



DPC: 24/30472765 DC

BSI Group Headquarters

389 Chiswick High Road London W4 4AL

Tel: +44 (0)20 8996 9000

Fax: +44 (0)20 8996 7400

www.bsigroup.com

Date: 05 April 2024

Origin: International

Latest date for receipt of comments: 29 May 2024

Project No. 2023/00850

Responsible committee: IST/15 Software and systems engineering

Interested committees:

Title: Draft BS EN ISO/IEC/IEEE 20582 Software and systems engineering - Capabilities of build and deployment tools.

Please notify admin.start@bsigroup.com if you are aware of any keywords that might assist in clarifying or identifying the standard or of the content of this standard:

- i) has any issues related to 3rd party IPR, patent or copyright
- ii) affects other national standard(s)
- iii) requires additional national guidance or information

WARNING: THIS IS A DRAFT AND MUST NOT BE REGARDED OR USED AS A BRITISH STANDARD. THIS DRAFT IS NOT CURRENT BEYOND 29 May 2024

This draft is issued to allow comments from interested parties; all comments will be given consideration prior to publication. No acknowledgement will normally be sent. **See overleaf for information on the submission of comments.**

No copying is allowed, in any form, without prior written permission from BSI except as permitted under the Copyright, Designs and Patent Act 1988 or for circulation within a nominating organization for briefing purposes. Electronic circulation is limited to dissemination by e-mail within such an organization by committee members.

Further copies of this draft may be purchased from BSI Shop <http://shop.bsigroup.com> or from BSI Customer Services, Tel: +44(0) 20 8996 9001 or email cservices@bsigroup.com. British, International and foreign standards are also available from BSI Customer Services.

Information on the co-operating organizations represented on the committees referenced above may be obtained from <https://standardsdevelopment.bsigroup.com/>.

E-mail: admin.start@bsigroup.com

Introduction

This draft standard is based on national and international discussions. Your comments on this draft are invited and will assist in the preparation of the consequent standard.

For international standards, comments will be reviewed by the relevant UK national committee before sending the consensus UK vote and comments to the international committee, which will then decide appropriate action. If the international standard is approved, it is usual for the text to be published as a British Standard.

For national standards, comments will be reviewed by the relevant UK national committee and the resulting standards published as a British Standard.

UK Vote

Please indicate whether you consider the UK should submit a negative (with supporting technical reasons) or positive vote on this draft. Please indicate if you are aware of any reason why this draft standard should not be published as a British Standard.

Submission of Comments

- **Annotated drafts are not acceptable and will be rejected.**
- All comments should be submitted online at <https://standardsdevelopment.bsigroup.com/>. You will need to register in order to comment.

Template for comments and secretariat observations

Date: xx/xx/20xx	Document: ISO/DIS xxxx
------------------	------------------------

1	2	(3)	4	5	(6)	7
MB	Clause No./ Subclause No./Annex (e.g. 3.1)	Paragraph/ Figure/Table/Note	Type of comment	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
	3.1	Definition 1	ed	Definition is ambiguous and needs clarifying.	Amend to read '...so that the mains connector to which no connection...'	
	6.4	Paragraph 2	te	The use of the UV photometer as an alternative cannot be supported as serious problems have been encountered in its use in the UK.	Delete reference to UV photometer.	



DRAFT International Standard

ISO/IEC DIS 20582

Software and systems engineering — Capabilities of build and deployment tools

ICS: 35.080

ISO/IEC JTC 1/SC 7

Secretariat: **BIS**

Voting begins on:
2024-04-05

Voting terminates on:
2024-06-28

This document is circulated as received from the committee secretariat.

