Groupe de travail Réseau
Request for Comments : 3744
Catégorie : En cours de normalisation

Traduction Claude Brière de L'Isle

G. Clemm, IBM
J. Reschke, greenbytes
E. Sedlar, Oracle Corporation
J. Whitehead, U.C. Santa Cruz
mai 2004

Protocole de contrôle d'accès à la collecte des noms d'auteurs et de version répartie sur la Toile (WebDAV)

Statut de ce mémoire
Le présent document spécifie un protocole Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et des suggestions pour son amélioration. Prière de se reporter à l'édition actuelle du STD 1 "Normes des protocoles officiels de l'Internet" pour connaître l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé
Le présent document spécifie un ensemble de méthodes, d'en-têtes, de corps de messages, de propriétés, et de rapports qui définit les extensions de contrôle d'accès au protocole WebDAV de collecte de noms d'auteurs et de version répartie. Ce protocole permet à un client de lire et modifier les listes de contrôle d'accès qui donnent à un serveur les instructions pour permettre ou refuser les opérations sur une ressource (comme des invocations de méthode du protocole de transfert hypertexte (HTTP, *HyperText Transfer Protocol*)) par un certain principal. Une représentation légère des principaux comme ressources de la Toile prend en charge l'intégration d'une large gamme de répertoires de gestion d'utilisateur. Les opérations de recherche permettent la découverte et la manipulation des principaux en utilisant des noms de personnes.

Table des Matières

1. Introduction

Le but des extensions de contrôle d'accès à WebDAV est de fournir un mécanisme interopérable pour traiter le contrôle d'accès discrétionnaire des contenus et des métadonnées gérées par les serveurs WebDAV. Le contrôle d'accès WebDAV peut être mis en œuvre sur des répertoires de contenu avec une sécurité aussi simple que celle des fichiers système UNIX, ainsi qu'avec des modèles plus sophistiqués. Le principe sous jacent du contrôle d'accès est que qui vous êtes détermine quelles opérations vous pouvez effectuer sur une ressource. Le "qui vous êtes" est défini par un identifiant "principal" ; l'utilisateur, le logiciel client, les serveurs, et leurs groupes ont des identifiants de principal. Les "opérations que vous pouvez effectuer" sont déterminées par une seule "liste de contrôle d'accès (ACL, *Access Control List*) associée à une ressource. Une ACL contient un ensemble d'éléments de contrôle d'accès (ACE, *Access Control Element*) où chaque ACE spécifie un principal et un ensemble de privilèges qui sont accordés ou refusés à ce principal. Lorsque un principal soumet une opération (comme une méthode HTTP ou WebDAV) à une ressource pour exécution, le serveur évalue les ACE dans l'ACL pour déterminer si le principal a la permission pour cette opération.

Comme chaque ACE contient l'identifiant d'un principal, le logiciel client géré par une personne doit fournir un mécanisme pour choisir ce principal. La présente spécification utilise les URL de schéma http(s) pour identifier les principaux, qui sont représentés comme des ressources à capacité WebDAV. Il n'est pas garanti que les URL qui identifient les principaux vont avoir une signification pour les personnes. Par exemple, http://www.example.com/u/256432 et http://www.example.com/people/Greg.Stein sont tous deux des URL valides qui pourraient être utilisés pour identifier le même principal. Pour remédier à cela, chaque ressource principale a la propriété DAV:displayname qui contient un nom lisible par l'homme pour le principal.

Comme un principal peut être identifié par plusieurs URL, cela soulève le problème de la détermination exacte du principal référencé dans un ACE donné. Il est impossible à un client de déterminer qu'un ACE qui accorde le privilège de lecture http://www.example.com/people/Greg.Stein affecte aussi le principal à http://www.example.com/u/256432. C'est-à-dire qu'un

client n'a pas de mécanisme pour déterminer que deux URL identifient la même ressource principale. Par suite, la présente spécification exige que les clients utilisent juste un des nombreux URL possibles pour un principal lors de la création des ACE. Un client peut découvrir quel URL utiliser en restituant la propriété DAV:principal-URL (paragraphe 4.2) d'une ressource principale. Peu importe quel URL du principal est utilisé avec PROPFIND, la propriété retourne toujours le même URL.

Avec un système qui a entre des centaines et des milliers de principaux, le problème se pose de comment permettre à un opérateur humain d'un logiciel de client de choisir juste un de ces principaux. Une approche est d'utiliser des hiérarchies de collection larges pour étaler les principaux sur un grand nombre de collections, donnant peu de principaux par collection. Un exemple est celui d'une hiérarchie à deux niveaux dont le premier contient 36 collections (a-z, 0-9), et le second en contient 36 autres, créant des collections /a/a/, /a/b/, ..., /a/z/, de telle sorte qu'un principal avec le nom propre "Stein" va apparaître à /s/t/Stein. En effet, cela pré-calcule une interrogation courante de recherche sur le nom propre, et le code dans une hiérarchie. L'inconvénient de ce schéma est qu'il traite seulement un petit ensemble d'interrogations prédéfinies, et faire l'exercice à travers toute la hiérarchie des collections ajoute des étapes inutiles (naviguer de bas en haut et de haut en bas) alors que l'utilisateur connaît déjà le nom du principal. Bien qu'organiser les URL de principaux en une hiérarchie soit une organisation valide de l'espace de noms, les utilisateurs ne devraient pas être forcés de naviguer dans cette hiérarchie pour choisir un principal.

La présente spécification procure la capacité d'effectuer des recherches de sous chaînes sur un petit ensemble de propriétés sur les ressources représentant des principaux. Cela permet des recherches sur la base du nom propre, du prénom, du nom d'utilisateur, du titre de la fonction, etc. Deux recherches séparées sont supportées, toutes deux via la méthode REPORT, une pour rechercher les ressources principales (DAV:principal-property-search, paragraphe 9.4), et l'autre pour déterminer quelles propriétés peuvent faire l'objet d'une recherche (DAV:principal-search-property-set, paragraphe 9.5).

Une fois qu'un principal a été identifié dans un ACE, un serveur qui évalue cet ACE doit connaître l'identité du principal qui fait une demande de protocole, et doit valider que ce principal est qui il prétend être, processus connu comme l'authentification. La présente spécification omet intentionnellement la discussion de l'authentification, car le protocole HTTP a déjà un certain nombre de mécanismes d'authentification [RFC2617]. Certains mécanismes d'authentification (comme l'authentification HTTP par résumé, que toutes les mises en œuvre conformes à WebDAV sont obligées de prendre en charge) doivent être disponibles pour valider l'identité d'un principal.

Les questions suivantes ne sont pas abordées par le présent document :
- le contrôle d'accès qui s'applique seulement à une propriété particulière d'une ressource (sauf les propriétés de contrôle d'accès DAV:acl et DAV:current-user-privilege-set) plutôt qu'à la ressource entière,
- la sécurité fondée sur le rôle (où un rôle peut être vu comme un groupe de principaux défini de façon dynamique),
- la spécification de la façon d'initialiser une ACL sur une ressource,
- la spécification d'une ACL qui s'applique globalement à toutes les ressources, plutôt qu'à une ressource particulière.
- la création et la maintenance de ressources représentant des personnes ou des agents de calcul (principaux), et des groupes de personnes ou agents de calcul.

La présente spécification est organisée comme suit. Le paragraphe 1.1 définit les concepts clés utilisés dans la spécification, et est suivi par une discussion plus en profondeur des principaux (Section 2), et des privilèges (Section 3). Les propriétés définies sur les principaux sont spécifiées à la Section 4, et les propriétés de contrôle d'accès sur les ressources de contenu sont spécifiées à la Section 5. Les façons d'évaluer les ACL sont décrites à la Section 6. La découverte par le client des capacités de contrôle d'accès avec OPTIONS est décrite au paragraphe 7.2. Les interactions entre la fonction de contrôle d'accès et les méthodes HTTP et WebDAV existantes sont décrites dans le reste de la Section 7. La méthode d'établissement du contrôle d'accès, ACL, est spécifiée à la Section 8. Quatre rapports qui fournissent des capacités de recherche limitées du côté du serveur sont décrites à la Section 9. Les sections sur le traitement XML (Section 10), les considérations d'internationalisation (Section 11), de sécurité (Section 12), et d'authentification (Section 13) terminent la spécification. Un appendice (Appendice A) donne une définition de type de document (DTD, *Document Type Definition*) XML pour les éléments XML définis dans la spécification.

