

IEEE Standard Design Criteria of Complex Virtual Instruments for Household Appliance Test

IEEE Instrumentation and Measurement Society

Developed by the
Sensor Technology Committee

IEEE Std 2735.1™-2022

STANDARDS

IEEE Standard Design Criteria of Complex Virtual Instruments for Household Appliance Test

Developed by the
Sensor Technology Committee
of the
IEEE Instrumentation and Measurement Society

Approved 8 November 2022

IEEE SA Standards Board

Abstract: This standard designs and standardizes data input interfaces, computing service interfaces, interconnection service interfaces, Laboratory Information Management System (LIMS) service interfaces, functional design requirements, and constraints for Complex Virtual Instruments (CVI) in the field of household appliance testing based on IEEE Std 2735. The purpose is to process the data or metadata from different types of household appliance complex virtual instruments.

Keywords: Complex Virtual Instruments, design specifications, household appliance testing, IEEE 2735™, IEEE 2735.1™

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2023 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 6 February 2023. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-1-5044-9180-8 STD25775
Print: ISBN 978-1-5044-9181-5 STDPD25775

IEEE prohibits discrimination, harassment, and bullying.

For more information, visit <https://www.ieee.org/about/corporate/governance/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (<https://standards.ieee.org/ipr/disclaimers.html>), appear in all standards and may be found under the heading “Important Notices and Disclaimers Concerning IEEE Standards Documents.”

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE standards documents are supplied “AS IS” and “WITH ALL FAULTS.”

Use of an IEEE standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group. Statements made by volunteers may not represent the formal position of their employer(s) or affiliation(s).

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents.**

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and subcommittees of the IEEE SA Board of Governors are not able to provide an instant response to comments, or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile and Interests area of the [IEEE SA myProject system](#).¹ An IEEE Account is needed to access the application.

Comments on standards should be submitted using the [Contact Us](#) form.²

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption

¹Available at: <https://development.standards.ieee.org/myproject-web/public/view.html#landing>

²Available at: <https://standards.ieee.org/content/ieee-standards/en/about/contact/index.html>.

by public authorities and private users, neither IEEE nor its licensors waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; <https://www.copyright.com/>. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit [IEEE Xplore](#) or [contact IEEE](#).³ For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the [IEEE SA Website](#).⁴ Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in [IEEE Xplore](#). Users are encouraged to periodically check for errata.

Patents

IEEE Standards are developed in compliance with the [IEEE SA Patent Policy](#).⁵

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at <https://standards.ieee.org/about/sasb/patcom/patents.html>. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

³Available at: <https://ieeexplore.ieee.org/browse/standards/collection/ieee>.

⁴Available at: <https://standards.ieee.org/standard/index.html>.

⁵Available at: <https://standards.ieee.org/about/sasb/patcom/materials.html>.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. IEEE Standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

Participants

At the time this standard was completed, the Complex Instrument Working Group had the following entity membership:

Zhongwen Guo, Chair
Hongyang Ma, Vice Chair
Tianao Zhang, Secretary

<i>Organization Represented</i>	<i>Name of Representative</i>
China University of Mining and Technology	Pengpeng Chen
Haier Group Corporation	Ming Li
Hisense Co., Ltd.	Shiyong Liu
Midea Group	Guoping Xu
Ocean University of China	Zhongwen Guo
Qingdao Lanbo Ocean Technology Co. Ltd.	Yueqiang Zhang
Qingdao Lanwan Information Technology Co., Ltd	Tianao Zhang
Qingdao Star Information Technology Co. Ltd.	Gaoyi Wang
QingDao University of Technology	Hongyang Ma
Tianjin University	Lei Zhang

The Working Group gratefully acknowledges the contributions of the following participants. Without their assistance and dedication, this standard would not have been completed.

Yuan Cao	Yujun Lan	Xi Wang
Ziyuan Cui	Gang Li	Xin Wang
Qingsong Dong	Gongxiang Li	Zhe Wang
Yuan Feng	Xiaomei Li	Zihao Wang
Xuesong Gao	Chao Liu	Chenyang Wei
Jian Gou	Yuan Liu	Hui Xia
Shuai Guo	Hanjiang Luo	Lintao Xian
Feng Hong	Bo Pang	Mengke Xin
Chunpeng Hu	Yanqi Pei	Junlong Xue
Keyong Hu	Chaofan Song	Shijie Yang
Qinghua Hu	Hao Su	Yongzhen Zhang
Ruobing Jiang	Wenyuan Tao	Hailei Zhao
Sining Jiang	Bin Wang	Shuo Zhao
Wenxiang Jiang	Jinxin Wang	Yingxin Zheng
Wei Jing	Weigang Wang	Ming Zhou
Jinglin Ju		Mingmao Zhou

The following members of the entity Standards Association balloting group voted on this guide. Balloters may have voted for approval, disapproval, or abstention.

0xSenses Corporation
China University of Mining and Technology
Haier Group Corporation
Hisense Group Holdings Co., Ltd.
Institute of Biomedical Engineering, Chinese
Academy of Medical Sciences & Peking Union
Medical College
Midea Group

Ocean University of China
Qingdao Lanbo Ocean Technology Co. Ltd.
Qingdao Lanwan Information Technology Co.,
Ltd
Qingdao Star Information Technology Co. Ltd.
QingDao University of Technology
Tianjin University

When the IEEE SA Standards Board approved this standard on 8 November 2022, it had the following membership:

David J. Law, *Chair*
Ted Burse, *Vice Chair*
Gary Hoffman, *Past Chair*
Konstantinos Karachalios, *Secretary*

Edward A. Addy
Ramy Ahmed Fathy
J. Travis Griffith
Guido R. Hiertz
Yousef Kimiagar
Joseph L. Koepfinger*
Thomas Koshy
John D. Kulick

Johnny Daozhuang Lin
Kevin Lu
Daleep C. Mohla
Andrew Myles
Damir Novosel
Annette D. Reilly
Robby Robson
Jon Walter Rosdahl

Mark Siira
Dorothy V. Stanley
Lei Wang
F. Keith Waters
Karl Weber
Sha Wei
Philip B. Winston
Daidi Zhong

*Member Emeritus

Introduction

This introduction is not part of IEEE Std 2735.1-2022, IEEE Standard for Design Criteria of Complex Virtual Instruments for Household Appliance Test.

Appliance testing is an important application field of IoT technology, and appliance testing systems are significant to improving appliance testing efficiency and reducing appliance testing costs which play an essential role in all aspects of appliance testing. However, due to the changing test requirements and the increasing requirements of enterprises for intelligent manufacturing, appliance testing systems need to be updated continuously, leading to code redundancy and repeated development. In addition, most household appliance testing systems lack standardized data service interfaces, which cannot realize the interconnection of test data. As a result, testers need to switch between different systems to get test data, which seriously restricts the efficiency of product development and production.

This standard is part of the IEEE 2735 standard family, which standardizes the design of Complex Virtual Instruments (CVIs) for household appliance testing. Based on IEEE Std 2735, this standard classifies and designs the public service interfaces requirements and constraints for different types of CVI in household appliance testing. The relevant interest users of this standard include household appliance enterprises, developers of household appliance testing software, system integrators, and supervision departments of household appliance production.

The content of this standard is organized as follows: [Clause 1](#) introduces the scope and purpose of the standard; [Clause 2](#) lists the indispensable referenced documents of this standard; [Clause 3](#) lists the definitions and abbreviations used in this document; [Clause 4](#) provides a general introduction of the framework; [Clause 5](#) to [Clause 8](#) detail the functional requirements, interface requirements, and constraints for the Refrigerator, Washing Machine, Water Heater and Air Conditioner Testing CVI Software, respectively.