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENTS AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

Reference number
ISO/IEC DIS 20582:2024(en)

Full document details: 24/30472765 DC

 **Documents**
Document Sharing Hub
© ISO/IEC 2024



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2024

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

© ISO/IEC 2024 – All rights reserved

Contents

Page

Foreword	v
Introduction	vi
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
3.1 Definitions	1
3.2 Abbreviations	2
4 Object model for build and deployment tools	2
4.1 Overview	2
4.2 Use case	3
4.2.1 Build, deployment and CI/CD in system lifecycle	3
4.2.2 Use cases of Build and deployment tool	4
4.2.3 Use case scenario	5
4.3 Object model for Build and deployment entity	9
4.3.1 Overview	9
4.3.2 Resource Entities	9
4.3.3 Control Entities	10
5 Entities used in building, deployment, and CI/CD	15
5.1 Overview	15
5.2 Abstract Control Entities	16
5.2.1 Overview	16
5.2.2 Link to External Library	16
5.2.3 Link to Source Code	16
5.2.4 Link to Internal Library	16
5.2.5 Link to Data	16
5.2.6 Link to Support Tool	17
5.2.7 Link to Server	17
5.2.8 Dependency	17
5.2.9 Procedure	17
5.2.10 Step	17
5.3 Build Entities	18
5.3.1 Build Dependency	18
5.3.2 Build Step	19
5.3.3 Build Procedure	19
5.3.4 Link To Translator	20
5.4 Deployment Entities	20
5.4.1 Deployment Dependency	20
5.4.2 Deployment Step	21
5.4.3 Deployment Procedure	21
5.4.4 Link To Installer	21
5.5 CI/CD entities	21
5.5.1 CI/CD Dependency	21
5.5.2 CI/CD Step	22
5.5.3 CI/CD Procedure	22
5.5.4 Link To Test Support Tool	23
6 Capabilities of Build and deployment tool	23
6.1 Overview	23
6.2 Space of dependency in building, deployment, and CI/CD	23
6.3 Build related Capabilities	24
6.3.1 Register Build Procedures	24
6.3.2 Execute Build Procedure	24
6.3.3 Confirm Result of Build Procedures	26
6.4 Deployment related Capabilities	26

ISO/IEC DIS 20582:2024(en)

6.4.1	Register Deployment Procedure.....	26
6.4.2	Execute Deployment Procedure.....	26
6.4.3	Confirm Result of Deployment Procedures.....	28
6.5	CI/CD related Capabilities.....	28
6.5.1	Register CI/CD Procedure.....	28
6.5.2	Execute CI/CD Procedure.....	29
6.5.3	Confirm Result of CI/CD Procedures.....	30
Annex A (informative) How to use this document with ISO/IEC 20741.....		31
Annex B (informative) Overview of the approach for this document.....		32
Bibliography.....		34

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

Build and deployment tools have long been indispensable tools but, nowadays, it plays an incomparably important role. In the past, build and deployment tools were simple mechanisms to create an executable code from the source code written in a programming language, which were usually consisted of a compiler and a linker bundled with the OS. Currently, mechanisms become much more complicated. An application software is built with the source code and the libraries which are managed across multiple servers. In addition, testing and operating a system typically requires deploying the system to a large number of servers. Similarly, changes in the development environment have become a factor in increasing the importance of the role of build and deployment tools.

Dramatic changes in the development process have also impacted to the increasing importance of the role of build and deployment tools. For example, repeated development called agile has become a common and needs to be released every few weeks. And, in order to realize a practice called DevOps, which aims to shorten the time from system development to normal operation, it is indispensable to automate the work with build and deployment tools.

Many modern build and deployment tools are available now on the market which are independent of the vendor that provides the OS. Because the modern build and deployment tools are able to configure as independent automation tools on top of compilers and linkers bundled with individual OS, as well as repositories that manage assets. So there are many options and it is necessary to choose the right one. In other words, not only tools commercially available from independent vendors, but also tools provided as open source are subject to selection and subject to this standard.