1.1 Termes

Le présent document utilise les termes définis dans HTTP [RFC2616] et WebDAV [RFC2518]. De plus, on définit les termes suivants :

principal : Un "principal" est une personne ou un acteur de calcul distinct qui initie l'accès à des ressources du réseau. Dans le présent protocole, un principal est une ressource HTTP qui représente un tel acteur.

groupe : Un "groupe" est un principal qui représente un ensemble d'autres principaux.

privilège : Un "privilège" contrôle l'accès à un ensemble particulier d'opérations HTTP sur une ressource.

privilège agrégé : Un "privilège agrégé" est un privilège qui contient un ensemble d'autres privilèges.

privilège abstrait : Le modificateur "abstrait", lorsque appliqué à un privilège sur une ressource, signifie que le privilège ne peut pas être établi dans un élément de contrôle d'accès (ACE, *Access Control Element*) sur cette ressource.

liste de contrôle d'accès (ACL) : Une "ACL" est une liste d'éléments de contrôle d'accès qui définit le contrôle d'accès à une certaine ressource.

élément de contrôle d'accès (ACE) : Un "ACE" accorde ou refuse un certain ensemble de privilèges (non abstraits) pour un certain principal.

ACE hérité : Un "ACE hérité" est un ACE qui est partagé dynamiquement à partir de l'ACL d'une autre ressource. Lorsque un ACE partagé change sur la ressource principale, il est aussi changé sur les ressources héritières.

propriété protégée : Une "propriété protégée" est celle dont la valeur ne peut pas être mise à jour sauf par une méthode explicitement définie comme mettant à jour cette propriété spécifique. En particulier, une propriété protégée ne peut pas être mise à jour avec une demande PROPPATCH.

1.2 Conventions de notation

Le BNF augmenté utilisé par le présent document pour décrire les éléments de protocole est décrit au paragraphe 2.1 de la [RFC2616]. Parce que ce BNF augmenté utilise les règles de production de base fournies au paragraphe 2.2 de la [RFC2616], ces règles s'appliquent aussi au présent document.

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" dans le présent document sont à interpréter comme décrit dans la [RFC2119].

Dans le présent document, les définitions d'éléments XML utilisent les déclarations de type d'élément XML (comme on les trouve dans les déclarations de type de document XML), décrites au paragraphe 3.2 de [REC-XML]. Lorsque un type d'élément XML dans l'espace de noms "DAV:" est référencé dans le présent document en dehors du contexte d'un fragment XML, la chaîne "DAV:" sera en préfixe au nom de l'élément.

2. Principaux

Un principal est une ressource réseau qui représente une personne ou acteur de calcul distinct qui initie l'accès aux ressources du réseau. Les utilisateurs et groupes sont représentés comme des principaux dans de nombreuses mises en œuvre ; d'autres types de principaux sont aussi possibles. Un URI de tout schéma PEUT être utilisé pour identifier une ressource principale. Cependant, les serveurs qui mettent en œuvre la présente spécification DOIVENT exposer les ressources principales à un URL http(s), qui est un schéma privilégié pointant sur des ressources qui ont des propriétés supplémentaires, comme décrit à la Section 4. Ainsi, une ressource principale peut avoir plusieurs URI, dont l'un doit être un URL de schéma http(s). Bien qu'une mise en œuvre DEVRAIT prendre en charge PROPFIND et PUISSE prendre en charge PROPPATCH pour accéder et modifier les informations sur un principal, elle n'est pas obligée de le faire.

Une ressource principale peut être un groupe, où un groupe est un principal qui représente en ensemble d'autres principaux, appelés les membres du groupe. Si une personne ou un agent de calcul correspond à une ressource principale qui est un membre d'un groupe, elle correspond aussi au groupe. l'appartenance à un groupe est récurrente, de sorte que si un principal est membre d'un groupe GRPA, et si GRPA est membre du groupe GRPB, le principal est alors aussi membre de GRPB.

3. Privilèges

La capacité à effectuer une certaine méthode sur une ressource DOIT être contrôlée par un ou plusieurs privilèges. Les auteurs d'extensions de protocole qui définissent de nouvelles méthodes HTTP DEVRAIENT spécifier quels privilèges (en définissant de nouveaux privilèges, ou en transposant sur ceux ci-dessous) sont exigés pour effectuer la méthode. Un principal sans privilège pour une ressource DOIT se voir refuser tout accès HTTP à cette ressource, sauf si le principal correspond à un ACE construit en utilisant les pseudo principaux DAV:all, DAV:authenticated, ou DAV:unauthenticated (voir au paragraphe 5.5.1). Les serveurs DOIVENT rapporter une erreur 403 "Interdit" si l'accès est refusé, sauf dans le cas où le privilège interdit la

capacité de savoir si la ressource existe, auquel cas 404 "Pas trouvé" peut être retourné.

Les privilèges peuvent être des conteneurs d'autres privilèges, auquel cas ils sont appelés des "privilèges agrégés". Si un privilège agrégé est accordé ou refusé à un principal, cela est sémantiquement équivalent à accorder ou refuser individuellement chacun des privilèges agrégés. Par exemple, une mise en œuvre peut définir des privilèges "add-member" et "remove-member" qui contrôlent la capacité d'ajouter et supprimer un membre d'un groupe. Comme ces privilèges contrôlent la capacité de mettre à jour l'état d'un groupe, ces privilèges seraient agrégés par le privilège DAV:write sur un groupe, et accorder le privilège DAV:write sur un groupe accorderait aussi les privilèges "add-member" et "remove-member".

Des privilèges peuvent être déclarés "abstraits" pour une certaine ressource, auquel cas ils ne peuvent pas être établis sur un ACE sur cette ressource. Les privilèges agrégés et non agrégés sont tous deux capables d'être abstraits. Les privilèges abstraits sont utiles pour modéliser des privilèges qui autrement ne seraient pas exposés via le protocole. Les privilèges abstraits fournissent aussi aux mises en œuvre de serveur une souplesse dans l'application de privilèges définis dans la présente spécification. Par exemple, si un serveur est incapable de séparer la capacité de lire une ressource de la capacité de lire une ACL, il peut quand même modéliser les privilèges DAV:read et DAV:read-acl définis dans la présente spécification en les déclarant abstraits, et en les contenant au sein d'un privilège agrégé non abstrait (disons, read-all) qui contient DAV:read, et DAV:read-acl. De cette façon, il est possible d'établir le privilège agrégé, read-all, couplant ainsi l'établissement de DAV:read et de DAV:read-acl, mais il n'est pas possible d'établir DAV:read, ou DAV:read-acl individuellement. Comme les privilèges agrégés peuvent être abstraits, il est aussi possible d'utiliser des privilèges abstraits pour grouper ou organiser des privilèges non abstraits. Les boucles de privilèges ne sont pas permises ; donc, un privilège NE DOIT PAS se contenir lui-même. Par exemple, DAV:read ne peut pas contenir DAV:read.