Contents

1. Overview	18
1.1 Scope	18
1.2 Purpose	18
1.3 Word usage	18
2. Normative references	19
3. Definitions, acronyms, and abbreviations	19
3.1 Definitions	19
3.2 Acronyms and abbreviations	20
4. Framework introduction	20
5. Refrigerator Testing CVI Software	21
5.1 General	21
5.2 Acquisition parameters	21
5.3 Functional requirements	21
5.4 Interface requirements	26
5.5 Binding conditions	78
6. Washing Machine Testing CVI Software	79
6.1 General	79
6.2 Acquisition parameters	79
6.3 Functional requirements	79
6.4 Interface requirements	82
6.5 Binding conditions	82
7. Water Heater Testing CVI Software	82
7.1 General	82
7.2 Acquisition parameters	82
7.3 Functional requirements	83
7.4 Interface requirements	85
7.5 Binding conditions	118
8. Air Conditioner Testing CVI Software	118
8.1 General	118
8.2 Acquisition parameters	118
8.3 Functional requirements	119
8.4 Interface requirements	121
8.5 Binding conditions	130

List of Figures

Figure 1—Structure of BaseData element.....	28
Figure 2—Structure of sensorInfo element	29
Figure 3—Structure of groupInfo element	30
Figure 4—Structure of sensorData element	31
Figure 5—Structure of InterpolationData element	31
Figure 6—Structure of LoadPowerData element	32
Figure 7—Structure of HeaterInputData element.....	33
Figure 8—storageTemperature request parameters	34
Figure 9—Structure of ResultData element	35
Figure 10—Structure of CompartmentData1 element.....	36
Figure 11—Structure of CompartmentTempData1 element.....	37
Figure 12—Structure of CompartmentData2 element.....	38
Figure 13—Structure of CompartmentTempData2 element.....	39
Figure 14—powerConsumption request parameters	41
Figure 15—Structure of ResultData element	42
Figure 16—Structure of SS1Data element	43
Figure 17—Structure of IntergalAvgTemp element	44
Figure 18—Structure of TempChangeData element	45
Figure 19—Structure of SS2Data element	46
Figure 20—Structure of DF1Data element.....	48
Figure 21—Structure of DF2Data element.....	50
Figure 22—Structure of TempDeviation element	52
Figure 23—Structure of TempChange element	52
Figure 24—Structure of IntergalAvgTemp2 element	53
Figure 25—linearInterpolation request parameters.....	54
Figure 26—Structure of ResultData element	55
Figure 27—Structure of InterpolationResult element	55
Figure 28—loadPowerConsumption request parameters.....	56
Figure 29—Structure of ResultData element	57

Figure 30—heaterPowerConsumption request parameters	58
Figure 31—annualPowerConsumption request parameters	59
Figure 32—Structure of CVINode element.....	60
Figure 33—Structure of metaData element.....	61
Figure 34—Structure of contentSet element	62
Figure 35—Structure of content element	62
Figure 36—structure of selectedSensor element	63
Figure 37—Structure of selectedGroup element	64
Figure 38—Structure of the sensorData element.....	64
Figure 39—CVIInformation request parameters.....	65
Figure 40—metaData request parameters	66
Figure 41—Structure of the metaDataCollection element.....	66
Figure 42—selectSensor request parameters.....	67
Figure 43—Structure of SelectedSensorCollection element	68
Figure 44—selectGroup request parameters	69
Figure 45—selectedGroupCollection element structure	69
Figure 46—sensorData request parameters.....	70
Figure 47—Structure of the sensorDataCollection element	71
Figure 48—dataFile request parameters.....	72
Figure 49—Structure of Order element.....	73
Figure 50—orderInfo request parameters	74
Figure 51—Structure of OrderCollection element	74
Figure 52—startTaskTime request parameters	75
Figure 53—finishTaskTime request parameters.....	76
Figure 54—reportInfo request parameters	77
Figure 55—Structure of BaseData element.....	86
Figure 56—Structure of sensorInfo element	87
Figure 57—Structure of groupInfo element	87
Figure 58—Structure of sensorData element	88
Figure 59—ESWHstandingLoss24H request parameters	89
Figure 60—Structure of ResultData element	90

Figure 61—ESWHhotWaterOutputRate request parameters	93
Figure 62—Structure of ResultData element	94
Figure 63—ESWHheatingEfficiency request parameters	96
Figure 64—Structure of ResultData element	97
Figure 65—Structure of KeyPointTemp element	97
Figure 66—IWHheatingEfficiency request parameters	99
Figure 67—Structure of ResultData element	99
Figure 68—IWHheatingSpeed request parameters	101
Figure 69—Structure of ResultData element	101
Figure 70—IWHperformanceForPressure request parameters	102
Figure 71—Structure of ResultData element	103
Figure 72—IWHperformanceForVoltage request parameter	104
Figure 73—Structure of ResultData element	105
Figure 74—IWHperformanceForTemp request parameters.....	107
Figure 75—Structure of ResultData element	108
Figure 76—HPWHhotWaterOutputRate request parameters.....	109
Figure 77—Structure of ResultData element	110
Figure 78—HPWHheatingCapacity request parameters.....	112
Figure 79—Structure of ResultData element	112
Figure 80—HPWHstaticHeatingCapacity request parameters.....	114
Figure 81—Structure of ResultData element	115
Figure 82—Structure of BaseData element.....	122
Figure 83—Structure of nozzleInfo element	123
Figure 84—coolingCapacity request parameters	123
Figure 85—Structure of ResultData element	124
Figure 86—heatingCapacity request parameters.....	126
Figure 87—Structure of ResultData element	127

List of Tables

Table 1—Types of sensor parameters	21
Table 2—Description of BaseData element.....	28
Table 3—Description of sensorInfo element	29
Table 4—Description of groupInfo element.....	30
Table 5—Description of sensorData element	31
Table 6—Description of InterpolationData element.....	31
Table 7—Description of LoadPowerInput element	32
Table 8—Description of HeaterInputData element	33
Table 9—Description of storageTemperature request parameters	34
Table 10—Description of ResultData element	36
Table 11—Description of CompartmentData1 element	36
Table 12—Description of CompartmentTempData1 element	37
Table 13—Description of CompartmentData2 element	38
Table 14—Description of CompartmentTempData2 element	39
Table 15—Description of powerConsumption request parameters	41
Table 16—Description of ResultData element	42
Table 17—Description of SS1Data element.....	44
Table 18—Description of IntergalAvgTemp element.....	45
Table 19—TempChangeData element description	45
Table 20—Description of SS2Data element.....	47
Table 21—Description of DF1Data element	49
Table 22—Description of DF2Data element	51
Table 23—Description of TempDeviation element	52
Table 24—TempChange element description.....	52
Table 25—Description of IntergalAvgTemp2 element.....	53
Table 26—Description of linearInterpolation parameters	54
Table 27—Description of ResultData element	55
Table 28—InterpolationResult element description.....	55
Table 29—Description of loadPowerConsumption request parameters	56

Table 30—Description of ResultData element	57
Table 31—Description of heaterPowerConsumption request parameters	58
Table 32—Description of annualPowerConsumption request parameters	59
Table 33—Description of CVINode element	61
Table 34—Description of metaData element.....	62
Table 35—Description of contentSet element	62
Table 36—Description of content element	62
Table 37—description of selectedSensor element.....	63
Table 38—description of selectedGroup element.....	64
Table 39—Description of sensorData element	65
Table 40—Description of CVIInformation request parameters.....	65
Table 41—Description of metaData interface parameters.....	66
Table 42—Description of metaDataCollection element.....	66
Table 43—selectSensor interface parameter description.....	67
Table 44—Description of selectedSensorCollection element	68
Table 45—Description of selectGroup request parameters	69
Table 46—selectedGroupCollection element description	69
Table 47—Description of SensorData request parameters	70
Table 48—Description of metaDataCollection element.....	71
Table 49—Description of dataFile request parameters.....	72
Table 50—Description of Order element.....	73
Table 51—Description of orderInfo interface parameter description	74
Table 52—Description of OrderCollection element.....	74
Table 53—Description of startTaskTime request parameter	75
Table 54—Description of finishTaskTime interface parameter	76
Table 55—Description of ReportInfo interface parameter	77
Table 56—Exception element parameters.....	77
Table 57—Standard exception codes and meanings.....	78
Table 58—Types of sensor parameters	79
Table 59—Types of sensor parameters	83
Table 60—Description of BaseData elements	86

