INTERNATIONAL STANDARD

ISO/IEC 14834

First edition
1996-08-15

Information technology — Distributed Transaction Processing — The XA Specification

Technologies de l'information — Traitement transactionnel réparti — La spécification XA

Reference number
ISO/IEC 14834:1996(E)

Contents

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

List of Figures

List of Tables

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. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 14834 was prepared by X/Open Company Ltd. (as XO/CAE/91/300) and was adopted, under a special "fast-track procedure", by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

Appendix A forms an integral part of this International Standard. Appendices B to F are for information only.

Introduction

(This introduction is not a normative part of ISO/IEC 14834, Information technology—Distributed Transaction Processing—The XA Specification, but is included for information only.)

This International Standard specifies the bidirectional interface between a transaction manager and resource manager (the XA interface) in an X/Open Distributed Transaction Processing (DTP) environment. It is based on X/Open CAE Specification, Distributed Transaction Processing: The XA Specification (December 1991). This International Standard is technically identical to the X/Open version. For informative purposes, this International Standard also contains the text of the X/Open DTP Reference Model Version 3 which X/Open has published as a separate Guide.

Typographical Conventions

The following typographical conventions are used throughout this document:

- `Constant width` strings are code examples or literals and are to be typed just as they appear.

- *Italic* strings are used for emphasis or to identify the first instance of a word requiring definition. Italics also denote:
  - — variable names
  - — commands or utilities
  - — functions; these are shown as follows: *name*().
- The notation **"file.h"** indicates a header.
- The notation [ABCD] is the name of a return value.
- Ellipses (. . .) are used to show that additional arguments are optional.

Trademarks

X/Open® is a registered trade mark, and the "X" device is a trade mark, of X/Open Company Limited.

This page intentionally left blank

Information technology — Distributed Transaction Processing — The XA Specification

Chapter 1: General

1.1 Scope

This International Standard specifies the *XA interface*: the bidirectional interface between a transaction manager and a resource manager in an X/Open Distributed Transaction Processing (DTP) environment. The XA interface is not an ordinary Application Programming Interface (API); it is a system-level interface between DTP software components.

This International Standard is technically identical to X/Open CAE Specification, Distributed Transaction Processing: The XA Specification (December 1991). Like that specification, this International Standard does not define the full aspects of the DTP model that pertain to communication.

1.2 X/Open DTP Model

The X/Open Distributed Transaction Processing (DTP) model is a software architecture that allows multiple application programs to share resources provided by multiple resource managers, and allows their work to be coordinated into global transactions.

The full X/Open DTP model comprises five basic functional components:

- an Application Program (AP), which defines transaction boundaries and specifies actions that constitute a transaction
- Resource Managers (RMs) such as databases or file access systems, which provide access to resources
- a Transaction Manager (TM), which assigns identifiers to transactions, monitors their progress, and takes responsibility for transaction completion and for coordinating failure recovery.

- Communication Resource Managers (CRMs), which control communication between distributed applications within or across TM domains.
- a communication protocol, which provides the underlying communication services used by distributed applications and supported by CRMs.

1.3 Document Structure

Relevant definitions and other important concepts that pertain to this International Standard are discussed in Chapter 2. That chapter also defines the AP, TM, and RM in more detail, and describes their interaction. Chapter 3 is an overview of the XA interface, describing the situations in which each of the services is used. Chapter 4 discusses the data structures that are part of the XA interface. Reference manual pages for each routine in the XA interface are presented in Chapter 5; state tables follow in Chapter 6. Chapter 7 summarises the implications of this International Standard on the implementors of RMs and TMs; it also identifies features that are optional. Appendix A presents the contents of an **"xa.h"** header file in both ANSI C and Common Usage C. Appendix F contains a bibliography.

For informative purposes, this International Standard also contains the text of the X/Open DTP Reference Model Version 2 (November 1993) which X/Open publishes as a separate Guide. (See Appendix B, Appendix C, Appendix D, and Appendix E.)

1.4 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

1. ISO/IEC 8824:1990, *Information technology—Open Systems Interconnection—Specification of Abstract Syntax Notation One (ASN.1).*
2. ISO/IEC 8825:1990, *Information technology—Open Systems Interconnection—Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).*
3. ISO/IEC 9804:1994, *Information technology—Open Systems Interconnection—Service definition for the commitment, concurrency and recovery service element.*
4. ISO/IEC 9805-1:1994, *Information technology—Open Systems Interconnection—Protocol for the Commitment, Concurrency and Recovery service element: Protocol Specification.*
5. ISO/IEC 9899:1990, *Programming languages—C.*
6. ISO/IEC 10026-1:1992, *Information technology—Open Systems Interconnection—Distributed Transaction Processing—Part 1: OSI TP Model.*
7. ISO/IEC 10026-2:1996, *Information technology—Open Systems Interconnection—Distributed Transaction Processing—Part 2: OSI TP Service.*
8. ISO/IEC 10026-3:1996, *Information technology—Open Systems Interconnection—Distributed Transaction Processing—Part 3: Protocol Specification.*

See Appendix F for bibliographic references.

