

Software for computers important to safety for nuclear power plants —

**Part 2: Software aspects of defence
against common cause failures, use of
software tools and of pre-developed
software**

ICS 27.120.20; 35.080

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW

National foreword

This British Standard reproduces verbatim IEC 60880-2:2000 and implements it as the UK national standard.

The UK participation in its preparation was entrusted to Technical Committee NCE/8, Reactor instrumentation, which has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international/European committee any enquiries on the interpretation, or proposals for change, and keep the UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

A list of organizations represented on this committee can be obtained on request to its secretary.

From 1 January 1997, all IEC publications have the number 60000 added to the old number. For instance, IEC 27-1 has been renumbered as IEC 60027-1. For a period of time during the change over from one numbering system to the other, publications may contain identifiers from both systems.

Cross-references

The British Standards which implement these international publications referred to in this document may be found in the BSI Standards Catalogue under the section entitled “International Standards Correspondence Index”, or by using the “Find” facility of the BSI Standards Electronic Catalogue.

A British Standard does not purport to include all the necessary provisions of a contract. Users of British Standards are responsible for their correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, the IEC title page, pages 2 to 40, an inside back cover and a back cover.

The BSI copyright notice displayed in this document indicates when the document was last issued.

This British Standard, having been prepared under the direction of the Engineering Sector Committee, was published under the authority of the Standards Committee and comes into effect on 15 June 2001

© BSI 06-2001

ISBN 0 580 36878 5

Amendments issued since publication

Amd. No.	Date	Comments

**NORME
INTERNATIONALE
INTERNATIONAL
STANDARD**

**CEI
IEC**

60880-2

Première édition
First edition
2000-12

**Logiciel pour les calculateurs de sûreté
des centrales nucléaires –**

**Partie 2:
Défense contre les défaillances de cause
commune provoquées par le logiciel,
utilisation d'outils logiciels et de logiciels
prédéveloppés**

**Software for computers important to safety
for nuclear power plants –**

**Part 2:
Software aspects of defence against common
cause failures, use of software tools and of
pre-developed software**



Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

CONTENTS

	Page
FOREWORD	4
INTRODUCTION	5
Clause	
1 Scope and object	6
2 Normative references	6
3 Definitions and abbreviations	7
4 Requirements and recommendations	10
4.1 Defences against common cause failure due to software	10
4.1.1 Introduction	10
4.1.2 Design of software against CCF	11
4.1.3 Sources and effects of CCF due to software	11
4.1.4 Implementation of diversity	12
4.1.5 Balance of drawbacks and benefits connected with the use of diversity	13
4.2 Software tools for the development of software	13
4.2.1 Introduction	13
4.2.2 Selection of tools	14
4.2.3 Requirements for tools	14
4.3 Qualification of pre-developed software	19
4.3.1 Introduction	19
4.3.2 General requirements	20
4.3.3 Evaluation and assessment process	20
4.3.4 Requirements for integration in the system and maintenance of PDS	26
Annex A (informative) Considerations of CCF and diversity	29
A.1 CCF due to software	29
A.2 Potential CCF causes and effects	29
A.3 CCF defences	30
A.4 Demonstration of correctness	31
A.5 Diversity features	31
A.6 Drawbacks, benefits and justification of diversity	32
Annex B (informative) IEC 60880 requirements for the use and qualification of software tools	33
Annex C (informative) Tools for production and checking of specification, design and code	34
C.1 Constructive tools	34
C.2 Analytical tools	35

	Page
Annex D (informative) IEC 60880 requirements concerning PDS	36
D.1 Summary of IEC 60880 requirements concerning the PDS	36
D.2 Documentation for the evaluation of the PDS	36
D.3 Guidance for selecting applicable IEC 60880 requirements	37
D.4 Guidance for graduating non-conformities and compensating factors	37
D.5 Collection and validation of data on the operational history	38
 Bibliography	 40
 Figure 1 – Outline of the qualification process of pre-developed software	 27
Figure 2 – Relation of PDS evaluation and assessment with the qualification plan of the system in which it is integrated.....	28