Table 61—Description of sensorInfo element	87
Table 62—Description of groupInfo element.....	88
Table 63—Description of sensorData element	88
Table 64—Description of ESWHstandingLoss24H request parameters	89
Table 65—Description of ResultData element	91
Table 66—Description of hotWaterOutputRate request parameters.....	93
Table 67—Description of ResultData element	95
Table 68—Description of ESWHheatingEfficiency request parameters	96
Table 69—Description of ResultData element	97
Table 70—Description of KeyPointTemp element.....	98
Table 71—Description of IWHheatingEfficiency request parameters	99
Table 72—Description of ResultData element	100
Table 73—IWHheatingSpeed interface parameter description	101
Table 74—Description of ResultData element	101
Table 75—IWHperformanceForPressure interface parameter description	102
Table 76—Description of ResultData element	103
Table 77—Description of IWHperformanceForVoltage request parameters.....	105
Table 78—Description of ResultData element	106
Table 79—Description of IWHperformanceForTemp request parameters.....	107
Table 80—Description of ResultData element	108
Table 81—Description of HPWHhotWaterOutputRate request parameters.....	110
Table 82—Description of ResultData element	111
Table 83—Description of HPWHheatingCapacity request parameters.....	112
Table 84—Description of ResultData element	113
Table 85—Description of HPWHstaticHeatingCapacity request parameters.....	114
Table 86—Description of ResultData element	115
Table 87—Exception parameter description	116
Table 88—Standard exception codes and meanings.....	117
Table 89—Types of sensor parameters	118
Table 90—Description of BaseData element.....	123
Table 91—Description of nozzlesInfo element	123

Table 92—Description of coolingCapacity request parameters 124

Table 93—Description of ResultData structure 125

Table 94—Description of heatingCapacity request parameters..... 126

Table 95—Description of ResultData element 128

Table 96—Exception parameter description 129

Table 97—Standard exception codes and their meanings..... 129

IEEE Standard Design Criteria of Complex Virtual Instruments for Household Appliance Test

1. Overview

1.1 Scope

This standard defines the framework of building household appliance test software system based on complex virtual instruments (CVIs) which are used for processing and displaying the sensor data from intelligent sensors and the related metadata. This framework provides the guidelines for CVI-based development process, in which CVI structure design covers management of sensor data and metadata, virtual instrument engine based on description of appliance testing ambient and test information, data formats of test data, and Representational State Transfer (REST) service interfaces for CVI interactions; CVI mapping schemes describe the correspondence from test units to CVIs; CVI relations define the relationships between CVIs and describe the methods of extending and compositing multiple CVIs.

1.2 Purpose

The purpose of this standard is to advise and assist developers to adopt CVI-based development method in building household appliance test software system. Implementation of this standard simplifies software development process greatly on the basis of ensuring the quality of software and then resulting in high development efficiency. Implementation of this standard also achieves the standardization of household appliance test data, solves the interoperability and integration problems of test applications for household appliances, helps users share test data, and lays the data foundation for big data analysis.

1.3 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).^{6,7}

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals *is recommended that*).

⁶The use of the word *must* is deprecated and cannot be used when stating mandatory requirements; *must* is used only to describe unavoidable situations.

⁷The use of *will* is deprecated and cannot be used when stating mandatory requirements; *will* is only used in statements of fact.

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may equals is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can equals is able to*).

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE Std 2735-2022, IEEE Standard for Interoperability of Complex Virtual Instruments for Internet of Things.^{8,9}

IEEE Std 1851™-2012, IEEE Standard for Design Criteria of Integrated Sensor-Based Test Applications for Household Appliances.

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.¹⁰

household appliance testing: Verification of the different performances that an appliance shall have in normal conditions.

Household Appliance Testing CVI Software: A software system for collecting, managing, analyzing, and reporting data of household appliance testing.

laboratory: A room or building where the household appliance testing is performed.

physical sensors: The actual sensors can directly measure the data from the household appliance.

private sensors: The sensors assigned to a particular testbed are used independently by that testbed.

public sensors: The sensors that can be used by multiple testbeds in common.

sensor group: A logical entity, classified according to the function or deployment location of the sensors, is used to calculate the maximum, minimum, and average values in a collection.

start testing: Start a test task to monitor the household appliances.

stop testing: End a test task to stop monitoring the household appliances.

testbed: A bench for conducting appliance testing, which contains instruments and a series of sensors connected to the monitored household appliance. A laboratory has at least one testbed.

⁸IEEE publications are available from the Institute of Electrical and Electronics Engineers, Inc., (<https://standards.ieee.org/>).

⁹The IEEE standards or products referred to in this clause are trademarks of the Institute of Electrical and Electronics Engineers, Inc.

¹⁰*IEEE Standards Dictionary Online* is available at: <http://dictionary.ieee.org>. An IEEE Account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

test items: The testing content required to be performed in the testing standard for household appliances.

test metadata: The information to describe the household appliance testing. This includes the test task number, product number, test item, test purpose, etc.

test tasks: Complex tasks performed by household appliance manufacturers or commissioned companies on appliance products.

virtual sensors: Unlike physical sensors, the data obtained by the logical operation of physical sensors data.

3.2 Acronyms and abbreviations

CVI	Complex Virtual Instrument
HTTP	Hyper Text Transport Protocol
JSON	JavaScript Object Notation
LIMS	Laboratory Information Management System
REST	Representational State Transfer

4. Framework introduction

This standard proposes a CVI design specification for the household appliance testing, follows the general CVI architecture of IEEE Std 2735, and specifies the specific functions, public service interfaces, and constraints of the household appliance testing CVI software.¹¹

- This document standardizes the function set of CVI software for different application scenarios combined with the needs of appliance testing, including test metadata management, sensor management, monitoring control, navigation management, curve display, list display, calculation and analysis, data import, data export, comprehensive query, curve printing, report generation, data management, Laboratory Information Management System (LIMS), and logging.
- This document specifies the public service interfaces for the data interoperability, including data input interfaces, computing service interfaces, interconnection service interfaces, and LIMS service interfaces. Computing service interfaces, interconnection service interfaces, and LIMS service interfaces are implemented using microservice, and the interconnection service interfaces shall be designed according to IEEE Std 1851™-2012. The household appliance testing CVI software subscribes to the public service interfaces as a service consumer.

The public service interfaces that are described include the following:

- a) The appliance testing CVI software receives sensor data collected by the instrument interaction software through the data input interfaces.
- b) The appliance testing CVI software could analyze and calculate the sensor parameters collected from different types of household appliances testing through the computing service interfaces. The interfaces include refrigerator loading power consumption, the water heater heating efficiency, cooling capacity or heating capacity of the air conditioner, etc.
- c) The appliance testing CVI software provides test metadata and associated test data through the interconnection service interfaces. It includes the interface to obtain basic information on the CVI

¹¹Information on references can be found in [Clause 2](#).

software node, household appliance test metadata, sensor metadata, sensor group data, and sensor data.

- d) The household appliance testing CVI software achieves information interoperability with LIMS through the LIMS service interfaces, including the interfaces to obtain test business information from LIMS or upload start time, stop time, and report data to LIMS.

This document specifies the constraints for appliance testing CVI software.

5. Refrigerator Testing CVI Software

5.1 General

Refrigerator Testing CVI Software is to acquire, manage, display, and analyze test metadata and associated sensor data in refrigerator testing. This clause describes the acquisition parameters, functional requirements, interface requirements, and constraints that the Refrigerator Testing CVI Software shall have.

5.2 Acquisition parameters

The Refrigerator Testing Software can collect the types of sensor parameters shown in Table 1, where the user can define the number of parameters of each type.

Table 1—Types of sensor parameters

Number	Name
1	voltage
2	current
3	power
4	power consumption
5	power factor
6	frequency
7	temperature
8	ambient temperature
9	ambient relative humidity

5.3 Functional requirements

The Refrigerator Testing CVI Software provides the functions of test metadata management, sensor management, monitoring control, navigation management, curve display, data list display, calculation and analysis, data import and export, comprehensive query, curve printing, data management, test report generation, LIMS, and log record, as well as the graphical user interaction interface.

5.3.1 Test metadata management

The refrigerator test metadata contains multiple input items to describe refrigerator testing. It includes the test task number, LIMS business number, product number, product type, compressor model, test item, test person, commissioning company, test standard, test purpose, climate type, remarks, test start time, test stop time, etc. The refrigerator test metadata shall consist of the following:

- a) **test task number:** A unique number used to identify the test task.
- b) **LIMS business number:** The number used to associate LIMS business information.

