JTC 1

INTERNATIONAL STANDARD

ISO/IEC 9804

Second edition
1994-12-15

Information technology — Open Systems Interconnection — Service definition for the commitment, concurrency and recovery service element

Technologies de l'information — Interconnexion de systèmes ouverts (OSI) — Définition du service pour l'élément de service d'engagement, concurrence et reprise

ISO IEC

Reference number
ISO/IEC 9804:1994(E)

CONTENTS

ISO/IEC Copyright Office • Case Postale 56 • CH-1211 Genève 20 • Switzerland
Printed in Switzerland

- ITU-T Recommendation X.215 (1993) | ISO/IEC 8326:1994, *Information technology – Open Systems Interconnection – Connection oriented session service definition.*
- ITU-T Recommendation X.216 (1994) | ISO/IEC 8822:1994, *Information technology – Open Systems Interconnection – Connection oriented presentation service definition.*
- ITU-T Recommendation X.217 (1994) | ISO/IEC 8649:1994, *Information technology – Open Systems Interconnection – Service definition for the Association Control Service Element.*
- ITU-T Recommendation X.852 (1993) | ISO/IEC 9805:1994, *Information technology – Open Systems Interconnection – Protocol Specification for the Commitment, Concurrency and Recovery Service Element.*

2.2 Paired Recommendations | International Standards equivalent in technical content

- ITU-T Recommendation X.210 (1993), *Open system interconnection layer service definition conventions.*

  ISO/IEC 10731: ...[1], *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of ISO services.*

3 Definitions

3.1 Reference model definitions

3.1.1 Basic Reference Model definitions

This Service Definition is based on the concepts developed in ITU-T Rec. X.200 | ISO/IEC 7498-1. It makes use of the following terms defined in them:

a) application-entity;

b) Application Layer;

c) application-process;

d) application-service-element;

e) presentation-connection;

f) presentation-service;

g) session-connection;

h) session-service.

3.1.2 Naming and addressing definitions

This Service Definition makes use of the following terms defined in CCITT Rec. X.650 | ISO/IEC 7498-3:

application-entity title.[2]

3.2 Service conventions definitions

This Service Definition makes use of the following terms defined in ITU-T Rec. X.210 | ISO/TR 8509:

a) service-provider;

b) service-user;

c) confirmed service;

d) non-confirmed service;

e) provider-initiated service;

f) primitive;

[1] Presently at the stage of draft.

[2] As defined in CCITT Rec. X.650 | ISO 7498-3, an application-entity title is composed of an application-process title and an application-entity qualifier.

g) request (primitive);

h) indication (primitive);

i) response (primitive); and

j) confirm (primitive).

3.3 Presentation service definitions

This Service Definition makes use of the following terms defined in ITU-T Rec. X.216 | ISO/IEC 8822:

a) abstract syntax;

b) abstract syntax name;

c) defined context set;

d) functional unit [presentation];

e) presentation context; and

f) presentation data value.

3.4 ACSE service definitions

This Service Definition makes use of the following terms defined in ITU-T Rec. X.217 | ISO/IEC 8649:

a) association-initiator;

b) association-responder; and

c) disrupt.

3.5 Application Layer Structure definitions

This Service Definition makes use of the following terms defined in ITU-T Rec. X.207 | ISO/IEC 9545:

a) application-context;

b) application-entity invocation;

c) multiple association control function;

d) single association control function;

e) single association object.

3.6 CCR service definitions

3.6.1 acceptor: The CCR service-user that receives the indication primitive for a particular CCR service. For a confirmed service, it also issues the response primitive.

3.6.2 application failure: The failure of an application-entity invocation to meet its normal specification.

3.6.3 atomic action: A specific set of operations of a distributed application that may be characterized by the properties of atomicity, consistency, isolation, and durability.

3.6.4 atomic action branch; branch: A relationship between two CCR service-users representing an integral part of an atomic action. The relationship may survive both communication or application failure. It is begun by the use of CCR services and later completed by either the use of CCR services or by an application or communication failure.

3.6.5 atomic action branch identifier; branch identifier: A value assigned by the superior that uniquely identifies a branch within the scope of the atomic action.

