSO
and Queen's Printer for Scotland

ISSN 1744-0475

National Physical Laboratory
Queens Road, Teddington, Middlesex, United Kingdom. TW11 0LW

Extracts from this report may be reproduced provided the source is
acknowledged and the extract is not taken out of context.

We gratefully acknowledge the financial support of the UK Department
of Trade and Industry (National Measurement System Policy Unit)

Approved on behalf of the Managing Director, NPL
by Jonathan Williams, Knowledge Leader of the Electrical and Software team

Contents

1.	Introduction.....	1
2.	XML for measurement – now and going forward	2
2.1	Ad-hoc formats for measurement data.....	2
2.2	Benefits	2
2.3	Future activities.....	4
3.	Case study on the use of XML.....	5
3.1	Data Curation project activities	5
3.2	XML as uniform approach to certificate data.....	5
3.3	Generic approach to XML and certificate production	5
4.	Conclusions.....	7
	References.....	7
Appendix A	XML – the technology for data curation	8
A.1	XML.....	8
A.2	XSLT	9
A.3	The Document Object Model (DOM)	9
A.4	XPath	10
A.5	XLink.....	10
Appendix B	Case study files and tools.....	11
B.1	Initial Data Files.....	11
B.2	XML data files	11
B.3	Document definition files	11
B.4	Document processing.....	12
B.5	Modelling the certificate production.....	12
Appendix C	XML files and formats.....	14
C.1	Example of measurement data XML.....	14
C.2	Example of fitting data XML.....	15
C.3	Example of results XML.....	16
C.4	Schema for raw measurement data	17
C.5	Schema for fitting data.....	20
C.6	Schema for results.....	21
C.7	DTD for raw measurement data.....	22
C.8	DTD for fitting data	23
C.9	DTD for results	24
C.10	Sample calibration certificate	24

1. Introduction

This is the report from the SSfM project *Data Curation* (2004-2006). The report describes the use of XML [5] to represent data in the measurement and calibration process, showing how the various forms of data could be presented as XML and how the XML data could be processed to produce other XML and finally a calibration certificate.

The aim of the project was to study techniques for the curation of measurement data, specifically calibration history data. “Curation” includes the complete management of the data, protection of its integrity and traceability, providing appropriate storage and access technology, and providing data mining and other appropriate analysis and information extraction techniques. Part of the project was to conduct a case study with another NMS programme to demonstrate applicability of the techniques.

Techniques that are required for data curation include database design, data storage, data mining, data warehousing and data analysis. Such techniques may find wider application in metrology than just to calibration history data. However, since calibration history data arises throughout metrology, this provides an appropriate generic application to focus this project. There is a need for tools and techniques for producing certificates or calibration history data from data provided in a common format. The report [1] from the SSfM project on *Protecting the data* (in the context of the use of the Internet by calibration services) recommends XML as a common format for calibration data, hence the project should focussed on the use of XML.

This report explains why the technologies based on XML are suitable for data curation, and describes the case study where XML was used to format measurement data. The report covers both deliverables from the project:

- techniques for data curation and processing calibration data;
- a case study with another NMS programme.

Section 2 describes the benefits of using XML for measurement data, section 3 describes the work of the Data Curation project, and section 4 gives conclusions and references. Appendix A explains XML and the associated tools and techniques; Appendix B explains the technical details of the case study ; and Appendix C contains the documents, schemas and templates produced by the case study, including a sample calibration certificate.

2. XML for measurement – now and going forward

2.1 Ad-hoc formats for measurement data

Measurement data is currently saved in many different formats causing interface problems with different analysis tools. The lack of any consistent format for data means that the generation of certificates is currently done in many diverse ways leading in many cases to rewrites every few years.

Problems arise from the use of ad-hoc storage, processing, transmission and presentation of data that exists throughout measurement science. The use of common formats for measurement data allows a common approach to the processing and rendering of calibration certificates and measurement data, and addresses the need to store data in a future-proof format.

What is required of common formats for measurement data is a method for formatting data that provides a robust, reliable, and repeatable approach to the storage, processing, transmission and presentation of measurement data. The format must be easily read and parsed by computers, easily used by bespoke applications, through the supported of generic tools. The format must also be readable by humans, not in a binary format that is may not be processable by applications in the future; human-readability will also allow the integrity to be ensured. The format should be easily transmitted over the internet (and other media) to allow sharing of measurement data between organisations; again the human-readability and the availability of generic tools if necessary for the receiving organisation to be able to understand the data. The format should be self-documenting so human readers can understand the data as it stands; but it should also be possible to define constrained documents type, to aid machine processing of the data.

The use of XML in the Data Curation project shows that XML provides an approach to structuring data that provides these requirements. Further work will aim to demonstrate an XML-based approach to the formatting of measurement data that can be used by most measurement application areas.

2.2 Benefits

XML provides a means to save data in a self-documenting standardised way, in a format that is one of the most future-proof available. XML is both human- and machine- readable, with a wide range of generic tools and software packages to support process of XML files. XML is compatible with many other technologies, in particular internet technologies and formats. XML supports definable document types (schemas) supported by tools and these schemas are themselves standardisable.

Appropriately defined schema for different formats of measurement data will aid future-proof storage, processing and investigation of data (data warehousing, data curation and data mining). Tools can use the XML format to implement the measurement process and it will be possible to render certificates, calibration data, and other measurement data in many different ways.

The use of XML will streamline measurement and calibration processes, make calibration certificate data more reliable and add value to the data available to measurement services customers. This also will make easier development of software tools to support scientific research. All areas of science will benefit from XML representation of their data.

The use of XML by national measurement institutes will directly support the Test and Measurement industry providing leadership and facilitating standards for the handling of measurement data. Transparent standards will stimulate competition in test and calibration services and create process efficiencies by facilitating inter-operation of different manufacturer's analytical instrumentation. .

The efficient handling of data and its curation is central to the *Information and Communication Technologies*. XML is also an essential technology for databases and information storage which play a critical role in *Bioscience and Healthcare*. The use of XML to represent measurement data support innovative use of that data through the use of generic XML tools and by permitting novel data analysis.

The curation and processing of calibration data is of great importance in metrology. The trend towards the electronic laboratory requires data technology that will guarantee the accessibility, security and understanding of data over a long period of time. XML is a future proof technology that will ensure that NMS data and measurement results relating to national standards will be meet these long-term requirements. The provision of good practice and the establishment of schemas will provide the infrastructure for the NMS and the Test and Measurement industry to embrace XML technology and lead the world in the standardisation of these techniques.