L'ensemble des privilèges qui s'appliquent à une ressource particulière peut varier avec le DAV:resourcetype de la ressource, ainsi qu'entre des mises en œuvre de serveur différentes. Cependant, pour promouvoir l'interopérabilité, la présente spécification définit un ensemble de privilèges bien connus (par exemple, DAV:read, DAV:write, DAV:read-acl, DAV:write-acl, DAV:read-current-user-privilege-set, et DAV:all) qui peut au moins être utilisé pour classer les autres privilèges définis sur une ressource particulière. Les permissions d'accès sur les ressources nulles (définies dans la [RFC2518], Section 3) sont seulement celles dont elles héritent (si il en est) et elles ne sont pas découvrables (c'est-à-dire, les propriétés de contrôle d'accès spécifiées à la Section 5 ne sont pas définies sur les ressources nulles). À la transition de ressource nulle à ressource à état plein, la liste initiale de contrôle d'accès est établie par la politique de valeur d'ACL par défaut du serveur (si il en est une).

Les mises en œuvre de serveur PEUVENT définir de nouveaux privilèges au delà de ceux définis dans la présente spécification. Les privilèges définis par des mises en œuvre individuelles NE DOIVENT PAS utiliser l'espace de noms DAV:, et devraient plutôt utiliser un espace de noms qu'elles contrôlent, comme un URL de schéma http.

3.1 Privilège DAV:read

Le privilège read *(lecture)* contrôle des méthodes qui retournent des informations sur l'état de la ressource, incluant les propriétés de la ressource. Les méthodes affectées incluent GET et PROPFIND. Tout privilège défini par la mise en œuvre qui contrôle aussi l'accès à GET et PROPFIND doit être agrégé sous DAV:read – si une ACL accorde l'accès à DAV:read, le client peut s'attendre à ce qu'aucun autre privilège n'ait besoin d'être accordé pour avoir accès à GET et PROPFIND. De plus, le privilège read DOIT contrôler la méthode OPTIONS.

<!ELEMENT read EMPTY>

3.2 Privilège DAV:write

Le privilège write *(écriture)* contrôle des méthodes qui verrouillent une ressource ou modifient le contenu, les propriétés mortes, ou (dans le cas d'une collection) les membres d'une ressource, comme PUT et PROPPATCH. Noter qu'une modification d'état est aussi contrôlée via le verrouillage (voir le paragraphe 5.3 de la [RFC2518]), de sorte qu'un accès effectif en écriture exige que à la fois les privilèges d'écriture et les exigences de verrouillage d'écriture soient satisfaits. Tout privilège défini par la mise en œuvre qui contrôle aussi l'accès aux méthodes qui modifient le contenu, les propriétés mortes ou les membres de collection, doit être agrégé sous DAV:write, par exemple, si une ACL accorde l'accès à DAV:write, le client peut s'attendre à ce qu'aucun autre privilège n'ait besoin d'être accordé pour avoir accès à PUT et PROPPATCH.

<!ELEMENT write EMPTY>

3.3 Privilège DAV:write-properties

Le privilège DAV:write-properties contrôle des méthodes qui modifient les propriétés mortes de la ressource, comme PROPPATCH. C'est la mise en œuvre qui détermine si ce privilège peut être utilisé pour contrôler l'accès à des propriétés vives. Tout privilège défini par la mise en œuvre qui contrôle aussi l'accès à des méthodes qui modifient les propriétés mortes

doit être agrégé sous DAV:write-properties - par exemple, si une ACL accorde l'accès à DAV:write-properties, le client peut en toute sécurité s'attendre à ce qu'aucun autre privilège n'ait besoin d'être accordé pour avoir accès à PROPPATCH.

<!ELEMENT write-properties EMPTY>

3.4 Privilège DAV:write-content

Le privilège DAV:write-content contrôle les méthodes qui modifient le contenu d'une ressource existante, comme PUT. Tout privilège défini par la mise en œuvre qui contrôle aussi l'accès au contenu doit être agrégé sous DAV:write-content - par exemple, si une ACL accorde l'accès à DAV:write-content, le client peut en toute sécurité s'attendre à ce qu'aucun autre privilège n'ait besoin d'être accordé pour avoir accès à PUT. Noter que PUT – lorsque appliqué à un URI non transposé – crée une nouvelle ressource et est donc contrôlé par le privilège DAV:bind sur la collection parente.

<!ELEMENT write-content EMPTY>

3.5 Privilège DAV:unlock

Le privilège DAV:unlock contrôle l'utilisation de la méthode UNLOCK par un principal autre que le propriétaire du verrou (le principal qui a créé un verrou peut toujours effectuer un UNLOCK). Bien que l'ensemble des utilisateurs qui peuvent verrouiller une ressource soit habituellement le même que celui qui peut modifier une ressource, les serveurs peuvent permettre à diverses sortes d'administrateurs de déverrouiller des ressources verrouillées par d'autres. Tout privilège qui contrôle l'accès par des propriétaires non verrouilleurs pour UNLOCK DOIT être agrégé sous DAV:unlock.

Un propriétaire de verrou peut toujours supprimer un verrou en produisant un UNLOCK avec le jeton de verrou correct et des accréditifs d'authentification. C'est-à-dire que, même si un principal n'a pas de privilège DAV:unlock, il peut quand même supprimer les verrous qui lui appartiennent. Les principaux autres que le propriétaire du verrou ne peuvent supprimer un verrou que si ils ont le privilège DAV:unlock et si ils produisent un UNLOCK avec le jeton de verrou correct. Le temporisateur de verrou n'est pas affecté par le privilège DAV:unlock.

<!ELEMENT unlock EMPTY>

3.6 Privilège DAV:read-acl

Le privilège DAV:read-acl contrôle l'utilisation de PROPFIND pour restituer la propriété DAV:acl à la ressource.

<!ELEMENT read-acl EMPTY>

3.7 Privilège DAV:read-current-user-privilege-set

Le privilège DAV:read-current-user-privilege-set contrôle l'utilisation de PROPFIND pour restituer la propriété DAV:current-user-privilege-set à la ressource.

Les clients sont supposés utiliser cette propriété pour indiquer visuellement dans leur éléments d'URI qu'ils dépendent des permissions d'une ressource, par exemple, en mettant en gris les ressources qui ne sont pas accessible en écriture.

Ce privilège est séparé de DAV:read-acl parce qu'il y a un besoin de permettre à la plupart des utilisateurs l'accès aux privilèges permis aux utilisateurs courants (du fait de son utilisation pour créer l'URI) alors que l'ACL complète contient les informations qui peuvent n'être pas appropriées pour l'utilisateur authentifié actuel. Par suite, l'ensemble des utilisateurs qui peuvent voir l'ACL complète est supposé être beaucoup plus petit que celui de ceux qui peuvent lire l'ensemble de privilèges d'utilisateur courant, et donc, des privilèges distincts sont nécessaires pour chacun.

<!ELEMENT read-current-user-privilege-set EMPTY>

3.8 Privilège DAV:write-acl

Le privilège DAV:write-acl contrôle l'utilisation de la méthode ACL pour modifier la propriété DAV:acl de la ressource.

<!ELEMENT write-acl EMPTY>

3.9 Privilège DAV:bind

Le privilège DAV:bind permet une méthode pour ajouter un nouvel URL membre à la collection spécifiée (par exemple via PUT ou MKCOL). Il est ignoré pour les ressources qui ne sont pas des collections.

<!ELEMENT bind EMPTY>

3.10 Privilège DAV:unbind

Le privilège DAV:unbind permet une méthode pour supprimer un URL membre de la collection spécifiée (par exemple via DELETE ou MOVE). Il est ignoré pour les ressources qui ne sont pas des collections.

<!ELEMENT unbind EMPTY>

3.11 Privilège DAV:all

DAV:all est un privilège agrégé qui contient l'ensemble entier de privilèges qui peuvent être appliqués à la ressource.