3.6.6 atomic action data: State and control information about an atomic action and its branches. Atomic action data required for recovery persists if an application or communication failure occurs.

3.6.7 atomic action identifier: A value assigned by the master that uniquely identifies an atomic action within the OSI environment.

3.6.8 atomic action tree: A hierarchical relationship between CCR service-users involved in the operations of an atomic action.

3.6.9 atomicity: A property of a set of related operations such that the operations are either all performed, or none of them are performed.

3.6.10 bound data: Data that are accessed and manipulated by a CCR service-user as part of an atomic action. Its state is bound by the rules of CCR. Bound data survives application and communication failures and exists beyond the atomic action branch.

3.6.11 CCR service-provider: Two peer CCR application-service-elements involved in the same atomic action branch.

3.6.12 CCR service-user: That part of an application-entity invocation that makes use of CCR services to coordinate one or more branches of an atomic action tree.

3.6.13 commitment of an atomic action branch; commitment: Completion of an atomic action branch with the release of bound data in the final state.

3.6.14 communication failure: The unexpected release of the supporting association.

3.6.15 compensating action: Operations used to re-establish either the initial or the final state from a mixed situation that was brought about by a conflict between heuristic decision(s) and the decision of the master.

3.6.16 concurrency control: A real open system mechanism that coordinates modifications to bound data used by concurrent atomic actions so the isolation property of the atomic action is guaranteed.

3.6.17 confirmation of commitment: A statement from a subordinate to the superior that the subordinate has completed local commitment procedures.

3.6.18 consistency: A property of a set of related operations such that the effects of the operations are performed accurately, correctly, and with validity, with respect to application semantics.

3.6.19 cooperating main service: A referencing specification that incorporates the CCR semantics within its own service primitives and carries CCR transfer syntax within its own protocol-data-units.

3.6.20 distributed application: An information processing endeavor that is accomplished using two or more application-entity invocations interconnected within the OSI environment.

NOTE – This term will be removed from this subclause when its definition becomes available in another referenced Recommendation | International Standard.

3.6.21 doubt period: For a CCR service-user (that is not the master), the period during an atomic action that begins when it decides to offer commitment to its superior and ends when it receives either the order to commit or to rollback. The master CCR service-user does not have a doubt period.

3.6.22 durability: A property of a completed set of related operations such that all the effects of the operations are not altered by any sort of failure.

3.6.23 final state: The state of bound data produced as a result of the completed application operations of the atomic action.

3.6.24 heuristic decision: A decision of a CCR service-user that has offered commitment to the superior and then releases all or part of its bound data before it is ordered to commit or to roll back by the superior.

3.6.25 initial state: The state of bound data at the time of first use by an atomic action.

3.6.26 intermediate CCR service-user; intermediate: A CCR service-user that has the role of both subordinate and superior. It is a subordinate of the master CCR service-user or another intermediate CCR service-user. It is the superior of one or more other intermediate and/or leaf CCR service-users.

3.6.27 intermediate state: One of the states of bound data produced during the manipulation of bound data that is neither the initial nor the final state.

3.6.28 interrupted branch: An atomic action branch whose supporting association was released because of an application or communication failure.

3.6.29 isolation: A property of a set of related operations such that partial results of the set of operations are not accessible, except by operations of the set. This definition implies that different sets of related operations that have this property and that share bound data are serializable.

3.6.30 leaf CCR service-user; leaf: A CCR service-user that only has the role of subordinate. It is the subordinate of the master CCR service-user or an intermediate CCR service-user. It has no subordinates of its own.

3.6.31 local commitment procedures: Establishing the final state of all bound data, removal of concurrency controls, and release of all resources used in performing the atomic action.

3.6.32 local rollback procedures: Re-establishing the initial state of all bound data, removal of concurrency controls, and release of all resources used in performing the atomic action.

3.6.33 master CCR service-user; master: A CCR service-user that has the role of superior. As the creator of the atomic action tree, it has no superior, but it is the superior of one or more intermediate and/or leaf CCR service-users.

3.6.34 mixed heuristic situation; mixed situation: The state of bound data produced as the result of heuristic decision(s) when a CCR service-user releases bound data in a state different from the master.