This document treats tasks and tools of change management and configuration management as out of scope. Building and deploying are usually triggered by changes to the source code, as a work procedure, it is closely related to change management and configuration management. However, while the information to be processed in change management and configuration management is mainly source code, in building and deploying, it is transformed to create the desired information. For this reason, change management and configuration management are excluded to avoid complications.

From a user's perspective, it is important to choose the right tool from a variety of build and deployment tools, especially in large organizations. It is imperative that tool evaluation and selection be done in accordance with fair and public standards. To this end, ISO / IEC 20741 was published in 2017 as a guideline for the process of evaluating and selecting software engineering tools. However, ISO / IEC 20741 does not mention the standard capabilities specific to build and deployment tools because it is generalized without depending on a specific tool category.

This document defines the requirements of capabilities of build and deployment tools and is intended to be combined with ISO / IEC 20741 to select the appropriate tool (see [Annex A](#)). For this purpose, this document provides a list of capabilities in the build and deployment tools with its category. The capabilities introduced in this document are collected from existing tools (see [Annex B](#)).

Build, Deploy, and CI/CD tools are separate and independent tools, and can logically be separate standards. However, in actual development work, they are used almost continuously as a series of work. Also, evaluation and selection are likely to be conducted from similar viewpoints at the same time. For this reason, this document does not treat them as separate standards, but summarizes them as a single standard.

Software and systems engineering — Capabilities of build and deployment tools

1 Scope

This document defines the requirements of capabilities of build and deployment tools to support and automate building, packaging, and deploying work.

This document is intended for use in evaluating and selecting build and deployment tools according to the procedures defined in ISO/IEC 20741. The evaluation and selection of build and deployment tools can derive objective and rational results by using these two standards together.

This document is independent of development methodologies and approaches (such as Waterfall, Agile, or DevOps). In other words, this document is applicable to DevOps, Agile, and Waterfall.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply. Some definitions refer to ISO/IEC/IEEE 32675:2022.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1 Definitions

3.1.1

build

Process of generating an executable and testable system from source versions or baselines.

[SOURCE: ISO/IEC/IEEE 32675:2022]

3.1.2

continuous integration

Technique that continually merges artifacts, including source code updates from all developers on a team, into a shared mainline to build and test the developed system.

[SOURCE: ISO/IEC/IEEE 32675:2022]

3.1.3

continuous delivery

Software engineering practices that allow for frequent releases of new systems (including software) to staging or various test environments through the use of automated tools.

[SOURCE: ISO/IEC/IEEE 32675:2022]

3.1.4

continuous deployment

Automated process of deploying changes to production by verifying intended features and validations to reduce risk.

[SOURCE: ISO/IEC/IEEE 32675:2022]

3.1.5

deployment

Stage of a life cycle in which a system is put into operation and transition issues are resolved

[SOURCE: ISO/IEC/IEEE 32675:2022]

3.1.6

package

To combine related components into a single, deployable item.

[SOURCE: ISO/IEC/IEEE 32675:2022]

A package is used to group elements and provides a namespace for the grouped elements.

[SOURCE: ISO/IEC 19505 2]

3.2 Abbreviations

CD continuous delivery, continuous deployment

[SOURCE: ISO/IEC/IEEE 32675:2022]

CI continuous integration

4 Object model for build and deployment tools

4.1 Overview

Generally, tools reduce manual effort by generating entities automatically to be created during the software development process. On expressing this as a model, it is reasonable to express it using three views: a package representing a model related to a process view, a package representing a model related to an entity view, and a package representing a model related to a tool view.

The tool area of build & deployment is the same, and can be expressed as a package that represents three views as shown in [Figure 1](#). Here, the use case model corresponds to the process view.