INTERNATIONAL ELECTROTECHNICAL COMMISSION

SOFTWARE FOR COMPUTERS IMPORTANT TO SAFETY FOR NUCLEAR POWER PLANTS –

Part 2: Software aspects of defence against common cause failures, use of software tools and of pre-developed software

FOREWORD

- 1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, the IEC publishes International Standards. Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. The IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of the IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested National Committees.
- 3) The documents produced have the form of recommendations for international use and are published in the form of standards, technical specifications, technical reports or guides and they are accepted by the National Committees in that sense.
- 4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
- 5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.
- 6) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. The IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 60880-2 has been prepared by subcommittee 45A: Reactor instrumentation, of IEC technical committee 45: Nuclear instrumentation.

The text of this standard is based on the following documents:

FDIS	Report on voting
45A/402/FDIS	45A/406/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 3.

Annexes A, B, C and D are for information only.

When IEC 60880 (1986) is revised, it will be published as IEC 60880-1.

The committee has decided that the contents of this publication will remain unchanged until 2006. At this date, the publication will be

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

INTRODUCTION

This part of IEC 60880 provides requirements for software for computer-based safety systems in nuclear power plants and should be read in conjunction with IEC 60880. This part includes requirements on several topics arising from advances in technology and experience of software systems with a role in safety since the publication of IEC 60880.

Defence against software faults which can lead to Common Cause Failure (CCF) of computer-based systems has to be provided in accordance with the IAEA Safety Series No. 50-C-D, and technical requirements arising from these provisions are given. There has also been a rapid growth in the use of computerized support tools, and recommendations for their use are given. The practical use of software for computer-based safety systems often depends on the use of pre-developed software, and requirements and recommendations in this connection are given.

This part of IEC 60880 references the basic safety publication IEC 61513.

This part applies to Category A I&C functions and associated systems and equipment (FSE) of IEC 61226.

NOTE According to the classification scheme of IEC 61513, category A functions are implemented in class 1 systems, which correspond to the safety systems of the IAEA.

SOFTWARE FOR COMPUTERS IMPORTANT TO SAFETY FOR NUCLEAR POWER PLANTS –

Part 2: Software aspects of defence against common cause failures, use of software tools and of pre-developed software

1 Scope and object

This part of IEC 60880 is applicable to the software of computer-based safety systems in nuclear power plants. It gives requirements for software for category A functions, systems and associated equipment (FSE). It addresses

- defence against common cause failures, caused by software;
- automated tools for the development of software important to safety; and
- use of pre-developed software.

Preparation and confirmation of data is dealt with within these three topics as appropriate.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of IEC 60880. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of IEC 60880 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of IEC and ISO maintain registers of currently valid International Standards.

IEC 60880:1986, *Software for computers in the safety systems of nuclear power stations*

IEC 61226:1993, *Nuclear power plants – Instrumentation and control systems important for safety – Classification*

IEC 61508-4:1998, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 4: Definitions and abbreviations*

IEC 61513: —, *Nuclear power plants – Instrumentation and control for systems important to safety – General requirements for systems* ¹⁾

ISO/IEC 9126:1991, *Information technology – Software product evaluation – Quality characteristics and guidelines for their use*

IAEA 50-C-D (rev 1):1988, *Code on the safety of Nuclear Power Plants: Design*

IAEA 50-SG-D11:1986, *General design safety principles for nuclear power plants – A safety guide*

IEEE 610:1990, *Standard glossary of software engineering terminology*

¹⁾ To be published

3 Definitions and abbreviations

For the purposes of this part of IEC 60880, the terms and definitions given in IEC 60880 and in the IAEA Safety Series No. 50-C-D, as well as the following terms and definitions apply.

NOTE 1 “Shall”, “should” and “may” are used in accordance with IEC conventions.