3.6.35 offer of commitment of an atomic action branch; offer of commitment: A statement from the subordinate to the superior that the subordinate is ready for either commitment or rollback.

3.6.36 order of commitment of an atomic action branch; order of commitment: A statement by the superior to the subordinate to initiate commitment of the atomic action branch.

3.6.37 phase I: For a CCR service-user that is not the master, the period during an atomic action that ends when it decides to offer commitment to its superior. For the master CCR service-user, phase I ends when it decides to commit the atomic action. This Recommendation | International Standard does not specify when phase I starts.

3.6.38 phase II: For a CCR service-user that is not the master, the period during an atomic action that begins when it is ordered to commit by its superior. For the master CCR service-user, phase II begins when it decides to commit the atomic action. Phase II ends for any CCR service-user when it completes all of its branches and its involvement with the atomic action ends.

3.6.39 presumed rollback: The recovery mechanism used by CCR. It conditionally allows a CCR service-user to treat an application or communication failure as a rollback. This occurs if it has not recorded atomic action data for the branch. In addition, a CCR service-user acting as a subordinate may presume rollback under the following condition. It has recorded atomic action data for the branch but, during recovery, it discovers that the superior does not.

3.6.40 recovery of an atomic action branch; recovery: Procedures used by a CCR service-user to complete an interrupted atomic action branch for which it has recovery responsibility.

3.6.41 recovery responsibility for an atomic action branch; recovery responsibility: A property of a CCR service-user that determines whether it attempts recovery. The CCR service-user acquires this property as a result of using certain CCR services. It retains the property until the completion of the atomic action branch.

3.6.42 referencing specification: An Application Layer Recommendation | International Standard or other specification that specifies the use of CCR services. CCR services are always used in conjunction with a referencing specification.

3.6.43 requestor: The CCR service-user that issues the request primitive for a particular CCR service. For a confirmed service, it also receives the confirm primitive.

3.6.44 rollback of an atomic action branch; rollback: Completion of an atomic action branch with the release of bound data in the initial state.

3.6.45 subordinate of an atomic action branch; subordinate: The CCR service-user that receives the request to begin the branch, offers commitment and receives the order to commit.

3.6.46 superior of an atomic action branch; superior: The CCR service-user that requests the beginning of the branch, receives the offer of commitment and orders commitment.

4 Abbreviations

This Recommendation | International Standard uses the following abbreviations.

ACSE	Association control service element
AE	Application-entity
AEI	Application-entity invocation
Amd	Amendment to an ISO/IEC International Standard
ASE	Application-service-element

CCR	Commitment, concurrency, and recovery application-service-element
CCR-sp	Commitment, concurrency, and recovery service-provider
cnf	Confirm primitive
ind	Indication primitive
MACF	Multiple association control function
OSI	Open systems interconnection
OSIE	Open systems interconnection environment
req	Request primitive
rsp	Response primitive
SACF	Single association control function
SAO	Single association object
U-ASE	User application-service-element

5 Conventions

This Recommendation | International Standard defines services for CCR following the descriptive conventions defined in ITU-T Rec. X.210 | ISO/IEC 10731.

In clause 7, the definition of each CCR service includes a table that lists the parameters of its primitives. For a given primitive, the presence of each parameter is described by one of the following values:

Blank	Not applicable
C	Conditional
M	Mandatory
U	User option

In these tables, the notation (=) indicates that a parameter value is semantically equal to the value to its left in the table.

6 Concepts

6.1 Use of CCR in a distributed application environment

CCR services are defined for a single association. They are not concerned with and do not address the organization and topology of a distributed application. A referencing specification is always required to coordinate the use of CCR services. However, the use of CCR services requires an understanding of the distributed application environment.

6.1.1 Atomic action environment

6.1.1.1 Atomic action properties

An atomic action is a specific set of related distributed application operations that may be characterized by the following properties:

a) *Atomicity* – A property of a set of related operations such that the operations are either all performed or none of them are performed.

b) *Consistency* – A property of a related set of operations such that the effect of the operations are performed accurately, correctly and with validity, with respect to application semantics.

c) *Isolation* – A property of a set of related operations such that partial results are not accessible, except by operations of the set. This definition implies that different sets of related operations that have this property and that share bound data are serializable.

d) *Durability* – A property of a set of related operations such that all the effects of the operations are not altered by any sort of failure.