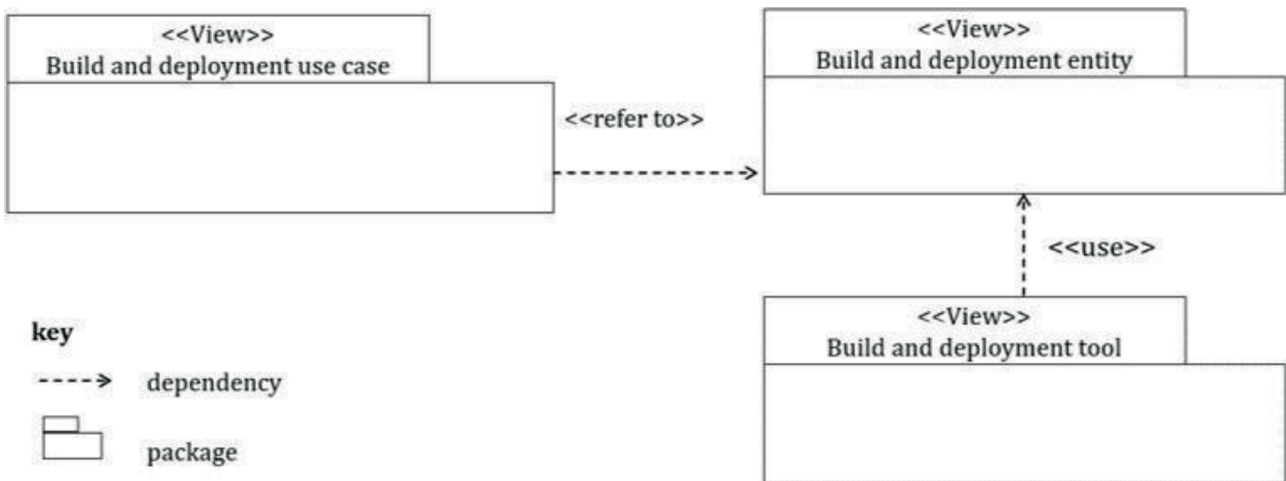


Figure 1 — Object model for build and deployment

The rest of this clause is based on this structure, with [4.2](#) defining use cases and [4.3](#) defining the structure of entities related to build & deployment. [Section 5](#) defines entities in more detail, based on which [section 6](#) specifies tool capabilities.

The object model diagrams, [Figure 1](#), [Figure 3](#) to [Figure 8](#), and [Figure 10](#) to [Figure 12](#) are described using Unified Modelling Language (UML) 2 (ISO/IEC 19505 2).

4.2 Use case

4.2.1 Build, deployment and CI/CD in system lifecycle

[Figure 2](#) shows the scope of build and deployment in the latter part of the system development process. Building is the use case of transforming the source code created by coding into an executable package. Deployment is the use case of setting the package created by build to operate in the test environment or production environment.