Further work on the application of XML to measurement data could:

- Enable the use of XML in measurement science, calibration and instrumentation.
- Provide a future-proof representation for measurement data that can be used for storage, processing and transmission of data.
- Demonstrate the use of XML formats for the better definition of common data formats.
- Allow for data be re-used, re-processed and more deeply investigated because it is represented in a standard format that can make use of existing tools and formats.
- Allow simpler, more repeatable generation of calibration certificates, avoiding transcription errors, making more efficient representations available to the customer.
- Facilitate moves towards greater use of XML in measurement science:
 - XML laboratory note book, for scientists;
 - XML calibration history of an instrument, for a calibration service;
 - XML log book, for an instrument, independent of the measurement services.

2.3 Future activities

Further work to enable the use of XML for the handling of measurement data should demonstrate the implementation of schemas and the processing of XML data to produce certificates, other calibration information, and measurement data for measurement service customers. The work would deal separately with data at different stages in measurement process.

The measurement data can be divided into

- raw and intermediate data, before it is in a form that can be made available externally;
- processed data, derived from the raw and intermediate data, that can be made available externally; and
- certificate data, data that appears on traditional calibration certificates derived from other processed data

There would be work to develop a mechanism for the provision of processed data to a customer through its representation as XML. The XML schemas will describe the various forms of processed data, which vary depending on the type of measurements to be represented. The tools will transform the processed data to calibration certificate data, and allow the customer to view/access their data (processed data downloaded from a server or delivered electronically).

There would be development of schema to describe all the information that appears on a traditional calibration certificate: customer details, methodology and calibration data; while allowing for greater detail. The XML certificate data will be able to be produced by transforming other XML measurement data, and in turn will be able to be viewed (rendered) on paper or on the web. The tool will create the calibration certificate as HTML or PDF.

There may be a requirement for many different schemas for raw measurement data and other intermediate stages of processing the data. A framework for these schemas will be developed using the XML Resource Development Framework (RDF). Example processing tools will be based on example XML representations of raw data or intermediate measurement data. The tools will transform the data to later stages in the measurement process, including the processed data that can be made available to the measurement service customer.

3. Case study on the use of XML

3.1 Data Curation project activities

The Data Curation project investigated techniques for the use of XML for Data Curation and carried out a case study on the use of XML. The techniques are described in Appendix A. The case study was based on data from the NMS Thermal programme. There were several stages to work of the case study.

- Represent the data as XML;
- Define XML formats to capture the format of the example XML data files, and validate the example XML data files against the formats;
- Show how the XML data files could be read, processed and output as other XML data formats;
- Show how the XML data files could be used as a basis for program to produce calibration certificates (as PDF).

Appendix B gives a detailed description of the files and formats and tools used in the case study.

3.2 XML as uniform approach to certificate data

The final step in the case study was to use the XML files as the basis of a calibration certificate. NPL produces many calibration certificates from different measurement services and there are many different processes for producing the certificates. There is also a growing desire to make available electronically, the data contained in the calibration certificates. We see the use of XML files containing the calibration information as a way of allowing a uniform approach to the process of producing certificates and a format for making available the certificate data; this is the approach outlined in the report [1] from the SS/M-2 project on *Protecting the data*.

The approach in the case study was based on the example XML files already created and on a form of calibration certificate from the thermal metrology area. The certificate includes data on the fitted polynomial used to produce calibration data and the calibration table itself. The certificate contains other information: details of the customer and the instrument, and information on the measurement and modelling processes used to produce the calibration data.

3.3 Generic approach to XML and certificate production

Based on the case study in one metrology area, we can identify a generic approach to the use of XML for measurement data to be used in a uniform approach to calibration certificates.

- 1) XML formats for raw data – these may be different for each measurement process
- 2) Programs to fit a model to the raw data – probably a different program for each raw data format and each fitted model

- 3) XML formats for fitting information – different format for each model
- 4) Programs to calculate calibration tables from a fitted model – different programs for each model
- 5) XML format for calibration certificate data
- 6) XML format for calibration customer/information data
- 7) Template text for each measurement service calibration certificate
- 8) Program to produce calibration certificates from items (3), (5), (6), and (7).

Ideally there would be just one program, with different options to handle the different fitting models in (3).

4. Conclusions

This report has shown that XML provides a number of techniques/applications that can be used for the data curation of measurement data, certificate data and calibration history data. In particular, XML-based files of measurement data can be processed to other give other forms of the measurement data and produce calibration certificates.

This project has addressed some issues of dealing with structured numerical (measurement) data. As more techniques to manage and process structured numerical data become available, the more (measurement) scientists will impose some structure on their numerical (measurement) data. This will allow this data to be managed and structured when it is produced or in the future. XML allows the structure of the data to be identified without knowledge of what will be done to the data in the future.

The size of structured measurement datasets will grow: from one XML document representing the data for one calibration of one instrument, to sets of XML documents including all the data for the all the calibrations of all the instruments for a range of measurements services over a period of decades. Even bigger datasets will be produced by measurement processes involving frequent measurements from multiple sensors. Some XML techniques will scale to allow investigation of these large datasets, but other techniques for investigating large datasets may require pre-processing from XML to some application-specific format before the real processing can be done effectively.

References

1. Barker, R.M. and G.I. Parkin. *Use of the Internet for Calibration Services – Protecting the Data – Final Report*. NPL Report CMSC 28/03, NPL, July 2003.
http://www.npl.co.uk/ssfm/download/documents/cmssc28_03.pdf
2. Gardner, S.C. *Software Support for Metrology – Good Practice Guide No. 14 Guidance and Tools for Interactive Web Pages*. NPL Report DEM-ES 002, NPL, April 2005.
<http://www.npl.co.uk/ssfm/download/documents/ssfmbpg14.pdf>.
This guide contains extensive list of reference and web resources for XML technologies. Below are included basic references for key technologies.
3. libxml2, *The XML C parser and toolkit of Gnome*. <http://www.xmlsoft.org>
4. W3C, *Document Object Model (DOM)*. <http://www.w3.org/DOM/>
5. W3C, *Extensible Markup Language (XML)*. <http://www.w3.org/XML/>
6. W3C, *XML Linking Language (XLink)*. <http://www.w3.org/TR/xlink/>
7. W3C, *XML Path Language (XPath)*. <http://www.w3.org/TR/xpath>
8. W3C, *XSL Transformations (XSLT)*. <http://www.w3.org/TR/xslt>

Appendix A XML – the technology for data curation

This section shows how the various XML technologies fulfil the needs of data curation; more details about XML and the supporting technologies can be found in [2]. It covers:

- Why XML? – readable, self-documenting, machine processable
- XSLT – transforming XML
- DOM – modelling XML for interactive manipulation
- XPath – accessing XML components
- XLink – linking from XML to other resources

A.1 XML

Examples of XML can be seen in Appendix C. The files are text files: they can be read by humans and edited with text editors, and are not tied to any proprietary applications.