In the ideal case, all these atomic action properties are maintained by the CCR service-user. However, the degree of achievement of these properties depends on the level of compliance to the CCR service-user rules (see Annex A) and the local strategies of the CCR service-users.

Taking heuristic decisions is an example of a local strategy that might violate the atomic action properties (see 6.3). Heuristic decisions do not guarantee atomicity of the atomic action. Another example is the use of a concurrency mechanism that allows intermediate states of bound data to be visible outside the atomic action.

6.1.1.2 The atomic action tree

The CCR service-users that participate in an atomic action form a relationship that has a tree structure. For this Recommendation | International Standard, such a relationship is modelled as an atomic action tree, as shown in Figure 1. An atomic action tree consists of CCR service-users and atomic action branches.

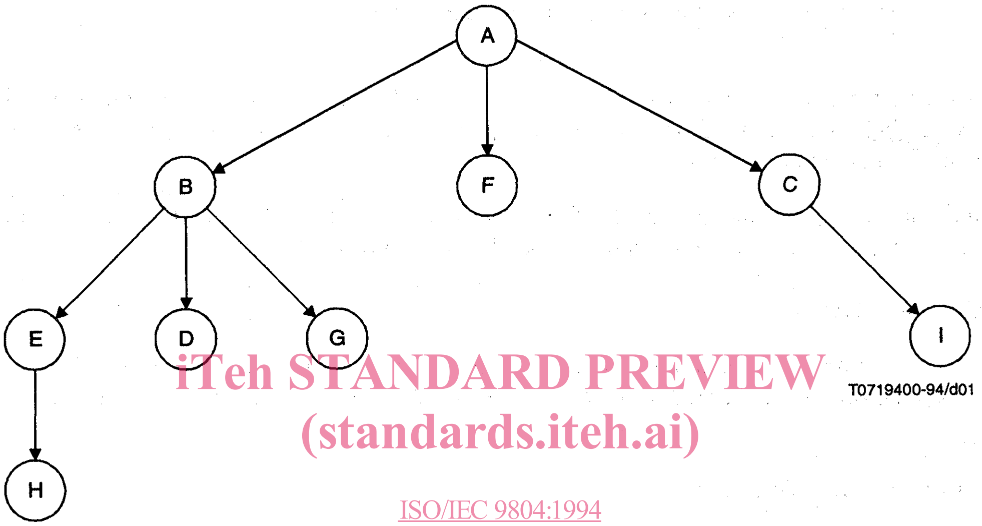

Figure 1 – Atomic action tree

A given AEI can represent one or more CCR service-users of the same or different atomic action trees.

NOTE – Atomic action branches between CCR service-users in the same AEI are outside of the scope of this Recommendation | International Standard.

A **branch** of the atomic action is the relationship between two logically adjacent CCR service-users.

An atomic action tree is dynamically constructed by the formation of its branches. The atomic action tree and its branches only exist for the lifetime of the atomic action.

An atomic action tree starts when a CCR service-user begins the first branch. This CCR service-user assigns this atomic action an atomic action identifier whose value uniquely identifies it within the OSIE. This value is propagated throughout the atomic action. A CCR service-user uses it to maintain concurrency controls. Following an application or communication failure, it is used to correlate recovery for interrupted branches of the atomic action.

Based on the requirements of the referencing specification, a CCR service-user can introduce another CCR service-user into the atomic action tree. This adds a new branch to the atomic action tree.

Beginning from any CCR service-user, an atomic action tree can be ordered hierarchically. Such an ordering that begins with the CCR service-user that started the atomic action defines the atomic action begin-tree. Figure 1 shows an atomic action tree in this ordering started by CCR service-user A. The arrow on each branch shows the direction in which it was started.

Following failure, the recovery facilities of CCR are used to ensure that branch completion procedures are correctly applied throughout the atomic action. An atomic action tree ends with the completion of all the individual branches.

6.1.2 Atomic action branch

An atomic action branch is a relationship between two logically adjacent CCR service-users. This relationship performs a portion of the work of an atomic action. The branch is requested by one of the CCR service-users and the other receives the request. Within the atomic action tree hierarchy, the CCR service-user that receives the request is one level lower than the CCR service-user that begins the branch.