NOTE 2 For clarification of the terms **error**, **fault**, **failure** and **signal trajectory** the following example is given.

If a person or process makes an **error** in producing something, this will result in a **fault** in the product. When the product is used, it may be satisfactory, or it may fail, if the **fault** is not corrected. If the use challenges the **fault**, the product will fail if no other defence prevents the **failure**. A **failure** is due to both a **fault** and a challenge, with no other defence operating. For software, a challenge to a **fault** is provided by a **signal trajectory**.

3.1

animation

process by which the behaviour defined by a specification is displayed with actual values derived from the stated behaviour expressions and from some input values

3.2

application function

function of an I&C system that performs a task related to the process being controlled rather than to the functioning of the system itself
[derived from 2.1 of IEC 60880]

3.3

channel

separate path along which information flows through a redundant or distributed system. That path may also contain redundancy

3.4

common cause failure (CCF)

failure which is the result of one or more events, causing coincident failures of two or more separate channels in a multiple channel system or in multiple systems, leading to system(s) failure
[3.6.10 of IEC 61508-4, modified]

NOTE 1 Depending on the context, a CCF may be considered at the system level or at the level of the systems which constitute a safety group.

NOTE 2 See definition of failure (3.8).

3.5

data

representation of information or instructions in a manner suitable for communication, interpretation, or processing by computers

[adapted from IEEE 610]

NOTE Data which are required to define parameters and to instantiate application and service functions in the system are called “application data”.

3.6

diversity

existence of two or more different ways or means of achieving a specified objective. Diversity is specifically provided as a defence against common mode failure. It may be achieved by providing systems that are physically different from each other, or by functional diversity, where similar systems achieve the specified objective in different ways (see clause 3 of IEC 61226)

3.7

dynamic analysis

process of evaluating a system or component based on its behaviour during execution. In contrast to static analysis (see IEEE 610)

3.8

failure

failure occurs when the delivered service deviates from the intended service

NOTE A failure is the result of a hardware fault, software fault, system fault, or operator or maintenance error, and the associated signal trajectory which results in the failure.

3.9

fault

defect in a hardware, software, or system component. Faults are subdivided into random faults (for example, resulting from wearing out of hardware), and systematic faults (for example, introduced in the design), and for software these include coding errors and specification errors

NOTE A fault (notably a design fault) may remain undetected in a part of the system until specific conditions or signal trajectories affecting that part of the system, are such that the result produced does not conform to the intended function. This results in a failure of that part of the system.

3.10

functional diversity

application of the diversity at the functional level (for example, to have trip activation on both pressure and temperature limit)

3.11

functions, and associated systems and equipment (FSE)

functions are carried out for a purpose or to achieve a goal. The associated systems and equipment are the collections of components and the components themselves that are employed to achieve the functions (see clause 3 of IEC 61226)

3.12

human error (or mistake)

human action or a procedure that produces an unintended result

3.13

library

collection of related software elements that are grouped together, but which are individually selected for inclusion in the final software product

3.14

n-version or multi-version software

set of different programs, known as versions, developed to meet a common requirement and common acceptance test. Concurrent and independent execution of these versions takes place, generally in redundant hardware. Identical inputs in test systems or corresponding inputs in redundant systems are used. A predetermined strategy such as voting is used to decide between conflicting outputs in different versions

3.15

operational system software

software running on the target processor during operation, such as input/output drivers and services, interrupt management, scheduler, communication drivers, application-oriented libraries, on-line diagnostic, redundancy and graceful degradation management

3.16

postulated initiating event (PIE)

identified events that lead to anticipated operational occurrences or accident conditions and their consequential failure effects

[IAEA 50-C-D]

3.17

pre-developed software (PDS)

software which already exists is available as commercial or proprietary product and is being considered for use in a computer-based system

3.18

reusable

pertaining to a software module that can be used in more than one computer program or software system (derived from IEEE 610)

3.19

signal trajectory

time histories of all equipment conditions, internal states, input signals, and operator inputs which determine the outputs of a system

3.20

software version

instance of a software product derived by modification or correction of a preceding software product

3.21

specification

document that specifies, in a complete, precise, verifiable manner, the requirements, design behaviour or other characteristics of a system or component and, often, the procedures for determining whether these provisions have been satisfied (see IEEE 610)

NOTE There are different types of specifications, for example, software requirements specification or design specification.