- c) **product number:** The unique code used to identify the refrigerator product.
- d) **product type:** It is used to identify a type of refrigerator product, and different types correspond to different models.
- e) **compressor model:** The compressor is the core component of the refrigerator, and the compressor model is used to identify a type of compressor.
- f) **test items:** There are various items of refrigerator testing, such as power consumption testing, freezing capacity testing, etc.
- g) **test person:** A tester conducts the refrigerator testing.
- h) **commissioned company:** The company is authorized to conduct refrigerator testing, including the development department, quality control department, etc.
- i) **test standard:** The testing specification requirements to which the refrigerator testing shall comply.
- j) **test purpose:** The purpose of refrigerator testing includes new product certification testing, module change testing, and annual review testing.
- k) **climate type:** The type of climate the refrigerator is testing, such as tropical, subtropical, temperate, or sub-temperate.
- l) **notes:** Describe some other considerations.
- m) **start time:** The time to start a refrigerator testing.
- n) **stop time:** The time to stop a refrigerator testing.

The method of the refrigerator test metadata generation as follows:

- The user can manually input the metadata information for a single testbed.
- The user can input the metadata information for multiple testbeds in batches.
- CVI software automatically inputs the metadata information relevant testbed by interconnecting with the LIMS.

5.3.2 Sensor management

The refrigerator testing contains multiple sensors listed in [Table 1](#), in which voltage, current, power, frequency, power factor, power consumption, and temperature are private sensors for each testbed. Ambient temperature is a common sensor for all testbeds as well as ambient relative humidity. The sensor management functions are as follows:

- a) Each sensor can be assigned to a corresponding testbed, and each testbed contains at least one voltage, current, power, frequency, power factor, power consumption, ambient temperature, ambient relative humidity sensor, and multiple temperature sensors.
- b) Each sensor has a default name, and the user can set an alias of the sensor as needed.
- c) Users can choose to use any sensor in the testbed, and they can borrow private sensors from other testbeds that are free when the number of sensors in the testbed is not enough, and each private sensor can only be used by one testbed independently at the same time.
- d) Users can add or delete sensor groups in each testbed and define the group name, and they can also add the sensors into the same group to form a sensor subset. It can support logical operations on the instantaneous values of the sensors in the subset to calculate the average, maximum, minimum, and integral average values to form a virtual sensor.

5.3.3 Monitoring and control

The Refrigerator Testing CVI Software can obtain the connection equipment of the current state and control the equipment. This function is configurable and can support the user in defining the function menu based on the requirements. It can control the testbed power, heater, humidifier, circulation fan, digital temperature or humidity controller, digital power meter, and frequency conversion power. The details are described as follows:

- a) It can control the power switch for each refrigerator testbed.
- b) It can control the switch of heaters, humidifiers, and circulation fans in the refrigerator laboratory.
- c) It can set the ambient temperature targets of the digital temperature controller or set the humidity targets of the digital humidity controller.
- d) It can set the voltage and frequency targets of the Variable Frequency Power supply.
- e) It can turn on or turn off the digital power meter and reset the accumulated power consumption.

It has the function of starting or stopping tests for refrigerator products, including the control of testing for single testbed or multiple testbeds in batches.

It can set the data acquisition cycle of the Refrigerator Testing CVI Software according to the requirements of refrigerator testing standards and be able to collect complete sensor data within the set acquisition cycle.

5.3.4 Navigation management

The Refrigerator Testing CVI Software can connect multiple laboratories of the same type, where each laboratory contains one or more testbeds. Navigation management can navigate to all the laboratory and testbeds to indicate the subordination of the laboratory and testbeds. After selecting a test laboratory, it supports displaying the testbeds to which the laboratory is subordinate, and it can check the usage status of all the testbeds. It can show the refrigerator test data or curves of the corresponding testbeds after switching between each other which is starting.

5.3.5 Curve display

The curve window is composed of horizontal and vertical axes. The horizontal axis is time, and the vertical axis is each type of sensor value, including voltage, current, power, frequency, power factor, power consumption, temperature, humidity, etc. Each sensor data can be drawn as a curve in the curve window and supports the real-time drawing of the sensor data curve from the selected testbed in the navigation management bar.

The curve window has the functions of curve scaling, curve self-adaptation, coordinate setting, curve movement, curve show and hide, curve annotation, instantaneous data display, data comparison, range selection, latest data display, and curve color setting. The details are described as follows:

- a) **curve scaling:** This function includes curve magnification and curve shrinkage. Curve magnification has the horizontal magnification and local magnification. Horizontal magnification enables users to select a curve section horizontally and fill the curve window horizontally; local magnification allows users to choose a rectangular area in the curve window. The rectangular area fills the whole curve window. There are two ways to shrink a curve, one is to the state before the last zoom in, and the other is to the initial state.
- b) **curve adaption:** When the sensor curve exceeds the entire curve window, this function can automatically adjust the display range of the horizontal and vertical coordinates so that the sensor curve can be fully adapted and displayed in the curve window.

- c) **coordinate setting:** The users can choose to show or hide any vertical coordinates, set the range of each coordinate of the curve window, including horizontal and vertical, and refresh the sensor curve to fit the new page layout after the coordinates setting is completed.
- d) **curve moving:** Due to the curve window size limitation, all curves cannot be displayed absolutely after the curve is horizontally enlarged. Users can drag the mouse to move the curve and move the sensor curve to the curve window that needs to be focused.
- e) **curve show and hide:** The users can choose to show or hide any sensor data curve in the curve window.
- f) **curve annotation:** The users can annotate specific points in the sensor curve, such as adding text descriptions, values of specific points, the acquisition time difference between different sensors, etc., while supporting the clearing of these annotations.
- g) **instantaneous data displaying:** The mouse pointer is displayed in the curve window in the form of cross coordinates during user operation, and the instantaneous data of each sensor at the moment corresponding to the cross coordinate position can be displayed.
- h) **data comparison:** Select the sensor data to compare and analyze at different moments.
- i) **range selection:** The users can mark a range of curves in the curve window.
- j) **latest data display:** The curve window dynamically updates the latest sensor data curve collected in real-time.
- k) **color setting:** The user can define the color of each sensor curve.

5.3.6 Data list show

This function is capable of displaying the sensor data associated with the navigation management (see 5.3.4) in the form of a list, including the real-time display and historical display.

- a) According to the acquisition cycle, the real-time display can add the latest sensor data to the list. The displayed data list shall contain the sensor name, sensor unit, sensor value, and the acquisition time stamp of sensor data.
- b) The history display can show sensor data for a specified time range or curve range (see 5.3.5); the displayed data list shall contain the sensor name, sensor value unit, sensor value, and the acquisition time stamp of sensor data.
- c) Both real-time and historical shows can support a user-defined selection of sensor parameters displayed in the data list. If there are too many parameters to show, the remaining parameters can be shown by adding scroll bars.

5.3.7 Computational analysis

It can conduct fundamental statistical analysis of sensor data collected in refrigerator performance testing or reliable testing, including calculating maximum, minimum, and arithmetic mean integral values in a selected range curve (see 5.3.5) or specified time range.

It can calculate the on-time, off-time, and on-stop time ratio of each on-off cycle; it can calculate the total on-time, off-time, on-time, off-time, and the on-off ratio of the test data in the user-selected curve range or the specified time range. If no curve range is selected, all the periods will be calculated by default.

Referring to the test standard specification requirements, it supports the analysis of sensor data collected from different refrigerator type test items, including storage temperature testing, steady-state power or temperature determination testing, power consumption testing, etc. The specific calculation service interfaces are shown in 5.4.3.

5.3.8 Data export

The users can export refrigerator test metadata and sensor data to a data file based on a user-specified time range. The exported file format supports excel, text, or other types.

5.3.9 Data import

The users can import data files storing metadata and sensor data from single testing of a refrigerator. Any Refrigerator Testing CVI Software can parse the information contained in the imported data files to recover the data lists or curves.

5.3.10 Comprehensive query

The users can query, select, and display records according to test time, test metadata, or testbed information for refrigerator testing. After finding the data that meet the users' requirements, select one of the test tasks to display the curve, the data list. Selecting the historical monitoring data of the two test tasks can visually analyze and compare the data of the two tests.

5.3.11 Curve printing

The users are supported to set up curve report templates and print curve reports. The templates include the title, curve, statistics, and legend areas and which size of each area can be set.