<!ELEMENT all EMPTY>

3.12 Agrégation de privilèges prédéfinis

Les mises en œuvre de serveur ont toute liberté pour agréger les privilèges prédéfinis (définis aux paragraphes 3.1 à 3.10) sous réserve des limitations suivantes :

DAV:read-acl NE DOIT PAS contenir DAV:read, DAV:write, DAV:write-acl, DAV:write-properties, DAV:write-content, ou DAV:read-current-user- privilege-set.
DAV:write-acl NE DOIT PAS contenir DAV:write, DAV:read, DAV:read-acl, ou DAV:read-current-user-privilege-set.
DAV:read-current-user-privilege-set NE DOIT PAS contenir DAV:write, DAV:read, DAV:read-acl, ou DAV:write-acl.
DAV:write NE DOIT PAS contenir DAV:read, DAV:read-acl, ou DAV:read-current-user-privilege-set.
DAV:read NE DOIT PAS contenir DAV:write, DAV:write-acl, DAV:write- properties, ou DAV:write-content.
DAV:write DOIT contenir DAV:bind, DAV:unbind, DAV:write-properties et DAV:write-content.

4. Propriétés de principaux

Les principaux se manifestent aux clients comme des ressources WebDAV, identifiées par un URL. Un principal DOIT avoir une propriété DAV:displayname non vide (définie au paragraphe 13.2 de la [RFC2518]), et une propriété DAV:resourcetype (définie au paragraphe 13.9 de la [RFC2518]). De plus, un principal DOIT rapporter l'élément XML DAV:principal dans la valeur de la propriété DAV:resourcetype. La déclaration de type d'élément pour DAV:principal est :

<!ELEMENT principal EMPTY>

Le présent protocole définit les propriétés supplémentaires suivantes pour un principal. Comme il peut être coûteux pour un serveur de restituer les informations de contrôle d'accès, le nom et la valeur de ces propriétés NE DEVRAIENT PAS être retournés par une demande allprop PROPFIND (comme défini au paragraphe 12.14.1 de la [RFC2518]).

4.1 DAV:alternate-URI-set

Cette propriété protégée, si elle n'est pas vide, contient les URI des ressources réseau avec des informations descriptives supplémentaires sur le principal. Cette propriété identifie des ressources réseau supplémentaires (c'est-à-dire, elle contient un ou plusieurs URI) qui peuvent être consultées par un client pour avoir des connaissances supplémentaires sur un principal. Une utilisation attendue de cette propriété est la mémorisation d'un URL de schéma LDAP [RFC2255]. Un agent d'utilisateur qui rencontre un URL LDAP pourrait utiliser LDAP [RFC2251] pour restituer des informations supplémentaires de répertoire lisibles par la machine sur le principal, et afficher ces informations dans son interface d'utilisateur. La prise en charge de cette propriété est EXIGÉE, et la valeur est vide si il n'existe pas d'URI de remplacement pour le principal.

<!ELEMENT alternate-URI-set (href*)>

4.2 DAV:principal-URL

Un principal peut avoir de nombreux URL, mais il doit y avoir un "URL principal" que les clients peuvent utiliser pour

identifier un principal de façon univoque. Cette propriété protégée contient l'URL qui DOIT être utilisé pour identifier ce principal dans une demande ACL. La prise en charge de cette propriété est EXIGÉE.

<!ELEMENT principal-URL (href)>

4.3 DAV:group-member-set

Cette propriété d'un groupe principal identifie les principaux qui sont des membres directs de ce groupe. Comme un groupe peut être membre d'un autre groupe, un groupe peut aussi avoir des membres indirects (c'est-à-dire, les membres de ses membres directs). Un URL dans le DAV:group-member-set pour un principal DOIT être le DAV:principal-URL de ce principal.

<!ELEMENT group-member-set (href*)>

4.4 DAV:group-membership

Cette propriété protégée identifie les groupes dans lesquels le principal est directement un membre. Noter qu'un serveur peut permettre qu'un groupe soit un membre d'un autre groupe, auquel cas le DAV:group-membership de ces autres groupes devrait être interrogé afin de déterminer les groupes dans lesquels le principal est indirectement membre. La prise en charge de cette propriété est EXIGÉE.

<!ELEMENT group-membership (href*)>

5. Propriétés de contrôle d'accès

La présente spécification définit un certain nombre de nouvelles propriétés pour les ressources WebDAV. Les propriétés de contrôle d'accès peuvent être restituées tout comme les autres propriétés WebDAV, en utilisant la méthode PROPFIND. Comme il est coûteux pour de nombreux serveurs, de restituer les informations de contrôle d'accès, une demande allprop PROPFIND (comme défini au paragraphe 12.14.1 de la [RFC2518]) NE DEVRAIT PAS retourner les noms et valeurs des propriétés définies dans cette section.

Les propriétés de contrôle d'accès (en particulier DAV:acl et DAV:inherited-acl-set) sont définies sur la ressource identifiée par l'URI de demande d'une demande PROPFIND. Une conséquence directe en est que si la ressource est accessible via plusieurs URI, la valeur des propriétés de contrôle d'accès est la même à travers ces URI.

Les ressources HTTP qui prennent en charge le protocole de contrôle d'accès WebDAV DOIVENT contenir les propriétés suivantes. Les ressources nulles (décrites à la Section 3 de la [RFC2518]) NE DOIVENT PAS contenir les propriétés suivantes.

5.1 DAV:owner

Cette propriété identifie un certain principal comme étant le "propriétaire" de la ressource. Comme le propriétaire d'une ressource a souvent des capacités de contrôle d'accès spéciales (par exemple, le propriétaire a fréquemment un privilège permanent DAV:write-acl) les clients peuvent afficher le propriétaire de ressource dans leur interface d'utilisateur.

Les serveurs PEUVENT mettre en œuvre DAV:owner comme propriété protégée et PEUVENT retourner un élément DAV:owner vide comme valeur de propriété lorsque aucune information de propriétaire n'est disponible.

<!ELEMENT owner (href?)>

5.1.1 Exemple : Restitution de DAV:owner

Cet exemple montre une demande de client pour la valeur de la propriété DAV:owner d'une ressource de collection avec l'URL http://www.example.com/papers/. Le principal qui fait la demande est authentifié en utilisant l'authentification par résumé. La valeur de DAV:owner est l'URL http://www.example.com/acl/users/gstein, enveloppé dans l'élément XML DAV:href.

>> Demande <<

```
PROPFIND /papers/ HTTP/1.1
Host: www.example.com
Content-type: text/xml; charset="utf-8"
Content-Length: xxx
```

```
Depth: 0
Authorization: Digest username="jim",
realm="users@example.com", nonce="...",
uri="/papers/", response="...", opaque="..."
<?xml version="1.0" encoding="utf-8" ?>
<D:propfind xmlns:D="DAV:">
<D:prop>
<D:owner/>
</D:prop>
</D:propfind>
```

>> Réponse <<

```
HTTP/1.1 207 Multi-Status
Content-Type: text/xml; charset="utf-8"
Content-Length: xxx
<?xml version="1.0" encoding="utf-8" ?>
<D:multistatus xmlns:D="DAV:">
<D:response>
<D:href>http://www.example.com/papers/</D:href>
<D:propstat>
<D:prop>
<D:owner>
<D:href>http://www.example.com/acl/users/gstein</D:href>
</D:owner>
</D:prop>
<D:status>HTTP/1.1 200 OK</D:status>
</D:propstat>
</D:response>
</D:multistatus>
```

5.1.2 Exemple : tentative d'établir DAV:owner

L'exemple suivant montre une demande de client pour modifier la valeur de la propriété DAV:owner sur la ressource qui a l'URL <http://www.example.com/papers>. Comme DAV:owner est une propriété protégée sur ce serveur particulier, il répond par un code 207 (Multi-Status) qui contient un code d'état 403 (Interdit) pour l'action de régler DAV:owner. Le paragraphe 8.2.1 de la [RFC2518] décrit les informations du code d'état PROPPATCH, la Section 11 de la [RFC2518] décrit la réponse Multi-Status et les paragraphes 1.6 et 3.12 de la [RFC3253] décrivent le rangement des erreurs supplémentaires pour les tentatives de PROPPATCH sur les propriétés protégées.