3.22

static analysis

process of evaluating a system or component based on its form, structure, content or documentation. In contrast to dynamic analysis

3.23

support software

software that aids in the development, test, or maintenance of other software such as compilers, code generators, simulators, off-line diagnostic, initializers (see 4.2)

3.24

system software

software designed for a specific computer system or family of computer systems to facilitate the operation and maintenance of the computer system and associated programs, for example, operating systems, compilers, utilities. System software is usually composed of operational system software and support software

4 Requirements and recommendations

In the following, the term "requirements" is used as an inclusive term for both requirements and recommendations. The distinction appears at the level of the individual clauses where requirements are expressed by "shall" and recommendations by "should".

4.1 Defences against common cause failure due to software

This subclause provides requirements for defences against software design and coding faults which can lead to common cause failures (CCF) of functions classified as category A according to IEC 61226.

4.1.1 Introduction

CCF may occur in the I&C architecture's systems and equipment implementing different lines of defence against the same PIE (see 5.3.1 of IEC 61513). Software by itself does not have a CCF mode. CCF is related to system failures arising from faults in the functional requirements, system design, or in the software.

Defence in depth is required by the IAEA (see 204 of IAEA 50-C-D) to be applied to all safety activities, whether organizational, behavioural or design related, to ensure that there are overlapping defences so that if a failure should occur in a subsystem, it would be compensated for, or corrected in the integral system.

The single-failure criterion (see 329 to 336 of IAEA 50-C-D) requires that the assembly of safety systems have the ability to meet its purpose despite a single random failure assumed to occur anywhere in the assembly.

Software faults are systematic, not random, faults and, therefore, the single-failure criterion can not be applied to the software design of a system in the same manner as it has been applied for hardware. When the defence-in-depth concept is applied, possible effects of CCF due to software inside each defence layer and between redundant layers have to be considered and appropriate counter-measures have to be adopted throughout the development process and in the evaluation processes, for example,

- a) in the development, verification and validation of each individual defence layer; and
- b) in the evaluation of the independence and diversity of redundant defence layers.

A means of enhancing the reliability of some systems and reducing the potential for certain CCFs is the use of diversity (see 337 to 339 of IAEA 50-C-D Rev 1).

4.1.1.1 Requirements of IEC 60880

Subclause 4.2 of IEC 60880 requires the system reliability requirements to be the background to the description of the computer system configuration, and refers to A.2.2 of IEC 60880. This latter subclause indicates relevant factors (including defence in depth, graceful degradation, management of failures, functional diversity, software diversity, spatial separation and modularization, decoupling, logical separation) that might be used to avoid CCF, but does not specifically address all these factors. Therefore, 4.1 of this part of IEC 60880 specifies the way to assess the potential for CCF caused by software and also specifies techniques to avoid the occurrence of CCF.

4.1.1.2 Rationale for defence against CCF due to software

The rationale for defence against software faults is that any software fault will remain in the system or channel concerned until detected and corrected, and can cause failure if a specific signal trajectory challenges it. If two or more systems or channels implementing different lines of defence for the same PIE (see 5.3.1.5 of IEC 61513) contain the fault, and are exposed to specific signal trajectories within a sensitive time period, both (or all) systems or channels can fail, which is called a CCF. A more detailed description of these conditions is given in clause A.1.

The potential for CCF due to software should therefore be considered during design. If postulated conditions of CCF can be foreseen, design changes and defence features, including software diversity, may be needed for protection against CCF due to software.

