



BSI Standards Publication

**Information technology —
Programming languages,
their environments and
system software interfaces
— Guidelines for language
bindings**

National foreword

This Published Document is the UK implementation of ISO/IEC TR 10182:2016. It supersedes BS ISO/IEC TR 10182:1993 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/5, Programming languages, their environments and system software interfaces.

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

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2016. Published by BSI Standards Limited 2016

ISBN 978 0 580 88135 0

ICS 35.060

Compliance with a British Standard cannot confer immunity from legal obligations.

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 31 March 2016.

Amendments issued since publication

Date	Text affected
------	---------------

**Information technology —
Programming languages, their
environments and system software
interfaces — Guidelines for language
bindings**

*Technologies de l'information — Langages de programmation, leurs
environnements et interfaces logicielles des systèmes — Techniques
d'interface pour les normes de langages de programmation*



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2016, Published in Switzerland

All rights reserved. Unless otherwise specified, 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
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword	iv
Introduction	v
1 Scope	1
2 Terms and definitions	1
2.1 Terms.....	1
2.2 Abbreviated terms	3
3 Overview of functional binding methods	3
3.1 Introduction to Methods.....	3
3.2 System Facility Standard Procedural Interface (Method 1).....	4
3.3 User-Defined Procedural Interfaces (Method 2).....	5
3.4 Programming Language Extensions with Native Syntax (Method 3).....	5
3.5 Programming Languages with Embedded Alien Syntax (Method 4).....	6
3.6 Binding Pre-Existing Language Elements (Method 5).....	6
3.7 Conclusions.....	6
4 Guidelines	7
4.1 Organizational Guidelines for Preparation of Language Bindings	7
4.2 General Technical Guidelines.....	9
4.3 Recommendations for Functional Specifications.....	9
4.4 Method-Dependent Guidelines for Language Bindings.....	10
4.4.1 Introduction to Method-Dependent Guidelines.....	10
4.4.2 Guidelines for Standard Procedural Interfaces	10
4.4.3 Guidelines for User Defined Procedural Interfaces	17
4.4.4 Guidelines for Programming Language Extensions with Native Syntax.....	18
4.4.5 Uses of Programming Languages with Embedded Alien Syntax.....	18
5 Future directions	18
Annex A (informative) Graphic Binding Examples	20
Annex B (informative) GKS Bindings Generic Issues	27
Bibliography	40

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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 document 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 and IEC 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 on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT), see the following URL: [Foreword — Supplementary information](#).

The committee responsible for this Technical Report is ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

This first edition of ISO/IEC TR 10182:2016 cancels and replaces the first edition of ISO/IEC TR 10182:1993, of which it constitutes a minor revision with the following changes:

- the references section has been deleted;
- minor editorial errors have been corrected.

Introduction

This Technical Report is a compilation of the experience and knowledge gained by the members of ISO/IEC JTC1/SC22/WG11 (Techniques for Bindings) from the generation of programmers' interfaces to FUNCTIONAL INTERFACE STANDARDS. Although current experience was derived from the fields of computer graphics and database management, the problems discussed are thought to be generally applicable for mappings of other functional interface standards to programming languages. This Technical Report is intended

- a) to identify the problems and conflicts which shall be resolved;
- b) to suggest guidelines for future use;
- c) to provide scope and direction to required additional work, such as common procedural calling mechanisms and data types; and
- d) as a historical record of past experiences and decisions.

This Technical Report is incomplete; the authors have concentrated on those areas where experience and expertise was readily available. The ideas and issues brought forward here emerged from more than 10 years of work, and are represented in International Standards.

[Clause 3](#) of this Technical Report contains the results of a survey of current methods used for language binding development. Characteristics of each method are given, followed by reasons for the selection of the method.

Application of the methods has suggested some guidelines that are presented in [Clause 4](#). [Clauses 3](#) and [4](#) contain documentation of the current state of language binding efforts; [Clause 5](#) addresses future directions for language bindings.

Circulation of this Technical Report is necessary at this stage, as input and discussion from representatives of ISO/IEC JTC1/SC21 (functional specification standards developers), ISO/IEC JTC1/SC24 (computer graphics standards developers), and ISO/IEC JTC1/SC22 (language standards developers) is urgently sought. The Technical Report in its current form may be useful for those about to embark on language binding developments.

Information technology — Programming languages, their environments and system software interfaces — Guidelines for language bindings

1 Scope

This Technical Report is based on experience gained in the standardization of two major areas in information processing. One area covers programming languages. The other area is composed of the services necessary to an application program to achieve its goal. The services are divided into coherent groups, each referred to as a SYSTEM FACILITY, that are accessed through a FUNCTIONAL INTERFACE. The specification of a system facility, referred to as a FUNCTIONAL SPECIFICATION, defines a collection of SYSTEM FUNCTIONS, each of which carries out some well-defined service.

Since in principle there is no reason why a particular system facility should not be used by a program, regardless of the language in which it is written, it is the practice of system facility specifiers to define an 'abstract' functional interface that is language independent. In this way, the concepts in a particular system facility may be refined by experts in that area without regard for language peculiarities. An internally coherent view of a particular system facility is defined, relating the system functions to each other in a consistent way and relating the system functions to other layers within the system facility, including protocols for communication with other objects in the total system.

However, if these two areas are standardized independently, it is not possible to guarantee that programs from one operating environment can be moved to another, even if the programs are written in a standard programming language and use only standard system facilities. A language binding of a system facility to a programming language provides language syntax that maps the system facility's functional interface. This allows a program written in the language to access the system functions constituting the system facility in a standard way. The purpose of a language binding is to achieve portability of a program that uses particular facilities in a particular language. Examples of system facilities that have had language bindings developed for them are GKS, NDL, and SQL (see Bibliography). It is anticipated that further language binding development will be required. Some system facilities currently being standardized have no language bindings and additional system facilities will be standardized. There is a possibility of $n \times m$ language bindings, where n is the number of languages and m the number of system facilities.

The scope of this Technical Report is to classify language binding methods, reporting on particular instances in detail, and to produce suggested guidelines for future language binding standards.

Note that the language bindings and the abstract facility interfaces shall have a compatible run time representation, but the abstract facility does not necessarily have to be written in the host language. For example, if the application program is using a Pascal language binding and the corresponding facility is written in FORTRAN, there shall be a compatible run time representation in that operating environment. How this compatibility is achieved is outside the scope of these guidelines. This is generally a property of the operating environment defined by the implementor, and is reviewed briefly in this Technical Report.

2 Terms and definitions

2.1 Terms

ABSTRACT SERVICE INTERFACE: An interface having an abstract definition that defines the format and the semantics of the function invoked independently of the concrete syntax (actual representation) of the values and the invocation mechanism.