>> Demande <<

```
PROPPATCH /papers/ HTTP/1.1
Host: www.example.com
Content-type: text/xml; charset="utf-8"
Content-Length: xxx
Depth: 0
Authorization: Digest username="jim",
realm="users@example.com", nonce="...",
uri="/papers/", response="...", opaque="..."
<?xml version="1.0" encoding="utf-8" ?>
<D:propertyupdate xmlns:D="DAV:">
  <D:set>
    <D:prop>
      <D:owner>
        <D:href>http://www.example.com/acl/users/jim</D:href>
      </D:owner>
    </D:prop>
  </D:set>
</D:propertyupdate>
```

>> Réponse <<

```
HTTP/1.1 207 Multi-Status
Content-Type: text/xml; charset="utf-8"
Content-Length: xxx
<?xml version="1.0" encoding="utf-8" ?>
<D:multistatus xmlns:D="DAV:">
  <D:response>
    <D:href>http://www.example.com/papers/</D:href>
    <D:propstat>
      <D:prop><D:owner/></D:prop>
      <D:status>HTTP/1.1 403 Interdit</D:status>
      <D:responsedescription>
        <D:error><D:cannot-modify-protected-property/></D:error>
        Échec à régler la propriété protégée (DAV:owner)
      </D:responsedescription>
    </D:propstat>
  </D:response>
</D:multistatus>
```

5.2 DAV:group

Cette propriété identifie un certain principal comme étant le "groupe" de ressources. Cette propriété est couramment trouvée sur des répertoires qui mettent en œuvre le modèle de privilèges Unix.

Les serveurs PEUVENT mettre en œuvre DAV:group comme propriété protégée et PEUVENT retourner un élément DAV:group vide comme valeur de propriété dans le cas où aucune information de groupe n'est disponible.

<!ELEMENT group (href?)>

5.3 DAV:supported-privilege-set

C'est une propriété protégée qui identifie les privilèges définis pour la ressource.

<!ELEMENT supported-privilege-set (privilèges pris en charge*)>

Chaque privilège apparaît comme un élément XML, où les privilèges agrégés énumèrent comme sous éléments tous les privilèges qu'ils agrègent.

<!ELEMENT supported-privilege (privilege, abstract?, description, supported-privilege*)>
<!ELEMENT privilege ANY>

Un privilège abstrait NE DOIT PAS être utilisé dans un ACE pour cette ressource. Le serveurs DOIVENT faire échouer une tentative d'établir un privilège abstrait.

<!ELEMENT abstract EMPTY>

Une description est une description lisible par l'homme de ce à quoi ce privilège contrôle l'accès. Les serveurs DOIVENT indiquer le langage humain de la description en utilisant l'attribut xml:lang et DEVRAIENT considérer l'en-tête de demande HTTP Accept-Language lors du choix d'un des multiples langages disponibles.

<!ELEMENT description #PCDATA>

Il est envisagé qu'un client administratif WebDAV à capacité d'ACL fasse la liste des privilèges pris en charge dans une boîte de dialogue, et permette à l'utilisateur de choisir des privilèges non abstraits à appliquer dans un ACE. L'arborescence des privilèges est utile dans les programmes pour transposer les privilèges bien connus (définis par WebDAV ou d'autres groupes de normes) en privilèges qui sont supportés par toute mise en œuvre de serveur particulière. L'arborescence des privilèges sert aussi à cacher la complexité dans les mises en œuvre qui permettent qu'un grand nombre de privilèges soient définis en affichant les agrégats à l'utilisateur.

5.3.1 Exemple : Restitution d'une liste de privilèges supportés sur une ressource

Cet exemple montre une demande de client pour la propriété DAV:supported-privilege-set sur la ressource

http://www.example.com/papers/. La valeur de la propriété DAV:supported-privilege-set est une arborescence de privilèges pris en charge (en utilisant "[XML Namespace , localname]" pour identifier chaque privilège) :

```
[DAV:, all] (aggregate, abstract)
   |
   +-- [DAV:, read] (aggregate)
          |
          +-- [DAV:, read-acl] (abstract)
          +-- [DAV:, read-current-user-privilege-set] (abstract)
   |
   +-- [DAV:, write] (aggregate)
          |
          +-- [DAV:, write-acl] (abstract)
          +-- [DAV:, write-properties]
          +-- [DAV:, write-content]
   |
   +-- [DAV:, unlock]
```

Cette arborescence de privilèges n'est pas normative (mais elle reflète les règles d'agrégation normatives données au paragraphe 3.12), et de nombreuses arborescences de privilèges sont possibles.

>> Demande <<

```
PROPFIND /papers/ HTTP/1.1
Host: www.example.com
Content-type: text/xml; charset="utf-8"
Content-Length: xxx
Depth: 0
Authorization: Digest username="gclemm",
 realm="users@example.com", nonce="...",
 uri="/papers/", response="...", opaque="..."
<?xml version="1.0" encoding="utf-8" ?>
<D:propfind xmlns:D="DAV:">
  <D:prop>
    <D:supported-privilege-set/>
  </D:prop>
</D:propfind>
```

>> Réponse <<

```
HTTP/1.1 207 Multi-Status
Content-Type: text/xml; charset="utf-8"
Content-Length: xxx
<?xml version="1.0" encoding="utf-8" ?>
<D:multistatus xmlns:D="DAV:">
  <D:response>
    <D:href>http://www.example.com/papers/</D:href>
    <D:propstat>
      <D:prop>
        <D:supported-privilege-set>
          <D:supported-privilege>
            <D:privilege><D:all/></D:privilege>
           <D:abstract/>
            <D:description xml:lang="fr">
              Toute opération
            </D:description>
            <D:supported-privilege>
              <D:privilege><D:read/></D:privilege>
              <D:description xml:lang="fr"> Lire tout objet </D:description>
              <D:supported-privilege>
                <D:privilege><D:read-acl/></D:privilege>
                <D:abstract/>
                <D:description xml:lang="en">Lire l'ACL</D:description>
```

```
          </D:supported-privilege>
          <D:supported-privilege>
            <D:privilege>
              <D:read-current-user-privilege-set/>
            </D:privilege>
            <D:abstract/>
            <D:description xml:lang="fr">
              Lire la propriété d'ensemble de privilèges d'utilisateur actuelle
            </D:description>
          </D:supported-privilege>
        </D:supported-privilege>
        <D:supported-privilege>
          <D:privilege><D:write/></D:privilege>
          <D:description xml:lang="fr">
            Écrire tout objet
        </D:description>
          <D:supported-privilege>
            <D:privilege><D:write-acl/></D:privilege>
            <D:description xml:lang="fr">
              Écrire l'ACL
            </D:description>
            <D:abstract/>
          </D:supported-privilege>
          <D:supported-privilege>
            <D:privilege><D:write-properties/></D:privilege>
            <D:description xml:lang="fr">
              Écrire les propriétés
            </D:description>
          </D:supported-privilege>
          <D:supported-privilege>
            <D:privilege><D:write-content/></D:privilege>
            <D:description xml:lang="fr">
              Écrire le contenu de la ressource
            </D:description>
          </D:supported-privilege>
        </D:supported-privilege>
        <D:supported-privilege>
          <D:privilege><D:unlock/></D:privilege>
          <D:description xml:lang="fr">
            Déverrouiller la ressource
          </D:description>
        </D:supported-privilege>
      </D:supported-privilege>
    </D:supported-privilege-set>
  </D:prop>
  <D:status>HTTP/1.1 200 OK</D:status>
 </D:propstat>
</D:response>
</D:multistatus>
```

5.4 DAV:current-user-privilege-set

DAV:current-user-privilege-set est une propriété protégée contenant l'ensemble exact de privilèges (calculé par le serveur) accordé à l'utilisateur HTTP actuellement authentifié. Les privilèges agrégés et les privilèges qu'ils contiennent sont énumérés. Un agent d'utilisateur peut utiliser la valeur de cette propriété pour ajuster son interface d'utilisateur pour effectuer des actions inaccessibles (par exemple, en grisant un élément de menu ou un bouton) pour lesquelles le principal actuel n'a pas de permission. Cette propriété est aussi utile pour déterminer quelles opérations peut effectuer le principal actuel, sans avoir à réellement exécuter une opération.

```
<!ELEMENT current-user-privilege-set (privilege*)>
<!ELEMENT privilege ANY>
```

Si un privilège spécifique est accordé à l'utilisateur actuel, ce privilège doit appartenir à l'ensemble de privilèges qui peuvent être établis sur cette ressource. Donc, chaque élément dans la propriété DAV:current-user-privilege-set DOIT identifier un privilège non abstrait à partir de la propriété DAV:supported-privilege-set.