Chapter 2: Model and Definitions

This chapter discusses the XA interface in general terms and provides necessary background material for the rest of this International Standard. The chapter shows the relationship of the interface to the X/Open DTP model. The chapter also states the design assumptions that the interface uses and shows how the interface addresses common DTP concepts.

2.1 X/Open DTP Model

The boxes in Figure 2-1 are the functional components and the connecting lines are the interfaces between them. The arrows indicate the directions in which control may flow.

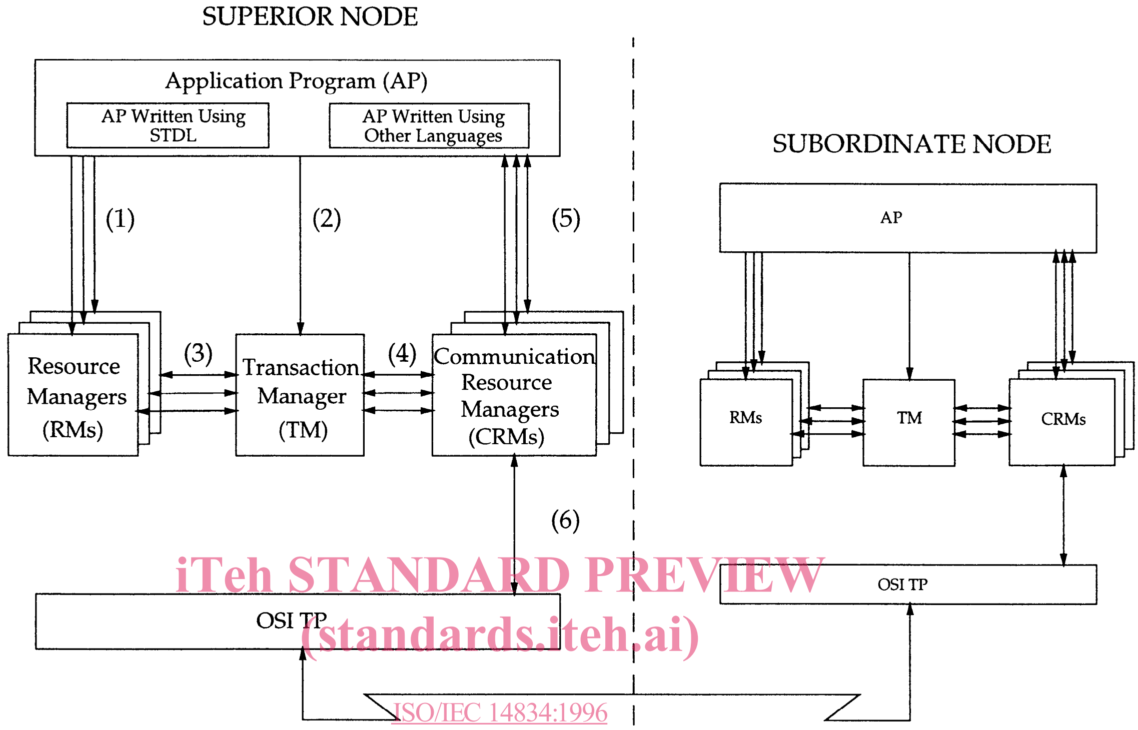

Figure 2-1 Functional Components and Interfaces

The numbers in brackets in Figure 2-1 represent the different X/Open interfaces that are used in the DTP model. The subject of this International Standard is interface (3): the XA interface by which TMs and RMs interact. Descriptions of the functional components relevant to this International Standard can be found in Section 2.2 on page 5. For more details of the the DTP model as shown in Figure 2-1, including definitions of all components and interfaces, see Appendix B, Appendix C, Appendix D, and Appendix E.

2.2 Definitions

2.2.1 Transaction

A transaction is a complete unit of work. It may comprise many computational tasks, which may include user interface, data retrieval, and communications. A typical transaction modifies shared resources. (The referenced OSI TP standard (model) defines transactions more precisely.)

Transactions must be able to be *rolled back*. A human user may roll back the transaction in response to a real-world event, such as a customer decision. A program can elect to roll back a transaction. For example, account number verification may fail or the account may fail a test of its balance. Transactions also roll back if a component of the system fails, keeping it from retrieving, communicating, or storing data. Every DTP software component subject to transaction control must be able to undo its work in a transaction that is rolled back at any time.

When the system determines that a transaction can complete without failure of any kind, it *commits* the transaction. This means that changes to shared resources take permanent effect. Either commitment or rollback results in a consistent state. *Completion* means either commitment or rollback.

2.2.2 Distributed Transaction Processing

Within the scope of this International Standard, DTP systems are those where work in support of a single transaction may occur across RMs. This has several implications:

- The system must have a way to refer to a transaction that encompasses all work done anywhere in the system.
- The decision to commit or roll back a transaction must consider the status of work done anywhere on behalf of the transaction. The decision must have uniform effect throughout the DTP system.

Even though an RM may have a standard interface, such as Structured Query Language (SQL), it must also address these two items to be useful in the DTP environment.

2.2.3 Application Program

The AP defines transactions and accesses resources within transaction boundaries. Each AP specifies a sequence of operations that involves resources such as terminals and databases. This International Standard generally uses the term AP to refer to a single instance of an application program.