After some header information, the elements of an XML documents are matching pairs of tags

```
<data> ... </data>
```

The content of the element is the text between the tags; it contains further nested elements delimited matching pairs of tags and other text. (In the other text, ‘<’ and ‘>’ are represented as ‘<’ and ‘>’ – so they do not appear to be delimiting tags.) Elements can also include attributes: these are **name=value** pairs that are included in the opening tag (before the ‘>’). The values are text strings in quotes (the text does not contain elements).

In this example, the element `point` includes the two elements `x` and `y`, which in turn contain text (decimal number). Element `point` has an attribute `label` and the elements `x` and `y` have attributes `name`.

```
<point label="a">
  <x unit="oC">      2203.02      </x>
  <y unit="A">       41.1834      </y>
</point>
```

XML structure

Intelligent naming of elements (and attributes) makes the XML content self-documenting: without other documentation, the intended meaning of XML elements should be evident. The structure of an XML document is such that it can be read/parsed by a computer to give an abstract data structure of nested elements and content, this structure can be used to process the document.

The structure of an XML document is important for the processing of the document. If the structure is described then a given example of a document can be checked to see if it conforms to the description. This allows those (human or machine) that create the XML document in a given format to know what structure is expected, and it allows those that process the document to know what structures to expect.

The simpler form of XML document description is provided by a DTD (Document Type Definition): this describes the contents of each element, which elements it may contain, how many, and in what order; and it also describes what attributes and element may have. There are examples of DTDs in Appendix C.

The richer form of XML document description is the XML schema definition: an XML format for describing XML formats. This allows more precise definition of the content of XML elements and the types of textual data that can occur. Because XML schema definitions are themselves XML documents, they can be processed by other XML tools, and edited using XML editors. Again, there are examples of XML schema definitions in Appendix C.

A.2 XSLT

XSLT (Extensible Stylesheet Language Transformations [8]) specify how to transform one XML document into another and form a core technology in handling XML documents in different formats. XSLT was originally intended for transforming XML documents into forms that they could be viewed or presented, e.g. as HTML. In this context, stylesheets (e.g. CSS – cascading stylesheets) are ways of specifying the presentation style of a document.

However, XSLT is more powerful than CSS and general in scope, because of the range of available XML formats. For example, an XSLT stylesheet could transform an XML dataset into an SVG image of a graph representing that dataset.

The importance for data curation of XSLT is that it allows us to transform existing data in old XML format into new or different format – not necessarily an XML format. XSLT is a standard XML-defined language, so converting between XML formats will always be possible as long as XSLT is supported. XSLT provides access to your old XML data even if you change formats in the future.

XSLT also maintains its original purpose of providing presentation of XML documents. A good choice of target format for XSLT stylesheets is XSL-FO (Extensible Stylesheet Language Formatting Objects): an XML document containing information together with instructions on how to lay out its contents in pages. Having transformed a document to XSL-FO, tools exist to convert XSL-FO to PDF. For example, the DocBook markup language for technical documents defines an XML schema and provides an XSLT stylesheet mapping the DocBook XML source to XSL-FO, from which it can be converted to PDF.

A.3 The Document Object Model (DOM)

DOM [4] describes a standard interface for software (e.g. browsers) to access and manipulate XML documents. Scripts running in DOM-compliant browsers can manipulate web pages dynamically. Since DOM is applicable to any XML technology, it can, for example, construct mathematics (MathML) or graphic (SVG) as XML on the fly, in response to user actions.

The specifications for DOM are for *ECMAScript* and Java implementations, but libraries providing DOM interfaces exist for several other languages too, including PHP, Perl and C. In the context of Web browsers, it is the *ECMAScript* interface that is most relevant. DOM can be used to provide dynamic access to XML documents; in the context of data curation for measurement/calibration data,

it provides access to the measurement data sets (formatted as XML) where the user can interact with the data.

The part of the DOM that specifies which elements can trigger changes is called the “Event Model”. Events are generated by actions in a browser such as moving the mouse over an element, loading a page, submitting a form, clicking on a form input field, and so on. The triggering of an event can be keyed to a function written in a scripting language, so that for a specific event, such as a mouse click, the page is changed dynamically in some way, such as starting an animation for instance.

A.4 XPath

XPath [7] is a language for addressing parts of an XML document, designed to be used by XSLT and XPointer (a method for linking to parts of an XML document). The XPath language is supported by many programming languages and can be used to manipulate and extract data from an XML document. XPath has elements of wildcard file name syntax and regular expression syntax.

In the example in section A.1, the value of the x-coordinate of the point, could be addressed as

```
*/point/x
```

This could be used to return the data value 2203.02, or the value of the attribute unit: “oC” (for degrees celsius).

XPath expressions are used to address element and combine their value, for instance, using arithmetic operators; XSLT supports the evaluation of these XPath expressions.

For data curation, the value of XPath is that it is robust and flexible: this allows data to be addressed without necessarily knowing the full structure of the document. So if a collection of datasets is stored in number of different but closely related format, it may be possible to address the important data in all data sets using the same XPath expression.

A.5 XLink

XLink [6] is an extension to XML of the linking mechanism of HTML. In HTML, <a> elements always denote outbound links to other resources but XLink also supports inbound and third-party links. Third-party links provide a number of “arcs”, each of which links any number of resources. XLink provides a mechanism to support “linkbases” directories of links that can be published independently of the resources from which links are traversed.

For data curation, XLink allows links between related data in different XML documents to be created and managed, increasing the value of data that can be retrieved starting from any one document.

Appendix B Case study files and tools

B.1 Initial Data Files

The initial data files were plain text files describing three stage of the measurement/ calibration process.

- Raw data files, points (x, y) that were measured in the calibration process;
- Fitted data files, coefficients of the fitted polynomial and information on fitting and residuals at the measured points;
- Calibration data files, the fitted polynomial evaluated at a range of points to be used in a calibration table on the calibration certificate.

The files are human-readable with various labels to indicate the significance of the data and other comments to explain how the data is to processed.