5.4.1 Exemple : Restitution de l'ensemble courant de privilèges alloués à l'utilisateur

En continuant l'exemple du paragraphe 5.3.1, cet exemple montre un client qui demande la propriété DAV:current-user-privilege-set de la ressource avec l'URL http://www.example.com/papers/. Le nom d'utilisateur du principal qui fait la demande est "khare", et l'authentification par résumé est utilisée dans la demande. Le principal qui a pour nom d'utilisateur "khare" a reçu le privilège DAV:read. Comme le privilège DAV:read contient les privilèges DAV:read-acl et DAV:read-current-user-privilege-set (voir au paragraphe 5.3.1) le principal qui a le nom d'utilisateur "khare" peut lire la propriété ACL, et la propriété DAV:current-user-privilege-set. Cependant, les privilèges DAV:all, DAV:read-acl, DAV:write-acl et DAV:read-current-user-privilege-ne sont pas énumérés dans la valeur de DAV:current-user-privilege-set, car (pour cet exemple) ils sont des privilèges abstraits. DAV:write ne figure pas parce que le principal du nom d'utilisateur "khare" n'est pas sur la liste d'un ACE qui accorde à ce principal la permission d'écriture.

>> Demande <<

```
PROPFIND /papers/ HTTP/1.1
Host: www.example.com
Content-type: text/xml; charset="utf-8"
Content-Length: xxx
Depth: 0
Authorization: Digest username="khare",
  realm="users@example.com", nonce="...",
  uri="/papers/", response="...", opaque="..."
<?xml version="1.0" encoding="utf-8" ?>
<D:propfind xmlns:D="DAV:">
  <D:prop>
    <D:current-user-privilege-set/>
  </D:prop>
</D:propfind>
```

>> Réponse <<

```
HTTP/1.1 207 Multi-Status
Content-Type: text/xml; charset="utf-8"
Content-Length: xxx
<?xml version="1.0" encoding="utf-8" ?>
<D:multistatus xmlns:D="DAV:">
  <D:response>
  <D:href>http://www.example.com/papers/</D:href>
  <D:propstat>
    <D:prop>
      <D:current-user-privilege-set>
        <D:privilege><D:read/></D:privilege>
      </D:current-user-privilege-set>
    </D:prop>
    <D:status>HTTP/1.1 200 OK</D:status>
  </D:propstat>
  </D:response>
</D:multistatus>
```

5.5 DAV:acl

C'est une propriété protégée qui spécifie la liste des éléments de contrôle d'accès (ACE), qui définit quels principaux vont obtenir quels privilèges pour cette ressource.

```
<!ELEMENT acl (ace*) >
```

Chaque élément DAV:ace spécifie l'ensemble de privilèges à accorder ou refuser à un seul principal. Si la propriété DAV:acl est vide, aucun privilège n'est accordé à aucun principal.

<!ELEMENT ace ((principal | invert), (grant|deny), protected?, inherited?)>

5.5.1 Principal ACE

L'élément DAV:principal identifie le principal auquel cet ACE s'applique.

<!ELEMENT principal (href | all | authenticated | unauthenticated | property | self)>

L'utilisateur actuel ne correspond à DAV:href que si il est authentifié comme étant le principal (ou un membre du) identifié par l'URL contenu par ce DAV:href.

L'utilisateur actuel correspond toujours à DAV:all.

<!ELEMENT all EMPTY>

L'utilisateur actuel ne correspond à DAV:authenticated que si il est authentifié.

<!ELEMENT authenticated EMPTY>

L'utilisateur actuel ne correspond à DAV:unauthenticated que si il n'est pas authentifié.

<!ELEMENT unauthenticated EMPTY>

DAV:all est l'union de DAV:authenticated, et de DAV:unauthenticated. Pour une certaine demande, l'utilisateur correspond à DAV:authenticated, ou à DAV:unauthenticated, mais pas aux deux (c'est à dire que DAV:authenticated et DAV:unauthenticated sont des ensembles disjoints).

L'utilisateur actuel ne correspond à un principal DAV:property dans une propriété DAV:acl d'une ressource que si la valeur de la propriété identifiée de cette ressource contient au plus un élément XML DAV:href, la valeur d'URI de DAV:href identifie un principal, et l'utilisateur actuel est authentifié comme étant ce principal (ou un membre de ce principal). Par exemple, si l'élément DAV:property contenait <DAV:owner/>, l'utilisateur actuel ne correspondrait au principal DAV:property que si l'utilisateur actuel était authentifié comme correspondant au principal identifié par la propriété DAV:owner de la ressource.

<!ELEMENT property ANY>

L'utilisateur actuel ne correspond à DAV:self dans une propriété DAV:acl de la ressource que si cette ressource est un principal et si ce principal correspond à l'utilisateur actuel ou, si le principal est un groupe, un membre de ce groupe correspond à l'utilisateur actuel.

<!ELEMENT self EMPTY>

Certains serveurs peuvent prendre en charge les ACE qui s'appliquent aux utilisateurs qui NE correspondent PAS au principal actuel, par exemple, tous les utilisateurs qui ne sont pas dans un certain groupe. Ceci peut être fait en enveloppant l'élément DAV:principal dans DAV:invert.

<!ELEMENT invert principal>

5.5.2 ACE d'accord et de refus

Chaque élément DAV:grant ou DAV:deny spécifie l'ensemble de privilèges à accorder ou à refuser au principal spécifié. Un élément DAV:grant ou DAV:deny de la DAV:acl d'une ressource DOIT seulement contenir des éléments non abstraits spécifiés dans le DAV:supported-privilege-set de cette ressource.

<!ELEMENT grant (privilege+)>
<!ELEMENT deny (privilege+)>
<!ELEMENT privilege ANY>

5.5.3 Protection d'ACE

Un serveur indique qu'un ACE est protégé en incluant l'élément DAV:protected dans l'ACE. Si l'ACL d'une ressource contient un ACE avec un élément DAV:protected, une tentative de suppression de cet ACE de l'ACL DOIT échouer.

<!ELEMENT protected EMPTY>

5.5.4 Héritage d'ACE

La présence d'un élément DAV:inherited indique que cet ACE est hérité d'une autre ressource qui est identifiée par l'URL contenu dans un élément DAV:href. Un ACE hérité ne peut pas être modifié directement, mais par contre l'ACL sur la ressource de laquelle il est hérité doit être modifiée.

Noter que l'héritage d'ACE n'est pas le même que l'initialisation d'ACL. L'initialisation d'ACL définit l'ACL que va utiliser une ressource nouvellement créée (si non spécifiée). L'héritage d'ACE se réfère à un ACE qui est partagé logiquement – où une mise à jour de la ressource contenant un ACE va affecter l'ACE de chaque ressource qui hérite de cet ACE. La méthode par laquelle les ACL sont initialisées ou par laquelle les ACE sont hérités n'est pas définie par le présent document.