The CCR service-user that begins a branch uses the appropriate atomic action identifier. It assigns a **branch identifier** whose value is unique within the scope of the atomic action. This branch identifier is used to identify a particular branch of the atomic action tree during recovery following an application or communication failure.

A branch is supported by an association. If an application or communication failure occurs, the branch may endure and continue with another association (see 6.2.2.2).

6.1.3 Bound data

The operations of an atomic action involve specific CCR service-user data as determined by the requirements of the referencing specification. For this Service Specification, such data under the control of an atomic action are called bound data.

Modifications made by the operations of the atomic action change the bound data from an initial state to a final state. The modifications are indivisible and either all are applied (placing the bound data in the final state) or none are applied (placing the bound data in the initial state).

During an atomic action, an intermediate state of the bound data is invisible outside of the atomic action. Any modifications are isolated from concurrent operations that take place outside of the atomic action.

6.1.4 Atomic action data

For this Service Specification, the term atomic action data refers to state and control information about an atomic action and its branches. Atomic action data needed for recovery is required to persist if an application or communication failure occurs.

6.1.5 Operation of an atomic action

The overall goal of an atomic action is to exchange application semantics to coordinate the setting of the final state of all bound data. To achieve this, CCR supports a two-phase commitment mechanism. During phase I offers of commitment are collected. During phase II commitment is ordered and confirmed.

Within the atomic action, each CCR service-user may offer commitment on one branch or it may make no offer of commitment. The atomic action may therefore be represented as a hierarchical tree ordered on the basis of offers of commitment – this is the atomic action commit-tree.

6.1.6 Roles in an atomic action

Offers of commitment on a branch are only made from the CCR service-user that received the C-BEGIN indication primitive for the branch.

The following roles for a branch can therefore be distinguished:

a) *Superior* (of the branch) – The CCR service-user that requests the beginning of the branch, receives the offer of commitment and orders commitment.

b) *Subordinate* (of the branch) – The CCR service-user that receives the request to begin the branch, offers commitment and receives the order to commit.

Three types of CCR service-users exist within an atomic action:

a) *Master* – The role of superior. As the creator of the atomic action tree, it has no superior, but it is the superior of one or more intermediate and/or leaf CCR service-users.

b) *Intermediate* – Has the role of both subordinate and superior. It is a subordinate of the master CCR service-user or of another intermediate CCR service-user. It is the superior of one or more other intermediate and/or leaf CCR service-users.

c) *Leaf* – Only has the role of subordinate. It is the subordinate of the master CCR service-user or an intermediate CCR service-user. It has no subordinates of its own.

6.1.7 Two-phase commitment

CCR supports a two-phase commitment mechanism. During phase I offers of commitment are collected. This Recommendation | International Standard does not specify when phase I starts.

A CCR service-user offers commitment to its superior when it has received offers of commitment from all its subordinates and has completed all operations. At this point, it is capable of placing its bound data in either the initial or final state.

The master leaves phase I and enters phase II when it decides to commit the atomic action. To do this, it has received offers of commitment from all its subordinates. It also is capable of placing its bound data in the final state. The master then orders its subordinates to commit. The master leaves phase II after receiving commitment confirmation from all its subordinates to which it has ordered commitment.

A CCR service-user, that is not the master, leaves phase I and enters the doubt period when it decides to offer commitment to its superior. It leaves the doubt period and enters phase II when it receives the order to commit from its superior. An intermediate then orders its subordinates to commit. Finally, it leaves phase II when it sends commitment confirmation to its superior.

6.1.8 Commitment procedure

Commitment is the procedure whereby the CCR service-users participating in an atomic action release their bound data in the final state.

Commitment only occurs after all participating CCR service-users (other than the master) have offered commitment. The master initiates commitment. When the master decides to commit, it enters phase II. As each CCR service-user commits, it releases its bound data in the final state and orders all of its subordinates to commit.

6.1.9 Rollback procedure

Rollback is the procedure used to force the completion of some or all the branches of an atomic action. The procedure results in the release of related bound data in the initial state. Rollback may apply to an entire atomic action. It may also apply to a sub-tree of the atomic action tree whose root is an intermediate or a leaf.