- a) The title area displays the report title and monitoring object description information, and the users can define the monitoring object input items displayed in the report.
- b) The curves area displays a curve window including data curves and marker information.
- c) The statistics area displays statistical values for critical sensor data, including maximum, minimum, average, etc.
- d) The legend area is colored to identify the different sensor data curves in the curve window.

5.3.12 Data management

This function can read, write, and back up data for refrigerator testing, provide data scheduling management for the above functions, and isolate other parts from database operations.

This function can manage CVI software information, instrument interaction software information, database information, and information related to monitoring objects. Monitoring objects include test metadata, navigation data, sensor data, sensor grouping data, etc.

This function provides two-level storage of data, including memory and a database. It can support data recovery in memory after restarting CVI software in case of sudden power failure or crash of the computer.

5.3.13 Report generation

The Report generation is able to configure a report template based on an Excel file, which refers to associating Refrigerator Testing CVI Software data with Excel report file cell mapping. Refrigerator Testing CVI Software data includes test metadata, sensor data, manual input data, and test result data.

- a) It can map refrigerator testing description information items with Excel report file cells, which can support the configuration of test metadata according to user requirements.
- b) It can associate sensor data for refrigerator testing with Excel report file cells and support the configuration of sensor data according to user requirements.

- c) Manual input data is repeated input data in refrigerator reports or data filled in after manual measurements, such as refrigerator chamber number and volume.
- d) The test result data is the final test result of refrigerator products according to the test standards, such as the actual power consumption and energy efficiency grade.

After the configuration is complete, the Refrigerator Testing CVI Software calls the report template file and automatically generates the test report.

5.3.14 LIMS

The LIMS is a system composed of computers and applications that can complete the information management of laboratory data, it shall consist of the following:

- a) The Refrigerator Testing CVI Software can receive test tasks and test metadata information dispatched by the LIMS and automatically input them into the test entry items of the corresponding testbeds, which are detailed in [5.3.1](#).
- b) When the refrigerator testing starts, the Refrigerator Testing CVI Software needs to submit starting time to the LIMS to mark that the refrigerator test has begun.
- c) When the refrigerator testing starts, the Refrigerator Testing CVI Software needs to submit stopping time to the LIMS to mark that the refrigerator test has ended. At the same time, it can submit the test report generated to the LIMS.
- d) The refrigerator testing report contains the test metadata, sensor data, and test result data. Users can define and edit the test report. The submitted test report types can be text, Excel, Word, picture, etc.

5.3.15 Logging

It can record the key events or execution processes during the Refrigerator Testing CVI Software operation. Once the software is a failure, anomaly, or even crashes, it can provide clues and guidance for the software developers to track back and locate problems during troubleshooting. The software logs include information, warning, and exception according to the importance level, each explained as follows:

- a) Information log records the status of the CVI software during regular operation, such as the initial loading time of the CVI software.
- b) Warning log records warning messages that occur during regular operation of the CVI software, such as errors in the CVI software data export format.
- c) Exception log records exception information during the operation of the CVI software, such as failure of the CVI software to read the database.

5.4 Interface requirements

This subclause describes the interface protocol and public service interfaces for the Refrigerator Testing CVI Software.

5.4.1 Interface protocols

The data input interfaces adopt socket and User Datagram Protocol (UDP) to provide a data source for Refrigerator Testing CVI Software; the computing service interfaces, interconnection service interfaces, and LIMS service interfaces are deployed in a microservice way for invoking remotely by Refrigerator Testing CVI Software. The communication parties use the Hyper Text Transport Protocol (HTTP) to transfer data in JavaScript Object Notation (JSON) format.

5.4.2 Data input interface

Refrigerator Testing CVI Software adopts the active acquisition data access method to control the instrument interaction software through normalized interaction commands to collect smart sensor data and achieve interoperability between smart sensors and instrument interaction software.

- a) The interactive commands that shall be available in the CVI software for refrigerator testing are as follows:
 - 1) **start command**: The Refrigerator Testing CVI Software sends a start command to the instrument interaction software, initializes the instruments and sensor parameters information, and responds to the CVI Software.
 - 2) **stop command**: The Refrigerator Testing CVI Software sends a stop command to the instrument interaction software to stop receiving sensor data, destroy the thread, and release the memory about data acquisition.
 - 3) **data acquisition command**: The Refrigerator Testing CVI Software sends a data acquisition command to the instrument interaction software to receive sensor data according to a specific communication protocol and respond to the CVI Software.
 - 4) **instrument control command**: The Refrigerator Testing CVI Software sends an instrument control command to the instrument interaction software to change the instrument state at a specific moment (see 5.3.3), such as controlling the digital power meter to turn on, stop, or reset remotely.
- b) To reduce socket communication from being maliciously attacked, which could cause data errors or communication interruption, the following security mechanisms are required:
 - 1) **authentication**: The data communication between the Refrigerator Testing CVI Software and the instrument interaction software can only be performed after verifying the identity information.
 - 2) **communication encryption**: The communication content between the Refrigerator Testing CVI Software and the equipment interaction software shall have data encryption ability. Others cannot be informed of the communication content except the communication parties.

5.4.3 Computing service interface

By invoking the computing service interfaces provided by the remote service providers, the Refrigerator Testing CVI Software realizes the calculation and analysis of the refrigerator collection parameters, including the calculation of the refrigerator storage temperature and the calculation of the loading power consumption.

5.4.3.1 Data structure

This subclause describes the Refrigerator Testing CVI Software data for analysis and calculation. This standard abstracts and proposes a common data model that covers the basic information required for refrigerator testing and can meet the requirements of calculation services for various types of refrigerators data.

5.4.3.1.1 BaseData

BaseData element contains all the information involved in the calculation process of refrigerator storage temperature, steady-state power consumption, and power consumption during defrosting/recovery periods, which is composed of sensor metadata, sensor group data, sensor data, test testbed, start time, etc. Its element structure is illustrated in Figure 1.

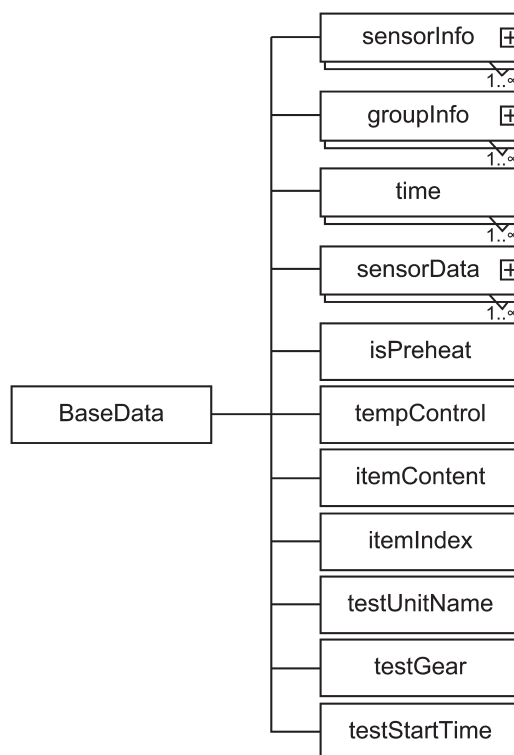


Figure 1—Structure of BaseData element

The description of the BaseData subelements is shown in [Table 2](#).

Table 2—Description of BaseData element

Name	Type	Description	Use
sensorInfo	SensorInfo	See 5.4.3.1.1.1	Required
groupInfo	GroupInfo	See 5.4.3.1.1.2	Required
time	array	Time difference sequence from the starting time.	Required
sensorData	SensorData	See 5.4.3.1.1.3	Required
isPreheat	boolean	Whether to preheat; 1 represents yes and 0 represents no.	Required
tempControl	double	The setting of temperature controller.	Required
itemContent	string	Input contents corresponding to input items. Each input content shall be separated by “@”.	Required
itemIndex	integer	The index of interpolated items.	Required
testUnitName	string	The name of testbed.	Required
testGear	integer	The gears of refrigerator testing.	Required
testStartTime	string	The starting time of refrigerator testing.	Required

5.4.3.1.1.1 SensorInfo

SensorInfo element contains the metadata of the sensor, which is composed of the sensor number, sensor name, sensor type, sampling value unit, and group number. Its element structure is illustrated in [Figure 2](#).