B.2 XML data files

The process of creating XML files and formats from the initial data started by putting XML tags and other formatting round the data from the initial data files. This involves choices as to whether data should be XML elements or attributes on those elements. These decisions were made based on which data was more important and would be used in onwards processing and which data was less important for further processing.

As the work progressed there were some changes to these the XML files, mainly to ensure that the XML formats were more easily documented. The final version of the XML files for the raw data file, fitted data file, and the calibration data file are given in Appendix C (sections C.1, C.2, and C.3).

B.3 Document definition files

Using the XML data files as generic examples, two sets of document definition files were developed. The document formats were defined using both DTD and XML schema definitions.

The DTDs were developed by describing the elements and attributes that occurred in the documents, assuming that the example files demonstrated the most general structure for the elements. Attributes that occurred in the examples were allowed but marked as optional (“#IMPLIED”) since there were either obvious default values or the values were not needed to process the documents. A tool (xmllint [3]) was used to check the example XML files were valid against the DTD given in the XML header. This allowed syntactic errors in the XML and DTD files to be corrected. The final DTD files describing the raw data, fitting data and the calibration results are given in the appendix (sections C.7, C.8, and C.9)

The schema definitions were defined by following the example XML files and supplying types for data where this was suggested by the scientific semantics of the original data. Again the xmllint tool was used to check the example XML files were valid against the schema, supplied as an argument to the tool. After

corrections to meet the exacting XML schema definition format, the final XML schema definitions for the raw data, fitting data and the calibration results were produced, as given in the appendix (sections C.4, C.5, and C.6).

B.4 Document processing

The next step in the case study was to show that the XML documents could be processed to produce other XML document representing the processed data.

The first process is to take the raw data and fit a polynomial to the data, recording the coefficients and information on the residuals. The raw data input is given in the XML raw-data format (defined by sections C.2/C.7) and the fitting data will be returned in the XML fitting format (defined by sections C.5/C.8). The program had to be able to parse and create XML and to be able to polynomial fitting: this is not a combination functionality that is readily available in any one programming language. So instead we chose separate languages for the two different components of required functionality: Perl for the XML handling, and Fortran for the numerical calculations.

The Fortran programs for fitting a polynomial and evaluating were simple wrappers around the NAG library routines, adapted so enable the routines to be called simply from Perl. The main Perl program (`fitting.perl`) used modules `XML::Parser` (to read the data from the XML) and `XML::Tree` (to create the new XML fitting information). In between the call to these to modules, the data is passed to the Fortran programs to calculate the fitted coefficients and the residuals. The demonstration program was tested by running it on the `example_of_raw_data.xml` file to create a new XML file: this file was validated using `xmllint` against the fitting XML schema and DTD (sections C.5/C.8).

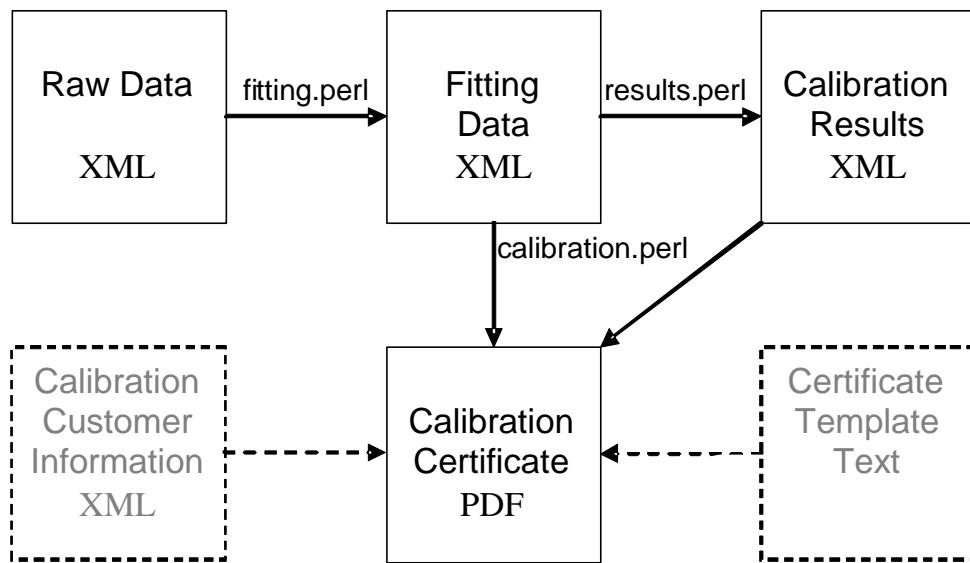
The program (`results.perl`) to derive calibration results from fitting information can be constructed similarly, again passing the data to a Fortran program to evaluate the fitted polynomial at the points required for the calibration table. The output of the program can be validated against the XML schema and DTD for calibration results (sections C.6/C.9).

B.5 Modelling the certificate production

We modelled the components of the calibration certificate by using the existing fitting and calibration XML formats for the data, another XML format to represent the customer and instrument information, and a further file as the template for the fixed text and layout in the certificate (fixed for a particular calibration service). The data and template are used to create a *TeX* file, which includes all the layout information and the *TeX* file is processed by `pdflatex` to produce a PDF certificate.

The process of putting the information together is in principle straightforward but there are many details to be solved to cope with the layout of arbitrary amounts of data. The Perl program (`certificate.perl`) handles all the reading of XML files and creating the *TeX* file based on the layout template, and the final processing as PDF. The program is broken down into a number of modules (in the `Certificate::` namespace) to handle the different aspects of process and layout.

The final certificate produced from some example files are shown in the final appendix section C.10.