2.2.4 Resource Manager

An RM manages a certain part of the computer's shared resources. Many other software entities can request access to the resource from time to time, using services that the RM provides. Here are some examples of RMs:

- A database management system (DBMS) is an RM. Typical DBMSs are capable of defining transactions and committing work atomically.
- A file access method such as the Indexed Sequential Access Method (ISAM) can be the basis for an RM. Typically, an ISAM RM must be enhanced to support transactions as defined herein.
- A print server might be implemented as an RM.

A single RM may service multiple independent resource domains. An RM *instance* services one of these domains. (See also Section 3.2 on page 13.) Unless specified otherwise, operations this International Standard allows on an RM are allowed on each RM instance.

2.2.5 Global Transactions

Every RM in the DTP environment must support transactions as described in Section 2.2.1 on page 5. Many RMs already structure their work into recoverable units.

In the DTP environment, many RMs may operate in support of the same unit of work. This unit of work is a *global transaction* (which is normally equivalent to an OSI TP transaction when OSI TP is used as the communication protocol; see the referenced OSI TP standards). For example, an AP might request updates to several different databases. Work occurring anywhere in the system must be committed atomically. Each RM must let the TM coordinate the RM's recoverable units of work that are part of a global transaction.

Commitment of an RM's internal work depends not only on whether its own operations can succeed, but also on operations occurring at other RMs, perhaps remotely. If any operation fails anywhere, every participating RM must roll back all operations it did on behalf of the global transaction. A given RM is typically unaware of the work that other RMs are doing. A TM informs each RM of the existence, and directs the completion, of global transactions. An RM is responsible for mapping its recoverable units of work to the global transaction.

2.2.6 Transaction Branches

A global transaction has one or more *transaction branches* (or *branches*). A branch is a part of the work in support of a global transaction for which the TM and the RM engage in a separate but coordinated transaction commitment protocol (see Section 2.3 on page 8). Each of the RM's internal units of work in support of a global transaction is part of exactly one branch. (The term *transaction branch* does not necessarily have the same meaning as in OSI TP. The concept is the same, but the parties may be different; see the referenced OSI TP standards.)

A global transaction might have more than one branch when, for example, the AP uses multiple processes or is involved in the same global transaction by multiple remote APs.

After the TM begins the transaction commitment protocol, the RM receives no additional work to do on that transaction branch. The RM may receive additional work on behalf of the same transaction, from different branches. The different branches are related in that they must be completed atomically.

Each transaction branch identifier (or XID — see Section 4.2 on page 19) that the TM gives the RM identifies both a global transaction and a specific branch. The RM may use this information to optimise its use of shared resources and locks.

2.2.7 Transaction Manager

TMs manage global transactions, coordinate the decision to commit them or roll them back, and coordinate failure recovery. The AP defines the start and end of a global transaction by calling a TM. The TM assigns an identifier to the global transaction (see Section 4.2 on page 19). The TM manages global transactions and informs each RM of the XID on behalf of which the RM is doing work. Although RMs can manage their own recoverable work units as they see fit, each RM must accept XIDs and associate them with those work units. In this way, an RM knows what recoverable work units to complete when the TM completes a global transaction.

2.2.8 Thread of Control

A thread of control (or a *thread*) is the entity, with all its context, that is currently in control of a processor. A thread of control is an operating-system process: an address space and single thread of control that executes within that address space, and its required system resources. The context may include the process' locks on shared resources, and the files the process has open. For portability reasons, the notion of thread of control must be common among the AP, TM, and RM.

The thread concept is central to the TM's coordination of RMs. APs call RMs to request work, while TMs call RMs to delineate transaction branches. The way the RM knows that a given work request pertains to a given branch is that the AP and the TM both call it from *the same thread of control*. For example, an AP thread calls the TM to declare the start of a global transaction. The TM records this fact and informs RMs. After the AP regains control, it uses the native interface of one or more RMs to do work. The RM receives the calls from the AP and TM in the same thread of control.

Certain XA routines, therefore, must be called from a particular thread. The reference manual pages in Chapter 5 indicate which routines require this.

2.2.9 Tightly- and Loosely-coupled Threads

Many application threads of control can participate in a single global transaction. All the work done in these threads is atomically completed. Within a single global transaction, the relationship between any pair of participating threads is either *tightly-coupled* or *loosely-coupled*:

- A tightly-coupled relationship is one where a pair of threads are designed to share resources. In addition, with respect to an RM's isolation policies, the pair are treated as a single entity. Thus, for a pair of tightly-coupled threads, the RM must guarantee that resource deadlock does not occur within the transaction branch.
- A loosely-coupled relationship provides no such guarantee. With respect to an RM's isolation policies, the pair may be treated as if they were in separate global transactions even though the work is atomically completed.

Within a single global transaction, a set of tightly-coupled threads may consist of more than just a pair. Moreover, many sets of tightly-coupled threads may exist within the same global transaction and each set is loosely coupled with respect to the others. The reference manual pages in Chapter 5 indicate how a TM communicates these relationships to an RM.