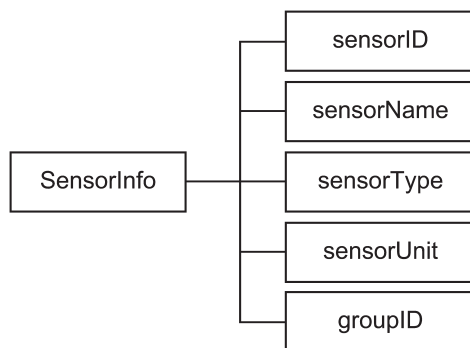


Figure 2—Structure of sensorInfo element

The description of sensorInfo element is shown in [Table 3](#).

Table 3—Description of sensorInfo element

Name	Type	Description	Use
sensorID	integer	The unique identifier of the sensor in a test unit.	Required
sensorName	string	The name of the sensor.	Required
sensorType	string	The type of the sensor.	Required
sensorUnit	string	The unit of the sensor sampling value.	Required
groupID	integer	The number of the group to which the sensor belongs.	Required

5.4.3.1.1.2 GroupInfo

GroupInfo element contains the basic attributes information of the group to which the sensor belongs, which is composed of the sensor group number, group type, group name, the volume of the compartment, etc. Its element structure is illustrated in [Figure 3](#).

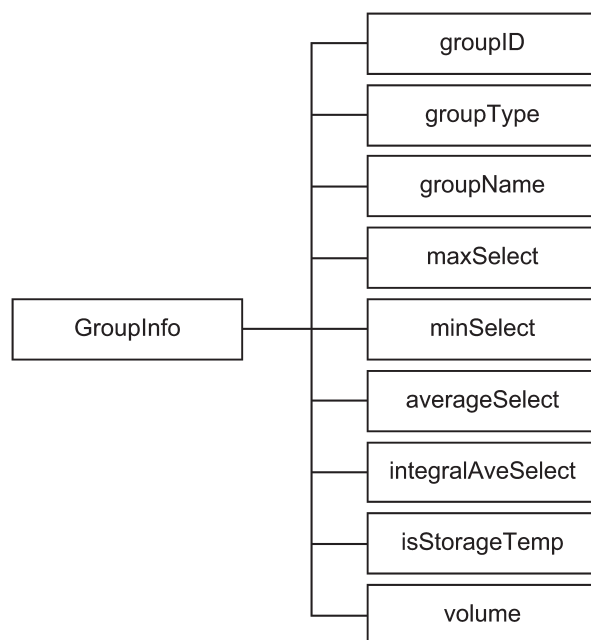


Figure 3—Structure of groupInfo element

The description of the groupInfo element is shown in [Table 4](#).

Table 4—Description of groupInfo element

Name	Type	Description	Use
groupID	integer	The unique identifier of the group.	Required
groupType	string	The type of the group.	Required
groupName	string	The name of the group.	Required
maxSelect	boolean	Whether to calculate the maximum value of the group; 1 represents yes and 0 represents no.	Required
minSelect	boolean	Whether to calculate the minimum value of the group; 1 represents yes and 0 represents no.	Required
averageSelect	boolean	Whether to calculate the average value of the group; 1 represents yes and 0 represents no.	Required
integralAveSelect	boolean	Whether to calculate the integral value of the group; 1 represents yes and 0 represents no.	Required
isStorageTemp	boolean	Whether to make storage temperature judgment; 1 represents yes and 0 represents no.	Required
volume	double	The volume of the refrigerator chamber, its unit is “L.”	Required

5.4.3.1.1.3 sensorData

SensorData element contains information about the data collected by the sensor, which is composed of sensor number and data sequences. Its element structure is illustrated in [Figure 4](#).

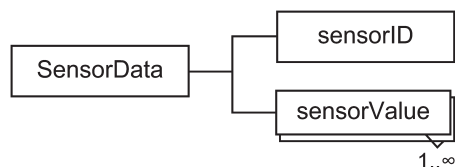


Figure 4—Structure of sensorData element

The description of sensorData element is shown in Table 5.

Table 5—Description of sensorData element

Name	Type	Description	Use
sensorID	integer	The unique identifier of the sensor in a test unit.	Required
sensorValue	array	All data collected by this sensor.	Required

5.4.3.1.2 InterpolationData

The InterpolationData element contains all the information involved in the calculation process of linear interpolation power consumption, composed of the temperature of two test points, compartment type, and other sub-elements. Its element structure is illustrated in Figure 5.

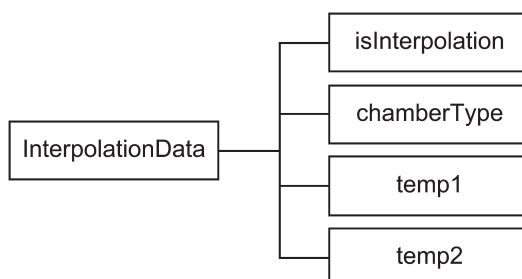


Figure 5—Structure of InterpolationData element

The description of InterpolationData element is shown in Table 6.

Table 6—Description of InterpolationData element

Name	Type	Description	Use
isInterpolation	boolean	Whether to use for interpolation; 1 represents yes and 0 represents no.	Required
chamberType	string	The type of chamber, such as freezer, refrigerator, etc.	Required
temp1	double	The temperature of test point 1.	Required
temp2	double	The temperature of test point 2.	Required

5.4.3.1.3 LoadPowerData

The LoadPowerData element contains all the information involved in calculating the load power consumption, which is composed of temperature, water quality in the storage chamber, start time, or end time of the loading process, etc. Its element structure is illustrated in Figure 6.

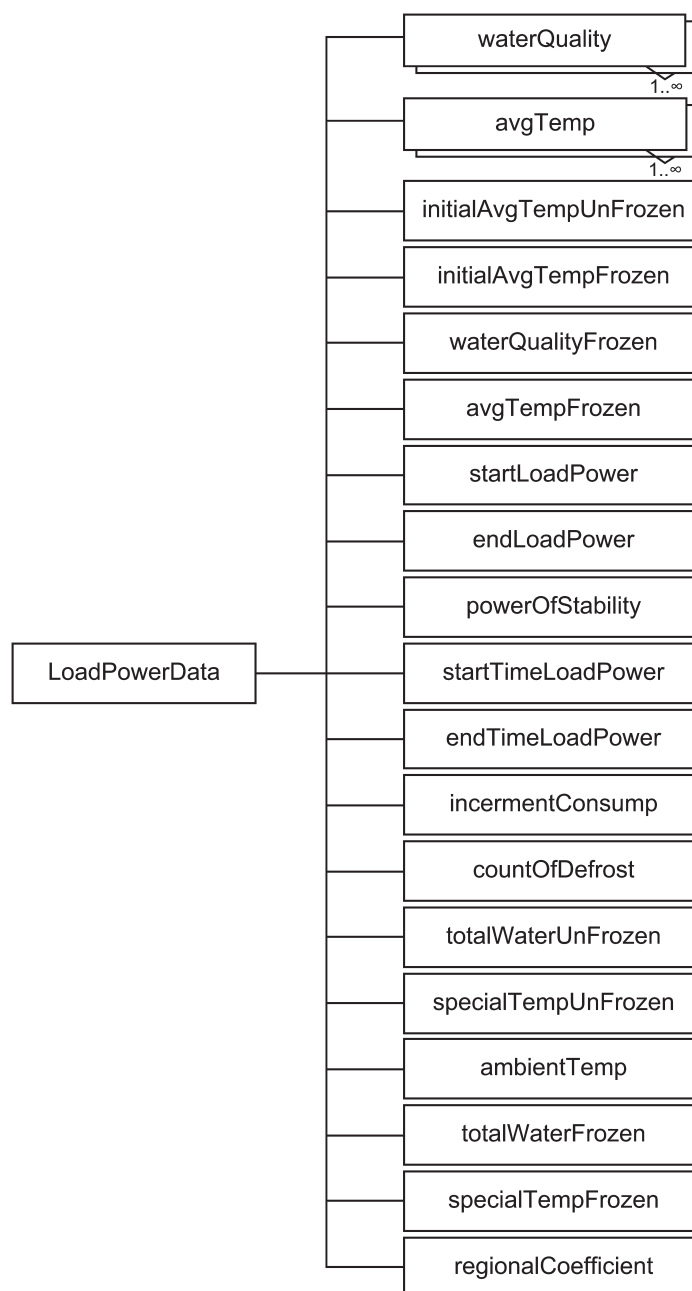


Figure 6—Structure of LoadPowerData element

The description of LoadPowerData element is shown in [Table 7](#).