<!ELEMENT inherited (href)>

5.5.5 Exemple : Restitution de la liste de contrôle d'accès d'une ressource

En continuant l'exemple des paragraphes 5.3.1 et 5.4.1, cet exemple montre un client qui demande la propriété DAV:acl à la ressource qui a l'URL http://www.example.com/papers/. Il y a deux ACE définis dans cette ACL :

ACE n° 1 : le groupe identifié par l'URL http://www.example.com/acl/groups/maintainers (le groupe des mainteneurs de site) reçoit le privilège DAV:write. Comme (pour cet exemple) DAV:write contient le privilège DAV:write-acl (voir le paragraphe 5.3.1) cela signifie que le groupe "maintainers" peut aussi modifier la liste de contrôle d'accès.

ACE n° 2 : tous les principaux (DAV:all) reçoivent le privilège DAV:read. Comme (pour cet exemple) DAV:read contient DAV:read-acl et DAV:read-current-user-privilege-set, cela signifie que tous les utilisateurs (incluant tous les membres du groupe "maintainers") peuvent lire la propriété DAV:acl et la propriété DAV:current-user-privilege-set.

>> Demande <<

```
PROPFIND /papers/ HTTP/1.1
Host: www.example.com
Content-type: text/xml; charset="utf-8"Content-Length: xxx
Depth: 0
Authorization: Digest username="masinter",
 realm="users@example.com", nonce="...",
 uri="/papers/", response="...", opaque="..."
<D:propfind xmlns:D="DAV:">
 <D:prop>
  <D:acl/>
 </D:prop>
</D:propfind>
```

>> Réponse <<

```
HTTP/1.1 207 Multi-Status
Content-Type: text/xml; charset="utf-8"
Content-Length: xxx
<D:multistatus xmlns:D="DAV:">
 <D:response>
  <D:href>http://www.example.com/papers/</D:href>
  <D:propstat>
   <D:prop>
    <D:acl>
    <D:ace>
     <D:principal>
      <D:href
      >http://www.example.com/acl/groups/maintainers</D:href>
     </D:principal>
     <D:grant>
      <D:privilege><D:write/></D:privilege>
```

```
        </D:grant>
       </D:ace>
       <D:ace>
        <D:principal>
         <D:all/>
        </D:principal>
        <D:grant>
         <D:privilege><D:read/></D:privilege>
        </D:grant>
       </D:ace>
      </D:acl>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
</D:multistatus>
```

5.6 DAV:acl-restrictions

Cette propriété protégée définit les types d'ACL pris en charge par ce serveur, pour éviter que des clients obtiennent inutilement des erreurs. Lorsque un client essaye d'établir une ACL via la méthode ACL, le serveur peut rejeter la tentative d'établir l'ACL comme spécifié. Les propriétés suivantes indiquent les restrictions que le client doit observer avant d'établir une ACL :
<grant-only> les ACE de refus ne sont pas pris en charge
<no-invert> les ACE inversés ne sont pas pris en charge
<deny-before-grant> tous les ACE de refus doivent survenir avant tout ACE d'octroi
<required-principal> indique de quels principaux la présence est exigée

<!ELEMENT acl-restrictions (grant-only?, no-invert?, deny-before-grant?, required-principal?)>

5.6.1 DAV:grant-only

Cet élément indique que les ACE qui refusent des clauses ne sont pas permis.

<!ELEMENT grant-only EMPTY>

5.6.2 Contrainte d'ACE DAV:no-invert

Cet élément indique que les ACE avec l'élément <invert> ne sont pas permis.

<!ELEMENT no-invert EMPTY>

5.6.3 DAV:deny-before-grant

Cet élément indique que tous les ACE de refus doivent précéder tous les ACE d'octroi.

<!ELEMENT deny-before-grant EMPTY>

5.6.4 Élément required-principal

Les éléments de principaux requis identifient quels principaux doivent avoir un ACE défini dans l'ACL.

<!ELEMENT required-principal (all? | authenticated? | unauthenticated? | self? | href* | property*)>

Par exemple, l'élément suivant exige que l'ACL contienne un ACE de propriété DAV:owner :

```
<D:required-principal xmlns:D="DAV:">
  <D:property><D:owner/></D:property>
</D:required-principal>
```

5.6.5 Exemple : Restitution de DAV:acl-restrictions

Dans cet exemple, le client demande la valeur de la propriété DAV:acl-restrictions. L'authentification par résumé fournit des accréditifs pour le principal qui fait fonctionner le client.

>> Demande <<

```
PROPFIND /papers/ HTTP/1.1
Host: www.example.com
Content-type: text/xml; charset="utf-8"
Content-Length: xxx
Depth: 0
Authorization: Digest username="srcarter",
  realm="users@example.com", nonce="...",
  uri="/papers/", response="...", opaque="..."
<?xml version="1.0" encoding="utf-8" ?>
<D:propfind xmlns:D="DAV:">
  <D:prop>
    <D:acl-restrictions/>
  </D:prop>
</D:propfind>
```

>> Réponse <<

```
HTTP/1.1 207 Multi-Status
Content-Type: text/xml; charset="utf-8"
Content-Length: xxx

<?xml version="1.0" encoding="utf-8" ?>
<D:multistatus xmlns:D="DAV:">
  <D:response>
    <D:href>http://www.example.com/papers/</D:href>
    <D:propstat>
      <D:prop>
        <D:acl-restrictions>
          <D:grant-only/>
          <D:required-principal>
            <D:all/>
          </D:required-principal>
        </D:acl-restrictions>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
</D:multistatus>
```

5.7 DAV:inherited-acl-set

Cette propriété protégée contient un ensemble d'URL qui identifient d'autres ressources qui contrôlent aussi l'accès à cette ressource. Pour avoir un privilège sur une ressource, non seulement l'ACL sur cette ressource (spécifiée dans la propriété DAV:acl de cette ressource) doit accorder le privilège, mais aussi doit le faire l'ACL de chaque ressource identifiée dans la propriété DAV:inherited-acl-set de cette ressource. Effectivement, les privilèges accordés par l'ACL actuelle sont ajoutés par l'opérateur logique ET avec les privilèges accordés par chaque ACL héritée.

```
<!ELEMENT inherited-acl-set (href*)>
```

5.8 DAV:principal-collection-set

Cette propriété protégée d'une ressource contient un ensemble d'URL qui identifient les collections racines qui contiennent les principaux disponibles sur le serveur qui met en œuvre cette ressource. Un agent d'utilisateur de protocole de contrôle d'accès WebDAV pourrait utiliser le contenu de DAV:principal-collection-set pour restituer la propriété DAV:displayname (spécifiée au paragraphe 13.2 de la [RFC2518]) de tous les principaux sur ce serveur, donnant par là les noms lisibles par l'homme pour chaque principal qui pourrait être affiché sur une interface d'utilisateur.

<!ELEMENT principal-collection-set (href*)>

Comme différents serveurs peuvent contrôler des parties différente de l'espace de noms d'URL, différentes ressources sur le même hôte PEUVENT avoir des valeurs différentes de DAV:principal-collection-set. Les collections spécifiées dans le DAV:principal-collection-set PEUVENT être localisées dans des hôtes différents à partir de la ressource. Les URL dans DAV:principal-collection-set DEVRAIENT être des URL de schéma http ou https. Pour des raisons de sécurité et d'adaptabilité, un serveur PEUT rapporter seulement un sous ensemble de l'ensemble entier de collections de principaux connus, et donc, les clients ne devraient pas supposer qu'ils ont restitué une liste exhaustive. De plus, un serveur PEUT choisir de ne rapporter aucune des collections de principaux qu'il connaît, auquel cas la valeur de la propriété sera vide.

La valeur de DAV:principal-collection-set donne la portée du rapport DAV:principal-property-search (défini au paragraphe 9.4). Les clients utilisent le rapport DAV:principal-property-search pour remplir leur interface d'utilisateur avec une liste de principaux. Donc, les serveurs qui limitent la capacité d'un client à obtenir des informations sur les principaux vont interférer avec la capacité du client à manipuler les listes de contrôle d'accès, du fait de la difficulté d'obtenir l'URL d'un principal pour l'utiliser dans un ACE.