Processes for XML-based measurement data and certificate production

Appendix C XML files and formats

C.1 Example of measurement data XML

```

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE raw-data SYSTEM "raw_data.dtd">
<raw-data info="PC870; TRL30-209; int=10; Lprt=1600; Uprt=2200; 0;" >
<points>
  <fitting list="1;0;">7</fitting>
  <sum-table list="1;">7</sum-table>
</points>
<data>
  <point label="a"><x name="oC">2203.02</x><y name="A">41.1834</y></point>
  <point label="b"><x name="oC">2103.85</x><y name="A">38.3148</y></point>
  <point label="c"><x name="oC">2004.27</x><y name="A">35.5338</y></point>
  <point label="e"><x name="oC">1803.05</x><y name="A">30.2331</y></point>
  <point label="d"><x name="oC">1901.79</x><y name="A">32.7789</y></point>
  <point label="f"><x name="oC">1703.36</x><y name="A">27.7714</y></point>
  <!-- LAST pair is J -->
  <point label="g"><x name="oC">1603.30</x><y name="A">25.4252</y></point>
</data>
<!-- CHEBYSHEV bds NEXT -->
<bounds>
  <min>1590</min><max>2210</max>
</bounds>
<order-data>
  <order-variables>
    <order points="seven">FOUR</order>
    <variables>
      <var>HSV_ten</var><var>HSG</var>
      <var>BBV_ten</var><var>CEG</var>
    </variables>
  </order-variables>
  <order-variables>
    <order points="eight">FIVE</order>
    <variables>
      <var>BBV_eight</var><var>BBG</var><var>CEV</var>
    </variables>
  </order-variables>
  <order-variables>
    <order points="ten">SIX</order>
    <variables><var>HSV_seven</var></variables>
  </order-variables>
  <order-bounds>
    <min info="t;o">4</min>
    <max note="max=min">4</max>
  </order-bounds>
</order-data>
<output list="0;1;1;1;1;62;">
  <coe list="0;10;">0;10;</coe>
  <x name="oC">4;2;</x>
  <y name="A" list="1;4;">2;4;</y>
  <res list="0;3;2;3;">1;4;</res>
  <rms list="0;">1;6;</rms>
</output>
</raw-data>

```

C.2 Example of fitting data XML

```

<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE fitting SYSTEM "fitting.dtd">
<fitting info="PC870; TRL30-209; int=10; Lprt=1600; Uprt=2200; 0;">
<!-- Lower and upper bounds as given by user -->
<bounds><min>1590.0</min><max>2210.0</max></bounds>
<polynomial>
  <order>4</order>
  <coeffs>
    <coeff degree="0"> 0.6598616270E+02</coeff>
    <coeff degree="1"> 0.8139620292E+01</coeff>
    <coeff degree="2"> 0.2618172517E+00</coeff>
    <coeff degree="3">-0.6420096973E-02</coeff>
    <coeff degree="4"> 0.1110595691E-02</coeff>
  </coeffs>
</polynomial>
<data>
  <point number="1">
    <x>2203.02</x> <y>41.1834</y>
    <fitted>41.1835</fitted> <residual> 0.0001</residual>
  </point>
  <point number="2">
    <x>2103.85</x><y>38.3148</y>
    <fitted>38.3144</fitted><residual>-0.0004</residual>
  </point>
  <point number="3">
    <x>2004.47</x><y>35.5338</y>
    <fitted>35.5340</fitted><residual> 0.0002</residual>
  </point>
  <point number="4">
    <x>1901.79</x><y>32.7789</y>
    <fitted>32.7795</fitted><residual> 0.0006</residual>
  </point>
  <point number="5">
    <x>1803.05</x><y>30.2331</y>
    <fitted>30.2320</fitted><residual>-0.0011</residual>
  </point>
  <point number="6">
    <x>1703.36</x><y>27.7714</y>
    <fitted>27.7721</fitted><residual> 0.0007</residual>
  </point>
  <point number="7">
    <x>1603.30</x><y>25.4252</y>
    <fitted>25.4250</fitted><residual>-0.0002</residual>
  </point>
</data>
<residuals>
  <sum-of-squares>0.000002</sum-of-squares>
  <rms>0.001093</rms>
  <maxima>
    <positive>
      <residual percent="0.00%"> 0.000723</residual>
      <number>6</number>
    </positive>
    <negative>
      <residual percent="0.00%">-0.001132</residual>
      <number>5</number>
    </negative>
  </maxima>

```

```
</residuals>
</fitting>
```

C.3 Example of results XML

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE results SYSTEM "results.dtd">
<results>
<device>
  <type>LAMP</type>
  <number>PC870</number>
  <reference>TRL30-209</reference>
</device>
<labels>
  <label>Temp.</label>
  <label>Current</label>
  <label>Slope</label>
</labels>
<units>
  <unit>C</unit>
  <unit>A</unit>
  <unit>A/ C</unit>
</units>
<table>
  <tr><td>1600</td><td>25.350</td><td>0.0228</td></tr>
  <tr><td>1610</td><td>25.578</td><td>0.0230</td></tr>
  <tr><td>1620</td><td>25.808</td><td>0.0231</td></tr>
  <tr><td>1630</td><td>26.039</td><td>0.0232</td></tr>
  <tr><td>1640</td><td>26.271</td><td>0.0234</td></tr>
  <tr><td>1650</td><td>26.505</td><td>0.0235</td></tr>
  <tr><td>1660</td><td>26.739</td><td>0.0236</td></tr>
  <tr><td>1670</td><td>26.976</td><td>0.0237</td></tr>
  <tr><td>1680</td><td>27.213</td><td>0.0239</td></tr>
  <tr><td>1690</td><td>27.451</td><td>0.0240</td></tr>
  <tr><td>1700</td><td>27.691</td><td>0.0241</td></tr>
  <tr><td>1710</td><td>27.932</td><td>0.0242</td></tr>
  <tr><td>1720</td><td>28.175</td><td>0.0243</td></tr>
  <tr><td>1730</td><td>28.418</td><td>0.0245</td></tr>
  <tr><td>1740</td><td>28.663</td><td>0.0246</td></tr>
  <tr><td>1750</td><td>28.908</td><td>0.0247</td></tr>
  <tr><td>1760</td><td>29.155</td><td>0.0248</td></tr>
  <tr><td>1770</td><td>29.404</td><td>0.0249</td></tr>
  <tr><td>1780</td><td>29.653</td><td>0.0250</td></tr>
  <tr><td>1790</td><td>29.903</td><td>0.0252</td></tr>
  <tr><td>1800</td><td>30.155</td><td>0.0253</td></tr>
  <tr><td>1810</td><td>30.408</td><td>0.0254</td></tr>
  <tr><td>1820</td><td>30.662</td><td>0.0255</td></tr>
  <tr><td>1830</td><td>30.917</td><td>0.0256</td></tr>
  <tr><td>1840</td><td>31.173</td><td>0.0257</td></tr>
  <tr><td>1850</td><td>31.430</td><td>0.0258</td></tr>
  <tr><td>1860</td><td>31.688</td><td>0.0259</td></tr>
  <tr><td>1870</td><td>31.948</td><td>0.0260</td></tr>
  <tr><td>1880</td><td>32.208</td><td>0.0262</td></tr>
  <tr><td>1890</td><td>32.470</td><td>0.0263</td></tr>
  <tr><td>1900</td><td>32.732</td><td>0.0264</td></tr>
  <tr><td>1910</td><td>32.996</td><td>0.0265</td></tr>
  <tr><td>1920</td><td>33.261</td><td>0.0266</td></tr>
  <tr><td>1930</td><td>33.527</td><td>0.0267</td></tr>
  <tr><td>1940</td><td>33.794</td><td>0.0268</td></tr>
  <tr><td>1950</td><td>34.062</td><td>0.0269</td></tr>
```