Table 7—Description of LoadPowerInput element

Name	Type	Description	Use
waterQuality	array	The water weight near the temperature sensors placed in the non-freezing storage chamber.	Required
avgTemp	array	The average temperature of all sensors placed in the non-freezing storage chamber.	Required

Table continues

Table 7—Description of LoadPowerInput element (continued)

Name	Type	Description	Use
initialAvgTempUnFrozen	double	The average ambient temperature measured before draining in the non-freezing chamber.	Required
initialAvgTempFrozen	double	The average ambient temperature measured before draining in the freezing chamber.	Required
waterQualityFrozen	double	The water weight in the freezing storage chamber.	Required
avgTempFrozen	double	The average temperature in the freezing chamber measured by SS1 or SS2.	Required
startLoadPower	double	The energy meter reading when the loading is start.	Required
endLoadPower	double	The energy meter reading when the loading is stop.	Required
powerStability	double	The power when the data is steady.	Required
startTimeLoadPower	string	The start time of loading process, its format is “yyyy- MM-dd HH:mm:ss”.	Required
endTimeLoadPower	string	The end time of loading process, its format is “yyyy- MM-dd HH:mm:ss”.	Required
incementPConsump	double	The power consumption at defrosting/recovery period.	Required
countDefrost	integer	The number of defrosting/recovery periods during loading.	Required
totalWaterUnfrozen	double	The total water volume in the non-freezing chamber.	Required
specialTempUnfrozen	double	The characteristics temperature in the non-freezing chamber.	Required
ambientTemp	double	Standard ambient temperature.	Required
totalWaterFrozen	double	The total water volume in the freezing chamber.	Required
specialTempFrozen	double	The characteristics temperature in the freezing chamber.	Required
regionalCoefficient	double	The coefficient of the load.	Required

5.4.3.1.4 HeaterInputData

The HeaterInputData element contains all the information involved in calculating the power consumption of an anti-condensation heater, including the corresponding power at different ambient temperatures. Its element structure is illustrated in Figure 7.

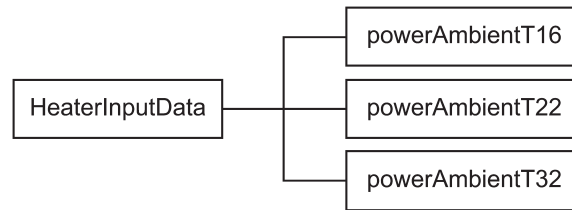


Figure 7—Structure of HeaterInputData element

The description of HeaterInputData element is shown in Table 8.

Table 8—Description of HeaterInputData element

Name	Type	Description	Use
powerAmbientT16	double	The power when the ambient temperature is 16 °C.	Required
powerAmbientT22	double	The power when the ambient temperature is 22 °C.	Required
powerAmbientT32	double	The power when the ambient temperature is 32 °C.	Required

5.4.3.2 Interface description

5.4.3.2.1 storageTemperature

The storageTemperature process consists of the following:

- Introduction:** Get the storage temperature of the refrigeration appliance.
- Request:** The POST request format is shown in [Figure 8](#).

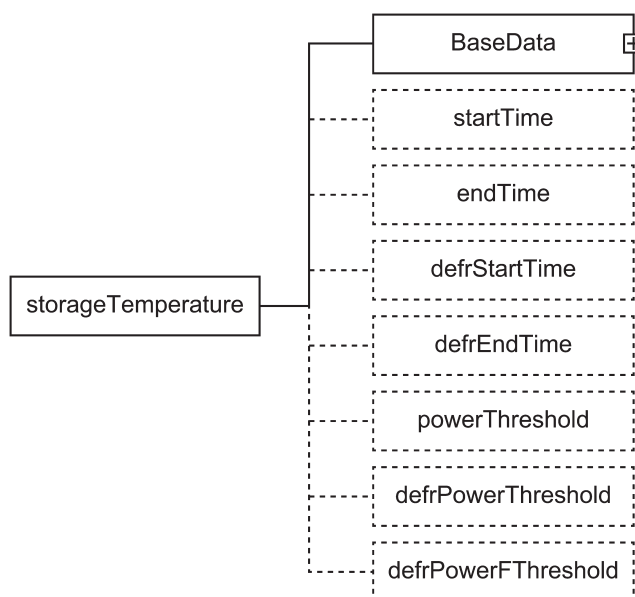


Figure 8—storageTemperature request parameters

The storageTemperature’s request parameters are shown in [Table 9](#).

Table 9—Description of storageTemperature request parameters

Name	Type	Description	Use
BaseData	BaseData	See 5.4.3.1.1	Required
startTime	string	The start time of the stable range, its format is “yyyy-MM-dd HH:mm:ss”	Optional
endTime	string	The end time of the stable range, its format is “yyyy-MM-dd HH:mm:ss”	Optional
defrStartTime	string	The start time of the defrosting range, its format is “yyyy-MM-dd HH:mm:ss”	Optional
defrEndTime	string	The end time of the defrosting range, its format is “yyyy-MM-dd HH:mm:ss”	Optional

Table continues

Table 9—Description of storageTemperature request parameters (continued)

Name	Type	Description	Use
powerThreshold	double	When power is less than the powerThreshold, the tested household appliance is assumed to be halted. Its unit is “W” and default value is 5.	Optional
defrPowerThreshold	double	When power is less than the defrPowerThreshold, the tested household appliance is assumed to be non-defrosting. Its unit is “W” and the default value is 140.	Optional
defrPowerFThreshold	double	When power factor is less than the defrPowerFThreshold, the tested household appliance is assumed to be non-defrosting. Its default value is 0.9.	Optional

- c) **Response:** The response information contains the status code, message, and response data. The response data includes a ResultData element that stores each refrigerator’s chamber’s storage temperature calculation results. Its element structure is shown in Figure 9.

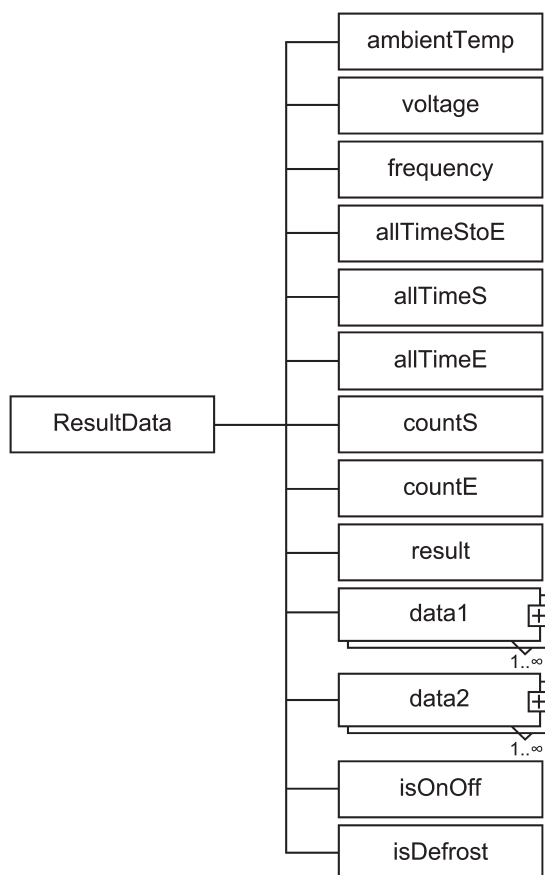


Figure 9—Structure of ResultData element

The description of the ResultData element is shown in Table 10.

Table 10—Description of ResultData element

Name	Type	Description	Use
ambientTemp	double	The ambient temperature of the laboratory.	Required
voltage	double	The voltage when the refrigerator is working.	Required
frequency	double	The frequency when the refrigerator is working.	Required
allTimeStoE	double	The total time from stage S to E.	Required
allTimeS	double	The total time at stage S.	Required
allTimeE	double	The total time at stage E.	Required
countS	integer	The number of temperature control cycles at stage S.	Required
countE	integer	The number of temperature control cycles at stage E.	Required
result	integer	The judgment result of temperature control cycle.	Required
data1	CompartmentData1	See 5.4.3.2.1.1.	Required
data2	CompartmentData2	See 5.4.3.2.1.3.	Required
isOnOff	boolean	Whether there is a on/off point; 1 represents yes and 0 represents no.	Required
isDefrost	boolean	Whether there is a defrosting stage; 1 represents yes and 0 represents no.	Required

5.4.3.2.1.1 CompartmentData1

The CompartmentData1 element contains relevant data within the chamber that does not need to be judged stable, which is composed of the chamber's name, average temperature, and characteristic temperature judgment. Its element structure is shown in Figure 10.