The degree of improvement of defence against CCF and improvement in reliability that can be achieved by diversity cannot be quantified. Judgement is required based on an evaluation of the qualitative reliability which the software can achieve.

If human errors are made before software design starts, they may lead to faults of requirements and potential system failures against which software engineering alone cannot provide a defence. Defence against such CCF is discussed at the system level in 5.3.1.5 of IEC 61513.

If human errors are made during the software engineering process, they may lead to software faults and potential system failures. Where such faults lead to the failure of more than one line of protection the failures are considered to be CCFs due to software.

4.1.2 Design of software against CCF

The basic, and most important, defence against CCF due to software is to produce software of the highest quality, i.e. as error-free as possible. The extent of coverage of self-monitoring features, such as for data plausibility, parameter range checking, and loop timing etc. as addressed by 4.8, 5.1 and A.2.8 in IEC 60880 is a further important factor in limiting the potential for CCF due to software.

Requirements to achieve highly reliable software with self-monitoring features are given in IEC 60880 and the following paragraphs of this standard.

The use of well-developed software engineering methods with software tool support for software development and verification can help to reduce the number of human design decisions and so potentially reduce the number of faults in the developed software.

4.1.3 Sources and effects of CCF due to software

4.1.3.1 An analysis of the potential for CCF due to software shall be performed and documented at the system level and/or at the level of the total I&C architecture of the I&C systems important to safety of the NPP.

NOTE 1 Requirements on the I&C architecture are given in 5.3.1 of IEC 61513.

NOTE 2 Requirements on the architecture of the individual I&C systems are given in 6.1.2 of IEC 61513.

4.1.3.2 The analysis should include the following steps:

- a) identification of the software components used in the system or I&C architecture;
- b) analysis of the potential CCF due to these components within the system or the I&C architecture;
- c) analysis of the possible effects of these CCF.

NOTE Performing an analysis of the potential effects of faults does not obviate the need to perform verification and validation activities as required by IEC 60880. The purpose of such an analysis is to reveal any weaknesses in the design, and hence initiate changes to the design, and/or to improve confidence in the software design.

4.1.3.3 If common modules are used in more than one system, they shall be identified and assurance of the reliability of such common modules shall be assessed. Methods supporting the demonstration of correctness are given in clause A.4.

4.1.3.4 Data transmitted inside a computer-based system or between computer-based systems shall be identified. An analysis shall be performed to determine if faulty data can lead to a CCF in receiving computers or systems.

4.1.3.5 The potential for plant conditions to subject the same software running in different hardware to identical and simultaneous signal trajectories and hence reveal the same software fault in several channels or functional paths shall be assessed.

NOTE Failures may be caused by signal trajectories which were not considered during the design, verification and validation of the software of the individual channels or systems.

4.1.3.6 Software maintenance activities have the potential to cause CCF and the processes used for software or data change assessment should provide assurance that such faults are not introduced.

4.1.3.7 The analysis of potential CCF due to software shall be performed and documented as part of the assessment of the defence against CCF of the design of the I&C architecture (see 5.3.3 of IEC 61513).

4.1.3.8 If the analysis identifies an unacceptable threat arising from CCF due to software then the design of the software or of the I&C architecture shall be improved. Methods supporting the implementation of CCF defences are given in clause A.3 and methods supporting the implementation of diversity features are given in clause A.5.

4.1.4 Implementation of diversity

4.1.4.1 Implementation of diversity should use independent systems with functional diversity. If functional diversity is not appropriate or possible, the use of system diversity, diverse software features and diverse design approaches should be considered. Features of importance are given in clause A.5. The techniques chosen for defence against CCF shall be documented and justified according to the analysis made.

4.1.4.2 At the software level, defences against CCF should be based on an appropriate selection of techniques, such as

- a) guarantee of diversified operational conditions of the software;
- b) defences against error and failure propagation;