```
|<td>1960</td><td>34.331</td><td>0.0270</td></tr>
|<td>1970</td><td>34.601</td><td>0.0271</td></tr>
|<td>1980</td><td>34.872</td><td>0.0272</td></tr>
|<td>1990</td><td>35.144</td><td>0.0273</td></tr>
|<td>2000</td><td>35.417</td><td>0.0274</td></tr>
|<td>2010</td><td>35.691</td><td>0.0275</td></tr>
|<td>2020</td><td>35.966</td><td>0.0276</td></tr>
|<td>2030</td><td>36.243</td><td>0.0277</td></tr>
|<td>2040</td><td>36.520</td><td>0.0278</td></tr>
|<td>2050</td><td>36.798</td><td>0.0279</td></tr>
|<td>2060</td><td>37.078</td><td>0.0280</td></tr>
|<td>2070</td><td>37.358</td><td>0.0281</td></tr>
|<td>2080</td><td>37.639</td><td>0.0282</td></tr>
|<td>2090</td><td>37.922</td><td>0.0283</td></tr>
|<td>2100</td><td>38.205</td><td>0.0284</td></tr>
|<td>2110</td><td>38.489</td><td>0.0285</td></tr>
|<td>2120</td><td>38.775</td><td>0.0286</td></tr>
|<td>2130</td><td>39.061</td><td>0.0287</td></tr>
|<td>2140</td><td>39.349</td><td>0.0288</td></tr>
|<td>2150</td><td>39.637</td><td>0.0289</td></tr>
|<td>2160</td><td>39.927</td><td>0.0290</td></tr>
|<td>2170</td><td>40.217</td><td>0.0291</td></tr>
|<td>2180</td><td>40.509</td><td>0.0293</td></tr>
|<td>2190</td><td>40.801</td><td>0.0294</td></tr>
|<td>2200</td><td>41.095</td><td>0.0000</td></tr>
</table>
</results>

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

|  |

```

C.4 Schema for raw measurement data

```

<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:annotation>
    <xsd:documentation>
      Raw measurement data.
      Robin Barker 2005
    </xsd:documentation>
  </xsd:annotation>

  <xsd:element name="raw-data" type="RawData"/>

  <xsd:complexType name="RawData">
    <xsd:sequence>
      <xsd:element name="points" type="PointsData"/>
      <xsd:element name="data" type="Points"/>
      <xsd:element name="bounds" type="Bounds"/>
      <xsd:element name="order-data" type="OrderData"/>
      <xsd:element name="output" type="Output"/>
    </xsd:sequence>
    <xsd:attribute name="info" type="xsd:string"/>
  </xsd:complexType>

  <xsd:complexType name="DecimalList">
    <xsd:simpleContent>
      <xsd:extension base="xsd:decimal">
        <xsd:attribute name="list" type="xsd:string"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>

```

```

<xsd:complexType name="StringList">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="list" type="xsd:string"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="PointsData">
  <xsd:sequence>
    <xsd:element name="fitting" type="DecimalList"/>
    <xsd:element name="sum-table" type="DecimalList"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="FloatName">
  <xsd:simpleContent>
    <xsd:extension base="xsd:float">
      <xsd:attribute name="name" type="xsd:string"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="Points">
  <xsd:sequence>
    <xsd:element name="point" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="x" type="FloatName" nillable="true"/>
          <xsd:element name="y" type="FloatName" nillable="true"/>
        </xsd:sequence>
        <xsd:attribute name="label" type="xsd:string"/>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Bounds">
  <xsd:sequence>
    <xsd:element name="min" type="xsd:float"/>
    <xsd:element name="max" type="xsd:float"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="OrderData">
  <xsd:sequence>
    <xsd:element name="order-variables" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="order">
            <xsd:complexType>
              <xsd:simpleContent>
                <xsd:extension base="xsd:string">
                  <xsd:attribute name="points" type="xsd:string"/>
                </xsd:extension>
              </xsd:simpleContent>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="variables">

```



```

        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="var" type="xsd:string" maxOccurs="unbounded" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="order-bounds">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="min">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:decimal">
              <xsd:attribute name="info" type="xsd:string"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="max">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:decimal">
              <xsd:attribute name="note" type="xsd:string"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Output">
  <xsd:sequence>
    <xsd:element name="coe" type="StringList"/>
    <xsd:element name="x">
      <xsd:complexType>
        <xsd:simpleContent>
          <xsd:extension base="xsd:string">
            <xsd:attribute name="name" type="xsd:string"/>
          </xsd:extension>
        </xsd:simpleContent>
      </xsd:complexType>
    </xsd:element>

    <xsd:element name="y">
      <xsd:complexType>
        <xsd:simpleContent>
          <xsd:extension base="xsd:string">
            <xsd:attribute name="list" type="xsd:string"/>
            <xsd:attribute name="name" type="xsd:string"/>
          </xsd:extension>
        </xsd:simpleContent>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```

    <xsd:element name="res" type="StringList"/>
    <xsd:element name="rms" type="StringList"/>
  </xsd:sequence>
  <xsd:attribute name="list" type="xsd:string"/>
</xsd:complexType>

</xsd:schema>

```

C.5 Schema for fitting data

```

<?xml version="1.0" encoding="utf-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation>
      Fitted measurement data with coefficients and residuals.
      Robin Barker 2005
    </xsd:documentation>
  </xsd:annotation>
  <xsd:element name="fitting" type="Fitting"/>
  <xsd:complexType name="Fitting">
    <xsd:sequence>
      <xsd:element name="bounds" type="Bounds"/>
      <xsd:element name="polynomial" type="Polynomial"/>
      <xsd:element name="data" type="Points"/>
      <xsd:element name="residuals" type="Residuals"/>
    </xsd:sequence>
    <xsd:attribute name="info" type="xsd:string"/>
  </xsd:complexType>

  <xsd:complexType name="Bounds">
    <xsd:sequence>
      <xsd:element name="min" type="xsd:float"/>
      <xsd:element name="max" type="xsd:float"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Polynomial">
    <xsd:sequence>
      <xsd:element name="order" type="xsd:decimal"/>
      <xsd:element name="coeffs" type="Coefficients"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Coefficients">
    <xsd:sequence>
      <xsd:element name="coeff" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:float">
              <xsd:attribute name="degree" type="xsd:decimal"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="Points">
    <xsd:sequence>
      <xsd:element name="point" maxOccurs="unbounded">

```