A CCR service-user, that is not the master, may initiate rollback prior to offering commitment. The master may initiate rollback prior to ordering commitment.

For rollback, a CCR service-user releases its bound data in the initial state. It forces the completion of the branches to its subordinates by propagating the rollback on them. If it initiated rollback, it forces the completion of the branch to its superior.

Prior to offering commitment (i.e. before entering the doubt phase), a CCR service-user may order any of its subordinates to roll back even if it does not roll back or release its own bound data. The branches with such subordinates are completed. The CCR service-user remains in the atomic action.

After offering commitment, a CCR service-user that has not taken a heuristic decision only rolls back if it receives an order to roll back from its superior (see 6.3).

6.1.10 Concurrency control

Concurrency control is a real open system mechanism. It coordinates modifications to bound data used by concurrent atomic actions. A concurrency control mechanism guarantees the atomic action isolation property.

NOTE – A concurrency control mechanism ensures that at least one serial sequence of a given set of atomic actions exists that produces the same result to the common bound data as the concurrent (parallel) operation of the same atomic actions on the same bound data. That is, the concurrent execution of atomic actions is serializable.

CCR requires concurrency control for the control of atomic actions. However, the facility to accomplish concurrency is outside the scope of this Recommendation | International Standard.

6.2 CCR facilities

CCR facilities support the beginning and completion of a single branch. The overall goal of a branch is to exchange application semantics to cause the modification of bound data in a coordinated manner.

6.2.1 Operation of a branch

The operation of a branch is divided into two parts:

a) creation of the branch and the exchange of application semantics between the two CCR service-users to produce the final state of the bound data; and

b) commitment whereby the final state of the bound data is made permanent (i.e. committed) or rollback whereby the bound data are restored to the initial state.

At any time before starting the commitment procedure, either CCR service-user may roll back the branch.

A branch can be **interrupted** by an application or communication failure. A CCR service-user with **recovery responsibility** attempts to recover an interrupted branch using another association. A CCR service-user acquires recovery responsibility for a branch before it uses specific CCR services (see 6.2.2.2). Both CCR service-users may have recovery responsibility for the branch.

This Recommendation | International Standard defines CCR services for creating and controlling an individual branch. It also defines rules that govern the exchange of application semantics on a branch.

NOTE – The exchange of application semantics within the framework of a branch is defined by the referencing specification.

6.2.2 Recovery

CCR addresses failure and subsequent recovery at the branch level.

6.2.2.1 Failure

AEIs involved in an atomic action can fail at any time. However, CCR functionality and applicability rely upon the preservation of the bound data and atomic action data over such failures. The loss of such data causes a breakdown of the CCR functionality and applicability and the atomic action properties are no longer guaranteed.

Following an application or communication failure, recovery on another association may be needed. This is done to preserve the atomic action properties and to place the bound data into a consistent state. In particular, the CCR service-user may invoke CCR recovery facilities on another association to recover CCR semantic exchanges that may have been lost.

The CCR service-user accesses atomic action data when it invokes the CCR recovery facilities. Atomic action data and the CCR recovery facilities enable the CCR service-user to complete the branch.

NOTE – Following an application failure, local recovery mechanisms may be needed to restore the CCR service-user. These mechanisms may be used at a later time and may involve human intervention.

6.2.2.2 Recovery mechanism

A recovery mechanism determines when the CCR service-users of a branch acquire recovery responsibility for the branch. If a failure occurs, a CCR service-user with recovery responsibility attempts the recovery of that branch.

CCR employs the presumed rollback (sometimes called "presumed abort") recovery mechanism. For this mechanism, the subordinate acquires recovery responsibility when it decides to offer commitment. The superior acquires recovery responsibility when it decides to order commitment. Both keep recovery responsibility until the completion of the branch.

NOTE – For the master, the presumed rollback recovery mechanism does not require the recording of atomic action data until it decides to commit the atomic action. For a leaf or intermediate, the recording of atomic action data does not occur until it decides to offer commitment. This reduces the overhead of recording atomic action data at the beginning of the branch.

