

---

# rfc5805

## Transactions LDAP

Les opérations de modifications LDAP tels que Add, Delete et Modify ont des propriétés ACID (Atomic, Consistency, Isolation, Durability). Chacune des ces modifications agissent sur une seule entrée à la fois. Il est souvent désirable de modifier plusieurs entrées en une seule unité d'interaction, une transaction. Les transactions sont nécessaires pour supporter certaines applications, tel que le provisioning. Cette extension consiste de 2 opérations étendue, un contrôle, et une notification non-sollicitée. l'opération Start Transaction est utilisé pour obtenir un identifiant de transaction.

## Requête et réponse Start Transaction

Une requête Start Transaction est un LDAPMessage de CHOICE **extendedReq** où **requestName** est **1.3.6.1.1.21.1** et **requestValue** est absent.

Une réponse Start Transaction est un LDAPMessage de CHOICE **extendedRes** envoyé en réponse à une requête Start Transaction avec **responseName** absent.

## Contrôle Transaction Specification

C'est un LDAPControl où **controlType** est **1.3.6.1.1.21.2**, la criticité à TRUE, et **controlValue** est un ID de transaction. Ce contrôle est approprié pour les requêtes de modification tels que Add, Delete, Modify, ModifyDN et Password Modify.

## Requête et réponse End Transaction

Une requête End Transaction est un LDAPMessage de CHOICE **extendedReq** où **requestName** est **1.3.6.1.1.21.3** et **requestValue** contient un **txnEndReq** encodé BER :

```
txnEndReq ::= SEQUENCE {
    commit BOOLEAN DEFAULT TRUE,
    identifiant OCTET STRING }
```

**commit** à TRUE, indique une requête pour envoyer la transaction identifiée par **identifiant**. à FALSE indique une requête d'abandon de la transaction identifiée.

Une réponse End Transaction est un LDAPMessage envoyé en réponse à un requête End Transaction. Le nom de réponse est absent. **responseValue** si présent, contient un **txEndRes** encodé BER :

```
txEndRes ::= SEQUENCE {
    messageID MessageID OPTIONAL,
    -- msgid associated with non-success resultCode
    updateControls SEQUENCE OF updateControls SEQUENCE {
        messageID MessageID,
        -- msgid associated with controls
        controls Controls
    } OPTIONAL }
```

---

}

MessageID fournit l'id de message de la requête de mise à jours associée avec une réponse autre que success. messageID est absent quand un resultCode est success. updatesControls fournit une facilité pour retourner les contrôles de réponse qui normalement (en l'absence de transactions) seraient retournés dans une réponse de mise à jours.

## Transaction annulée

Le Aborted Transaction Notice est un message de notification non-sollicité où responseName est 1.3.6.1.1.21.4 et responseValue est présent et contient un id de transaction.

## Découverte d'extension

Les serveurs implémentant cette spécification devraient publier **1.3.6.1.1.21.1** et **1.3.6.1.1.21.3** dans **supportedExtension** et **1.3.6.1.1.21.2** dans **supportedControl**

## Démarrer une transaction

Un client qui souhaite effectuer une séquence de modifications dans l'annuaire envoie une requête Start Transaction. Le serveur envoie une réponse Start Transaction avec un ID de transaction et un resultCode success.

## Spécifications d'une transaction

Le client peut fournir une ou plusieurs requêtes de modifications, chacune avec un contrôle Transaction Specification contenant l'ID de transaction indiquant les modifications à traiter. Chacune de ces requêtes doivent avoir une valeur messageID différente. Si le serveur ne peut pas traiter l'opération demandée, il retourne immédiatement le resultCode adéquat, sinon il retourne success. Si le serveur ne peut plus continuer la transaction, il fournit un Aborted Transaction Notice avec un resultCode adéquat.

## Règlement de transaction

Un client demande le règlement de la transaction en fournissant une requête End Transaction, indiquant son désir que la transaction soit traitée ou annulée.

Une fois reçu une annulation de transaction, le serveur annule l'ID de transaction et abandonne toutes les opérations de la transaction.

Une fois reçu une fin de transaction, le serveur traite toutes les opérations de la transaction. Le serveur retourne une réponse End Transaction avec resultCode à success et sans responseValue pour indiquer que toutes les modifications ont été appliquées. Sinon, le serveur retourne un resultCode adéquat. Si l'échec est associé avec un requête particulière, le txnEndRes.messageID dans responseValue est l'id de message de cette demande de modification. Si l'échec n'est pas associé à une requête de modification, aucun messageID n'est retourné.

---

# Divers problèmes

Les transactions ne peuvent pas être imbriquées. Chaque transaction doit être initiée, spécifiée et terminée avec un contexte de sécurité stable.

## Interaction avec d'autres extensions

**Contrôle d'assertion** Le contrôle Assertion est approprié pour les requêtes de modification dans une transaction. Il n'est pas approprié avec les opérations Start Transaction et End Transaction.

**Contrôle ManageDsaIT** Le contrôle ManageDsaIT est approprié pour les requêtes de modification dans une transaction. Il n'est pas approprié avec les opérations Start Transaction et End Transaction.

**Contrôle d'autorisation proxifié** Ce contrôle est approprié avec l'opération étendue Start Transaction, mais pas End Transaction.

**Contrôle Read Entry** Les contrôles Pre- et Post-Read sont appropriés avec des requêtes de modification dans une transaction. L'entrée est retournée dans `txnEndRes.updateControls` et `responseValue` d'une réponse End Transaction.