```

    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="x" type="xsd:float"/>
        <xsd:element name="y" type="xsd:float"/>
        <xsd:element name="fitted" type="xsd:float"/>
        <xsd:element name="residual" type="xsd:float"/>
      </xsd:sequence>
      <xsd:attribute name="number" type="xsd:decimal"/>
    </xsd:complexType>
  </xsd:element>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Residuals">
  <xsd:sequence>
    <xsd:element name="sum-of-squares" type="xsd:float"/>
    <xsd:element name="rms" type="xsd:float"/>
    <xsd:element name="maxima">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="positive" type="Maxima"/>
          <xsd:element name="negative" type="Maxima"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Maxima">
  <xsd:sequence>
    <xsd:element name="residual">
      <xsd:complexType>
        <xsd:simpleContent>
          <xsd:extension base="xsd:float">
            <xsd:attribute name="percent" type="Percentage"/>
          </xsd:extension>
        </xsd:simpleContent>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="number" type="xsd:decimal"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:simpleType name="Percentage">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="\d+\.\d+%"/>
  </xsd:restriction>
</xsd:simpleType>

</xsd:schema>

```

C.6 Schema for results

```

<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation>
      Certificate table data, with labels and units.
      Robin Barker 2005
    </xsd:documentation>
  </xsd:annotation>

```

```

</xsd:annotation>
<xsd:element name="results">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="device">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="type" type="xsd:string"/>
            <xsd:element name="number" type="xsd:string"/>
            <xsd:element name="reference" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="labels">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="label" type="xsd:string" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="units">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="unit" type="xsd:string" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:element name="table">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="tr" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="td" type="xsd:float" maxOccurs="unbounded" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>

</xsd:schema>

```

C.7 DTD for raw measurement data

```

<!ELEMENT raw-data (points, data, bounds, order-data, output)>
  <!ELEMENT points (fitting, sum-table)>
    <!ELEMENT fitting (#PCDATA)>
    <!ELEMENT sum-table (#PCDATA)>
  <!ELEMENT data (point)*>
    <!ELEMENT point (x, y)>
      <!ELEMENT x (#PCDATA)>

```

```

        <!ELEMENT y      (#PCDATA)>
<!ELEMENT bounds (min, max)>
    <!ELEMENT min      (#PCDATA)>
    <!ELEMENT max      (#PCDATA)>
<!ELEMENT order-data (order-variables*, order-bounds)>
    <!ELEMENT order-variables (order, variables)>
        <!ELEMENT order (#PCDATA)>
        <!ELEMENT variables (var*)>
            <!ELEMENT var (#PCDATA)>
    <!ELEMENT order-bounds (min, max)>
<!ELEMENT output (coe, x, y, res, rms)>
    <!ELEMENT coe      (#PCDATA)>
    <!ELEMENT res      (#PCDATA)>
    <!ELEMENT rms      (#PCDATA)>

<!ATTLIST raw-data    info  CDATA    #IMPLIED>
<!ATTLIST fitting     list  CDATA    #IMPLIED>
<!ATTLIST sum-table   list  CDATA    #IMPLIED>
<!ATTLIST point       label ID      #REQUIRED>
<!ATTLIST x            name CDATA    #IMPLIED
           list  CDATA    #IMPLIED>
<!ATTLIST y            name CDATA    #IMPLIED
           list  CDATA    #IMPLIED>
<!ATTLIST order        points CDATA    #IMPLIED>
<!ATTLIST min          info  CDATA    #IMPLIED>
<!ATTLIST max          note  CDATA    #IMPLIED>
<!ATTLIST output       list  CDATA    #IMPLIED>
<!ATTLIST coe          list  CDATA    #IMPLIED>
<!ATTLIST res          list  CDATA    #IMPLIED>
<!ATTLIST rms          list  CDATA    #IMPLIED>

```

C.8 DTD for fitting data