The CCR recovery mechanism for an individual branch makes three basic requirements of the CCR service-user:

a) the maintenance of atomic action data;

b) the ability to set the initial or final state of bound data; and

c) the initiation of recovery when it has recovery responsibility.

The CCR service-user uses atomic action data to determine if it has recovery responsibility.

Before commitment, a CCR service-user does not have recovery responsibility. If an application or communication failure occurs, the CCR service-user shall be capable of restoring its bound data to the initial state.

During the doubt period, a CCR service-user has recovery responsibility. If an application or communication failure occurs, the CCR service-user shall be capable of placing its bound data in either the initial or final state.

After an application failure, local recovery mechanisms re-establish the operation the CCR service-user. The CCR service-user then attempts to use a new association to recover any branch for which it has recovery responsibility.

After a communication failure, the CCR service-user attempts to use another association to recover the branch if it has recovery responsibility.

Recovery responsibility is determined by the atomic action data.

6.3 Heuristic decisions

This Recommendation | International Standard does not explicitly provide capabilities to communicate heuristic decisions, nor the means to reduce the impact of such decisions. This discussion is included because a referencing specification may define conditions concerning heuristic decisions that affect the use of CCR services.

6.3.1 Rationale for heuristic decisions

After a CCR service-user offers commitment, the CCR service-user is in the doubt period. It keeps the capability to commit or to roll back until ordered to do so by the superior. In practice, this may not be acceptable. A prolonged failure may occur or an exceptionally long delay may take place before the decision to commit or roll back is communicated to it.

In such circumstances, a CCR service-user may decide to take a heuristic decision. It puts some or all of its bound data into the initial state, the final state or some intermediate state. It does this while still in the doubt period.

For a heuristic decision, the CCR service-user considers the trade-off between:

a) keeping the capability to commit or to roll back (e.g. keeping locks on valuable data); and

b) taking a heuristic decision that possibly violates the atomic action properties and then coping with the effects of this violation.

6.3.2 Taking a heuristic decision

Any CCR service-user that has offered commitment may take a heuristic decision. This includes a CCR service-user involved in a branch interrupted by an application or communication failure. A CCR service-user may take more than one heuristic decision for a given atomic action.

A referencing specification may specify constraints on the taking of heuristic decisions. This includes not allowing heuristic decisions.

A heuristic decision of a CCR service-user that is different from that taken by the master results in a mixed situation.

Mixed situations are resolved by compensating actions. These compensating actions are application-specific as well as situation-specific. Compensating actions are outside the scope of this Service Specification.

6.3.3 Detection of heuristic mixed situation

The use of CCR services guarantees that any CCR service-user that took a heuristic decision eventually detects whether its decision was in line with the decision of the master or if a mixed situation has occurred.

6.3.4 Reporting of heuristic mixed situation

When a mixed situation is detected, the referencing specification is responsible for reporting to an entity capable of resolving the mixed situation.

A referencing specification may use the User Data parameter of some CCR service primitives to communicate the existence of a heuristic decision or a heuristic mixed situation. Such communication may not be reliable.

7 Service definition

This clause defines each CCR service. Clause 8 describes the allowed sequences of CCR service primitives used on one branch of an atomic action. Annex A specifies CCR service-user rules that a referencing specification shall incorporate.

The services provided by CCR allow two CCR service-users to communicate with each other. The use of the CCR services depends on whether a CCR service-user is acting as the superior or the subordinate.

A superior may invoke the following services:

a) C-BEGIN, to begin a branch;

b) C-PREPARE, to request that the subordinate offer commitment;

c) C-COMMIT, to order the subordinate to commit;

d) C-ROLLBACK, to order the subordinate to roll back; and

e) C-RECOVER, to perform recovery after an application or communication failure.

A subordinate may invoke the following services:

a) C-READY, to offer commitment to the superior;

b) C-ROLLBACK, to inform the superior that rollback has taken place; and

c) C-RECOVER, to perform recovery after an application or communication failure.

After a branch has been created by the C-BEGIN service, the application semantic exchange defined by the referencing specification occurs to progress the branch.

The completion of the branch is achieved either by

a) two-phase commitment, using the C-PREPARE, C-READY, C-COMMIT, and possibly C-RECOVER services; or

b) rollback, using the C-ROLLBACK, and possibly C-RECOVER services.