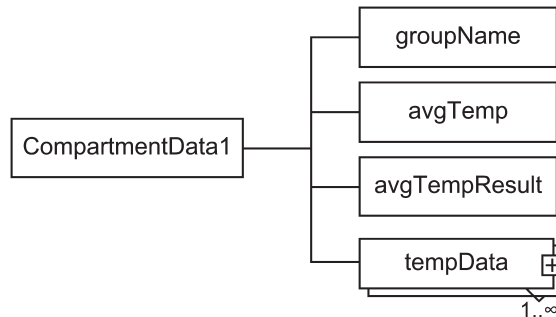


Figure 10—Structure of CompartmentData1 element

The description of the CompartmentData1 element is shown in Table 11.

Table 11—Description of CompartmentData1 element

Name	Type	Description	Use
groupName	string	The name of the group.	Required
avgTemp	double	The average temperature of a chamber.	Required
avgTempResult	integer	The characteristic judgment of a chamber's average temperature.	Required
tempData	CompartmentTempData1	See 5.4.3.2.1.1	Required

5.4.3.2.1.2 CompartmentTempData1

The CompartmentTempData1 element contains the temperature sensor data in the room that does not need to be judged stable, which is composed of the sensor name, the temperature condition of the stage SE, etc. Its element structure is shown in Figure 11.

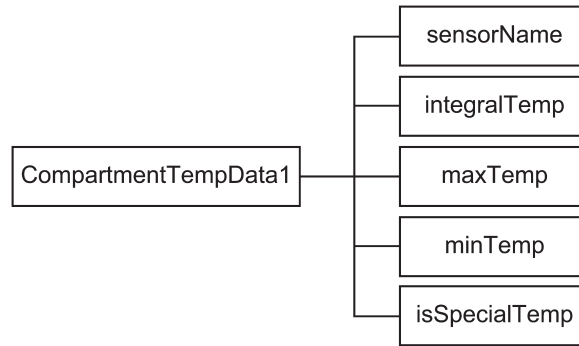


Figure 11—Structure of CompartmentTempData1 element

The description of the CompartmentTempData1 element is shown in Table 12.

Table 12—Description of CompartmentTempData1 element

Name	Type	Description	Use
sensorName	string	The name of the sensor.	Required
integralTemp	double	The integral average temperature from stage S to E.	Required
maxTemp	double	The maximum temperature from stage S to E.	Required
minTemp	double	The minimum temperature from stage S to E.	Required
isSpecialTemp	boolean	Whether the characteristic temperature meets the requirements; 1 represents yes and 0 represents no.	Required

5.4.3.2.1.3 CompartmentData2

CompartmentData2 element contains the relevant data within the judging chamber, consisting of sub-elements such as the chamber name, number of temperature control cycles, and temperature of the stage SE. Its element structure is shown in Figure 12.

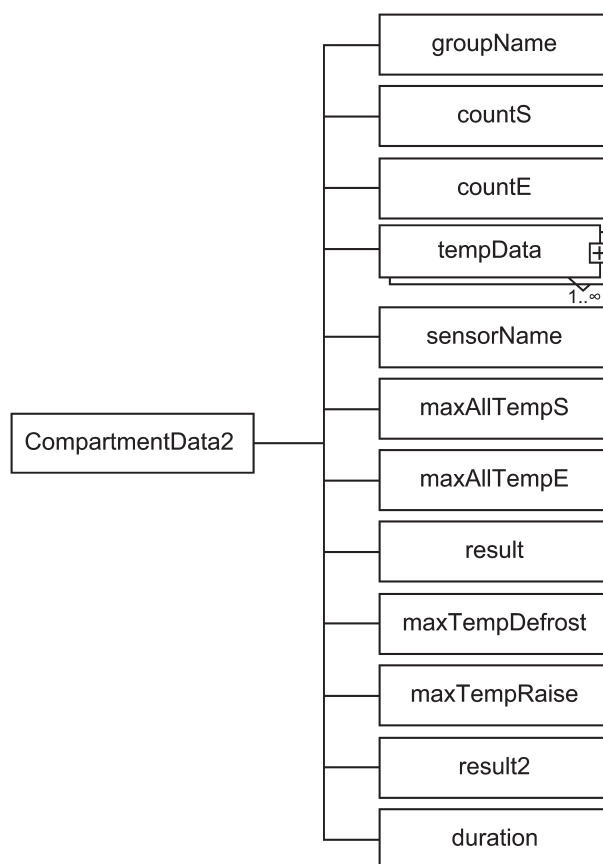


Figure 12—Structure of CompartmentData2 element

The description of the CompartmentData2 element is shown in Table 13.

Table 13—Description of CompartmentData2 element

Name	Type	Description	Use
groupName	string	The name of the group.	Required
countS	integer	The number of temperature control cycles at stage S.	Required
countE	integer	The number of temperature control cycles at stage E.	Required
tempData	CompartmentTempData2	See 5.4.3.2.1.4	Required
sensorName	string	The sensor name of the maximum temperature.	Required
maxAllTempS	double	The maximum temperature at stage S.	Required
maxAllTempE	double	The maximum temperature at stage E.	Required
result	integer	Whether the maximum temperature is less than -18 from stage S to E; 1 represents yes and 0 represents no.	Required
maxTempDefrost	double	The maximum temperature during defrosting.	Required
maxTempRaise	double	The maximum temperature beyond the reference temperature.	Required
result2	boolean	The results of the highest temperature appreciation; 1 represents pass and 0 represents fail.	Required
Duration	double	The duration beyond the reference temperature.	Required

5.4.3.2.1.4 CompartmentTempData2

The CompartmentTempData2 element contains the temperature in the chamber that needs to be judged, which is composed of the sensor name, the temperature of the stage S or stage E, etc. Its element structure is shown in Figure 13.

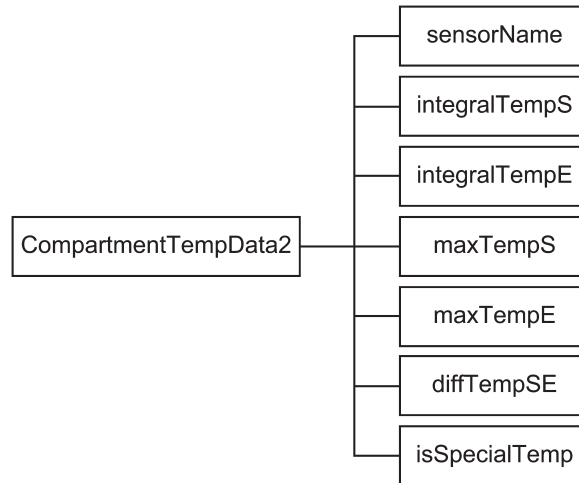


Figure 13—Structure of CompartmentTempData2 element

The description of the CompartmentTempData2 element is shown in Table 14.

Table 14—Description of CompartmentTempData2 element

Name	Type	Description	Use
sensorName	string	The name of the sensor.	Required
integralTempS	double	The integral average temperature at stage S.	Required
integralTempE	double	The integral average temperature at stage E.	Required
maxTempS	double	The maximum temperature at stage S.	Required
maxTempE	double	The maximum temperature at stage E.	Required
diffTempSE	double	The difference of integral temperature between stages S and E.	Required
isSpecialTemp	boolean	The judgment result of temperature difference; 1 represents stable and 0 represents unstable.	Required

- a) **Example:** The following is an example response file:

```

{
  "code": 200,
  "message": "success",
  "data": {
    "ResultData": {
      "ambientTemp": 32.28,
      "voltage": 100.11,
      "frequency": 60.05,
      "allTimeStoE": 16.986,
      "allTimeS": 3.995,
      "allTimeE": 4.019,
    }
  }
}
  
```