```

<!ELEMENT fitting      (bounds, polynomial, data, residuals)>
<!ATTLIST fitting      info  CDATA    #IMPLIED>
<!ELEMENT bounds      (min, max)>
<!ELEMENT min          (#PCDATA)>
<!ELEMENT max          (#PCDATA)>
<!ELEMENT polynomial   (order, coeffs)>
<!ELEMENT order        (#PCDATA)>
<!ELEMENT coeffs       (coeff*)>
<!ELEMENT coeff        (#PCDATA)>
<!ATTLIST coeff        degree CDATA    #IMPLIED>
<!ELEMENT data         (point*)>
<!ELEMENT point        (x, y, fitted, residual)>
<!ATTLIST point        number CDATA    #IMPLIED>
<!ELEMENT x            (#PCDATA) >
<!ELEMENT y            (#PCDATA) >
<!ELEMENT fitted       (#PCDATA) >
<!ELEMENT residual     (#PCDATA) >
<!ELEMENT residuals    sum-of-squares, rms, maxima)>
<!ELEMENT sum-of-squares (#PCDATA)>
<!ELEMENT rms          (#PCDATA)>
<!ELEMENT maxima       positive, negative)>
<!ELEMENT positive     residual, number)>
<!ELEMENT negative     residual, number)>
<!ATTLIST residual     percent CDATA    #IMPLIED>
<!ELEMENT number       (#PCDATA)>

```

C.9 DTD for results

<!ELEMENT results	(device, labels, units, table)>
<!ELEMENT device	(type, number, reference)>
<!ELEMENT type	(#PCDATA)>
<!ELEMENT number	(#PCDATA)>
<!ELEMENT reference	(#PCDATA)>
<!ELEMENT labels	(label*)>
<!ELEMENT label	(#PCDATA)>
<!ELEMENT units	(unit*)>
<!ELEMENT unit	(#PCDATA)>
<!ELEMENT table	(tr*)>
<!ELEMENT tr	(td*)>
<!ELEMENT td	(#PCDATA)>

C.10 Sample calibration certificate

The following pages are taken from a calibration generated from the XML files in this section. The data is nonsense: these pages serve only to demonstrate how the data is used in the final layout of the certificate.

RHODIUM-IRON RESISTANCE THERMOMETER
SERIAL No. X123

FOR: A Customer Ltd
256 Octet Road
Kingston
Surrey
KT1 1XP

DESCRIPTION: Rhodium-iron resistance thermometer,
type 5187U, four lead, helium filled
capsule.

IDENTIFICATION: Serial No. X123

BASIS OF TEST: Calibration from 1590 K to 2210 K

DATE OF CALIBRATION: 1 April 2006

The reported uncertainty is based on a standard uncertainty multiplied by a coverage factor of $k = 2$, providing a level confidence of approximately 95%.

Reference: MSC/06/795

Page 1 of 5

Date of issue:

Signed:

(Authorised Signatory)

Checked by:

Name: Dr P R Scientist

for Managing Director

MEASUREMENTS

The resistance of the thermometer was measured at 31 temperatures in the range 1590.0K to 2210.0K by comparison with the standard 10Ω coil using a resistance bridge operating at 25Hz. No DC measurements were made, but previous experiments have shown that differences between measurements at 25Hz AC and DC are equivalent to less than 0.0002K in thermometers of this type.

The effective measuring currents were approximately 0.3mA above 1K, 0.15mA below 1K and 0.1mA below 0.75K, and the calibration of the thermometer increases as the temperature decreases. Its effect at the specified currents did not exceed 0.0004K.

RESULTS

The calibration is in terms of the International Temperature Scale of 1990, ITS-90. The scale is maintained on standard rhodium-iron resistance thermometers.

The calibration data were fitted using a least-squares procedure by Chebyshev series, with the independent variable (resistance) transformed to lie in the range -1 to $+1$.

A fourth order curve gave a standard error of approximately 0.0002K, and this order was selected for tabulation. Descriptions of the use of the Chebyshev series and the equivalent power series, together with the coefficients, are given with the tables. Values of resistance are given in ohms and temperature in kelvins.

UNCERTAINTIES

The uncertainty in the calibration with respect to the ITS-90 as maintained at NPL, and to the magnetic thermometry measurements, is estimated to be not greater than $\pm 0.001\text{K}$. This included the random and systematic uncertainties associated with the measurements and with the standard thermometers, at the level of 95% confidence.

Thermometer Number X123

ITS-90 Calibration 1590.0 K to 2210.0 K.

Lower and upper bounds of resistance are 25.4 Ω and 41.2 Ω .

The data were fitted without forcing. Equal weight was given to each data point.

The fitted polynomial is order 4.

	Resistance	Temperature	Fitted	Residual
1	2203.02	41.1834	41.1835	0.0001
2	2103.85	38.3148	38.3144	-0.0004
3	2004.47	35.5338	35.5340	0.0002
4	1901.79	32.7789	32.7795	0.0006
5	1803.05	30.2331	30.2320	-0.0011
6	1703.36	27.7714	27.7721	0.0007
7	1603.30	25.4252	25.4250	-0.0002

Sum of Squares of Residuals 0.000002

Maximum Negative Residual -0.001132 at point number 5

Maximum Positive Residual 0.000723 at point number 6

Thermometer Number X123

ITS-90 Calibration 1590.0 K to 2210.0 K.

Lower and upper bounds of resistance are $R_l = 25.4 \Omega$ and $R_u = 41.2 \Omega$.

The Chebyshev coefficients are obtained using values of R transformed to a new variable x by the equation

$$x = ((R - R_l) - (R_u - R)/(R_u - R_l)).$$

This x lies in the range -1 to $+1$. The coefficients a_0, a_1, \dots, a_n , are given below for the Chebyshev series:-

$$\text{Temperature, } T = 0.5a_0 + \sum_{j=1}^n a_j T_j, \text{ where } T_j \text{ is } \cos(jy) \text{ for } \cos(y) = x.$$

The coefficients, b_j , of the corresponding power series

$$T = \sum_{j=0}^n b_j x^j$$

are also given below, with the warning that they may be less accurate.

The polynomial order, n , is 4.

	Chebyshev coeffs	Power series coeffs
0	65.98616270	32.73237469
1	8.13962029	8.15888058
2	0.26181725	0.51474974
3	-0.00642010	-0.02568039
4	0.00111060	0.00888477

A Chebyshev series is evaluated for a given value of x by the following method:-

Set $c_{n+2} = c_{n+1} = 0$.

For $j = n, n-1, \dots, 1, 0$ calculate $c_j = 2xc_{j+1} - c_{j+2} + a_j$

The value of the series, i.e., the temperature, is $0.5(c_0 - c_2)$.

When using the coefficient, Chebyshev or Power Series, value of the independent variable R must be transformed to x . Note that no term in either series exceeds its coefficient, so that the coefficients may be rounded or the series truncated to suit the desired accuracy.

The polynomials are not valid outside the range of the fitted data.

Thermometer Number X123

Temp. K	Current A	Slope A/ K	Temp. K	Current A	Slope A/ K
1600	25.350	0.0228	2000	35.417	0.0274
1610	25.578	0.0230	2010	35.691	0.0275
1620	25.808	0.0231	2020	35.966	0.0276
1630	26.039	0.0232	2030	36.243	0.0277
1640	26.271	0.0234	2040	36.520	0.0278
1650	26.505	0.0235	2050	36.798	0.0279
1660	26.739	0.0236	2060	37.078	0.0280
1670	26.976	0.0237	2070	37.358	0.0281
1680	27.213	0.0239	2080	37.639	0.0282
1690	27.451	0.0240	2090	37.922	0.0283
1700	27.691	0.0241	2100	38.205	0.0284
1710	27.932	0.0242	2110	38.489	0.0285
1720	28.175	0.0243	2120	38.775	0.0286
1730	28.418	0.0245	2130	39.061	0.0287
1740	28.663	0.0246	2140	39.349	0.0288
1750	28.908	0.0247	2150	39.637	0.0289
1760	29.155	0.0248	2160	39.927	0.0290
1770	29.404	0.0249	2170	40.217	0.0291
1780	29.653	0.0250	2180	40.509	0.0293
1790	29.903	0.0252	2190	40.801	0.0294
1800	30.155	0.0253	2200	41.095	0.0000
1810	30.408	0.0254			
1820	30.662	0.0255			
1830	30.917	0.0256			
1840	31.173	0.0257			
1850	31.430	0.0258			
1860	31.688	0.0259			
1870	31.948	0.0260			
1880	32.208	0.0262			
1890	32.470	0.0263			
1900	32.732	0.0264			
1910	32.996	0.0265			
1920	33.261	0.0266			
1930	33.527	0.0267			
1940	33.794	0.0268			
1950	34.062	0.0269			
1960	34.331	0.0270			
1970	34.601	0.0271			
1980	34.872	0.0272			
1990	35.144	0.0273			