Table 1 lists the CCR services, the type of service (confirmed, optionally confirmed or non-confirmed), and the requestor of the service.

Table 1 – CCR services

Service	Type	Requestor
C-BEGIN	Optionally confirmed	Superior
C-PREPARE	Non-confirmed	Superior
C-READY	Non-confirmed	Subordinate
C-COMMIT	Confirmed	Superior
C-ROLLBACK	Confirmed	Superior or subordinate
C-RECOVER	Confirmed Optionally confirmed	Superior Subordinate

7.1 C-BEGIN service

7.1.1 Purpose and use

7.1.1.1 A CCR service-user, called the superior, uses the C-BEGIN request primitive to request the beginning of a branch with another CCR service-user, called the subordinate. The C-BEGIN service is used on an established association. The subordinate is included in the same atomic action as the superior.

7.1.1.2 The subordinate may optionally use the C-BEGIN response primitive before sending the first application semantics for this new branch. In this case, application semantics sent before the C-BEGIN response primitive are not part of this new branch.

7.1.1.3 The use of the C-BEGIN service has the effect of establishing a minor synchronization point on the underlying session-connection that supports the branch. The superior shall own the synchronize-minor token.

7.1.1.4 The C-BEGIN service may be jointly issued with the C-COMMIT and C-ROLLBACK services (see 7.4 and 7.5, respectively).

7.1.2 C-BEGIN parameters

Table 2 lists the C-BEGIN service parameters. Each parameter is discussed below.

Table 2 – C-BEGIN parameters

Parameter Name	Req	Ind	Rsp	Cnf
Atomic Action Identifier – Master's Name	M	M(=)		
Atomic Action Identifier – Suffix	M	M(=)		
Branch Identifier – Superior's Name	M	M(=)		
Branch Identifier – Suffix	M	M(=)		
User Data	U	C(=)	U	C(=)

7.1.2.1 Atomic action identifier

7.1.2.1.1 The Atomic Action Identifier unambiguously identifies the atomic action to which this branch belongs. Its value is assigned by the master when the first branch of the atomic action is begun. This value is subsequently used by the superior of each branch.

7.1.2.1.2 The Atomic Action Identifier consists of the Master's Name parameter together with the Suffix parameter.

7.1.2.1.3 The value of the Master's Name parameter is the master's AE title. This value unambiguously identifies the master of the atomic action.

7.1.2.1.4 The master assigns the value of the Suffix parameter so the value unambiguously identifies the atomic action among all those with the same master's name.

7.1.2.2 Branch identifier

7.1.2.2.1 The Branch Identifier unambiguously identifies a branch of an atomic action within the scope of the value of the Atomic Action Identifier. It consists of the Superior's Name parameter together with the Suffix parameter.

7.1.2.2.2 The value of the Superior's Name is the superior's AE title. This value unambiguously identifies the superior of the branch.

NOTE – The A-ASSOCIATE service of ACSE provides a facility to exchange AE title values (see ITU-T Rec. X.217 | ISO/IEC 8649).

7.1.2.2.3 This Service Specification requires the use of either the Calling AE Title or Responding AE Title parameters of the A-ASSOCIATE service to identify the superior.

7.1.2.2.4 The superior assigns the value of the Suffix parameter so the value unambiguously identifies this branch among all those branches of this atomic action with the same superior's name.

7.1.2.3 User data

This parameter may carry an unlimited amount of information as determined by the referencing specification. It may contain one or more presentation data values from presentation contexts in the defined context set when the C-BEGIN request primitive is issued.

NOTE – The referencing specification determines the use of this parameter. For example, it can indicate the minimum requirements for commitment, the preferred character sets for diagnostics, or additional information about the nature of this branch.

7.2 C-PREPARE service

7.2.1 Purpose and use

7.2.1.1 C-PREPARE is a non-confirmed service. The superior may optionally invoke C-PREPARE if it has not received an offer of commitment from the subordinate. The superior requests that the subordinate complete processing for the branch and offer commitment. The superior shall not send the subordinate any further application semantics that change the bound data of this atomic action.

7.2.1.2 The C-PREPARE service is not needed when the application semantic exchange of a branch provides an equivalent prepare request from the superior.