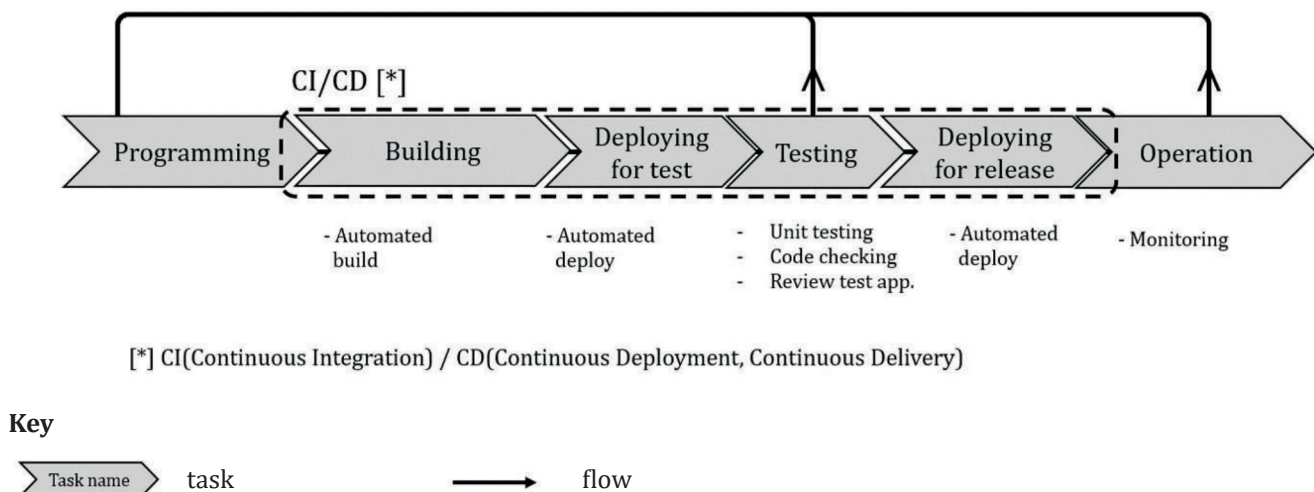


Figure 2 — Build, deployment and CI/CD in the system lifecycle

The details of each task described in [Figure 2](#) are shown below. In CI/CD, b), c), d), and e) are continuously and repeatedly performed by automation.

a) Programming

Developers create or modify source code or library according to application specifications or change order. To do this, both of change management tools and configuration management tools are usually used. The developer gets the source code as the personal version from the configuration management tools and reflects the changes to the original version.

This task is a trigger of building, however, is not in the scope of this document.

b) Building (include packaging)

Converting source code into an executable package.

c) Deploying for Test

Setting servers and the package to be executed in the test environment.

d) Testing

Perform dynamic testing of the package deployed in the test environment.

e) Deploying for release

Setting servers and the package to be executed the production environment.

f) Operation

Operating the package in the production environment.

4.2.2 Use cases of Build and deployment tool

Build and deployment tool is used in the following use cases (Building, Deployment, and CI/CD) and the following actors appear in the use cases, as shown in [Figure 3](#).

a) Developer

Members of the project who write the source code, build the source code, and create the package, by using Build and deployment tool.

b) Deployer

Members of the project who deploy the package by using Build and deployment tool to each environment.

c) Repository

Storages for resources such as source code and manage versions of resources.

d) Environment

Servers providing the hardware and software necessary for executing the package.

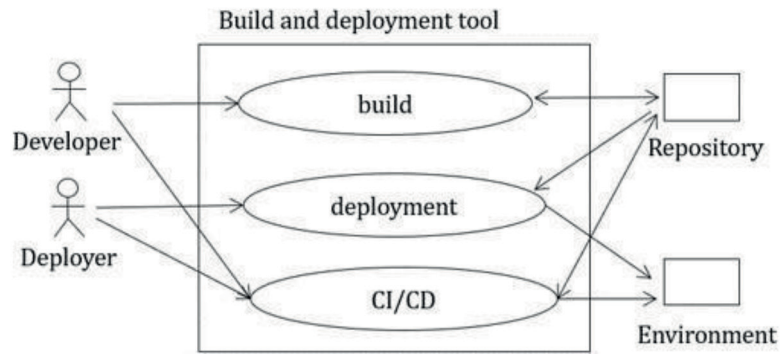


Figure 3 — Use case diagram of Build and deployment tool

4.2.3 Use case scenario

4.2.3.1 Use case scenario of building

Use case scenario of building is defined by using activity diagram shown as [figure 4](#).

Source repository is the repository that stores source code. External library repository is the repository storing libraries created outside the project applying the build and deployment tools. Package repository is the repository that stores libraries, executable packages, and data created in the project. Building is executed in the following procedure according to [Figure 4](#).

- Developer requests Build and deployment tool to build.
- Build and deployment tool gets the source code from Source repository. At the same time, the external library having a calling relationship is copied from External library repository, and the internal library is copied from the Package repository.
- Build and deployment tool invokes the translator to generate the package from source code. And then Build and deployment tool registers the package in Package repository.
- Build and deployment tool notifies Developer of the results of c).