5.8.1 Exemple : Restitution de DAV:principal-collection-set

Dans cet exemple, le client demande la valeur de la propriété DAV:principal-collection-set sur la ressource de collection identifiée par l'URL http://www.example.com/papers/. La propriété contient les deux URL, http://www.example.com/acl/users/ et http://www.example.com/acl/groups/, tous deux enveloppés dans des éléments XML DAV:href. L'authentification par résumé fournit des accréditifs pour le principal qui fait fonctionner le client.

Le client peut raisonnablement suivre cette demande avec deux demandes PROPFIND séparées pour restituer la propriété DAV:displayname des membres des deux collections (/acl/users et /acl/groups). Ces informations pourraient être utilisées pour l'affichage sur l'interface d'utilisateur pour créer des entrées de contrôle d'accès.

>> Demande <<

```
PROPFIND /papers/ HTTP/1.1
Host: www.example.com
Content-type: text/xml; charset="utf-8"
Content-Length: xxx
Depth: 0
Authorization: Digest username="yarong",
  realm="users@example.com", nonce="...",
  uri="/papers/", response="...", opaque="..."
 <?xml version="1.0" encoding="utf-8" ?>
<D:propfind xmlns:D="DAV:">
  <D:prop>
    <D:principal-collection-set/>
  </D:prop>
 </D:propfind>
```

>> Réponse <<

```
HTTP/1.1 207 Multi-Status
Content-Type: text/xml; charset="utf-8"
Content-Length: xxx

<?xml version="1.0" encoding="utf-8" ?>
<D:multistatus xmlns:D="DAV:">
  <D:response>
    <D:href>http://www.example.com/papers/</D:href>
    <D:propstat>
      <D:prop>
        <D:principal-collection-set>
          <D:href>http://www.example.com/acl/users/</D:href>
          <D:href>http://www.example.com/acl/groups/</D:href>
        </D:principal-collection-set>
      </D:prop>
```

```
    <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
</D:multistatus>
```

5.9 Exemple : PROPFIND pour restituer les propriétés de contrôle d'accès

L'exemple suivant montre comment les informations de contrôle d'accès peuvent être restituées en utilisant la méthode PROPFIND pour aller chercher les valeurs des propriétés DAV:owner, DAV:supported-privilege-set, DAV:current-user-privilege-set, et DAV:acl.

>> Demande <<

```
PROPFIND /top/container/ HTTP/1.1
Host: www.example.com
Content-type: text/xml; charset="utf-8"
Content-Length: xxx
Depth: 0
Authorization: Digest username="ejw",
  realm="users@example.com", nonce="...",
  uri="/top/container/", response="...", opaque="..."

<?xml version="1.0" encoding="utf-8" ?>
<D:propfind xmlns:D="DAV:">
  <D:prop>
    <D:owner/>
    <D:supported-privilege-set/>
    <D:current-user-privilege-set/>
    <D:acl/>
  </D:prop>
</D:propfind>
```

>> Réponse <<

```
HTTP/1.1 207 Multi-Status
Content-Type: text/xml; charset="utf-8"
Content-Length: xxx

<?xml version="1.0" encoding="utf-8" ?>
<D:multistatus xmlns:D="DAV:" xmlns:A="http://www.example.com/acl/">
  <D:response>
    <D:href>http://www.example.com/top/container/</D:href>
    <D:propstat>
      <D:prop>
        <D:owner>
          <D:href>http://www.example.com/users/gclemm</D:href>
        </D:owner>
        <D:supported-privilege-set>
          <D:supported-privilege>
            <D:privilege><D:all/></D:privilege>
            <D:abstract/>
            <D:description xml:lang="fr">
              Toute opération
            </D:description>
            <D:supported-privilege>
              <D:privilege><D:read/></D:privilege>
              <D:description xml:lang="fr">
                Lire tout objet
              </D:description>
            </D:supported-privilege>
            <D:supported-privilege>
              <D:privilege><D:write/></D:privilege>
```

```
      <D:abstract/>
      <D:description xml:lang="fr">
       Écrite tout objet
      </D:description>
      <D:supported-privilege>
     <D:privilege><A:create/></D:privilege>
       <D:description xml:lang="fr">
        Créer un objet
       </D:description>
      </D:supported-privilege>
      <D:supported-privilege>
       <D:privilege><A:update/></D:privilege>
       <D:description xml:lang="fr">
        Mettre à jour un objet
       </D:description>
      </D:supported-privilege>
    </D:supported-privilege>
    <D:supported-privilege>
     <D:privilege><A:delete/></D:privilege>
     <D:description xml:lang="fr">
      Supprimer un objet
     </D:description>
    </D:supported-privilege>
    <D:supported-privilege>
     <D:privilege><D:read-acl/></D:privilege>
     <D:description xml:lang="fr">
      Lire l'ACL
     </D:description>
    </D:supported-privilege>
    <D:supported-privilege>
     <D:privilege><D:write-acl/></D:privilege>
     <D:description xml:lang="fr">
      Écrire l'ACL
     </D:description>
    </D:supported-privilege>
  </D:supported-privilege>
</D:supported-privilege-set>
<D:current-user-privilege-set>
  <D:privilege><D:read/></D:privilege>
  <D:privilege><D:read-acl/></D:privilege>
</D:current-user-privilege-set>
<D:acl>
  <D:ace>
    <D:principal>
      <D:href>http://www.example.com/users/esedlar</D:href>
    </D:principal>
    <D:grant>
      <D:privilege><D:read/></D:privilege>
      <D:privilege><D:write/></D:privilege>
      <D:privilege><D:read-acl/></D:privilege>
    </D:grant>
  </D:ace>
  <D:ace>
    <D:principal>
      <D:href>http://www.example.com/groups/mrktng</D:href>
    </D:principal>
    <D:deny>
      <D:privilege><D:read/></D:privilege>
    </D:deny>
  </D:ace>
  <D:ace>
    <D:principal>
      <D:property><D:owner/></D:property>
```

```
        </D:principal>
        <D:grant>
          <D:privilege><D:read-acl/></D:privilege>
          <D:privilege><D:write-acl/></D:privilege>
        </D:grant>
      </D:ace>
      <D:ace>
        <D:principal><D:all/></D:principal>
        <D:grant>
          <D:privilege><D:read/></D:privilege>
        </D:grant>
        <D:inherited>
          <D:href>http://www.example.com/top</D:href>
        </D:inherited>
      </D:ace>
    </D:acl>
  </D:prop>
  <D:status>HTTP/1.1 200 OK</D:status>
 </D:propstat>
</D:response>
</D:multistatus>
```

La valeur de la propriété DAV:owner est un seul élément XML DAV:href qui contient l'URL du principal qui possède cette ressource.

La valeur de la propriété DAV:supported-privilege-set est une arborescence de privilèges pris en charge (en utilisant "[Espace de noms XML, nom local]" pour identifier chaque privilège) :

```
[DAV:, all] (aggregate, abstract)
    |
    +-- [DAV:, read]
    +-- [DAV:, write] (aggregate, abstract)
            |
            +-- [http://www.example.com/acl, create]
            +-- [http://www.example.com/acl, update]
            +-- [http://www.example.com/acl, delete]
    +-- [DAV:, read-acl]
    +-- [DAV:, write-acl]
```

La propriété DAV:current-user-privilege-set contient deux privilèges, DAV:read, et DAV:read-acl. Cela indique que l'utilisateur actuellement authentifié a seulement la capacité de lire la ressource, et de lire la propriété DAV:acl sur la ressource. La propriété DAV:acl contient un ensemble de quatre ACE :

ACE #1 : le principal identifié par l'URL http://www.example.com/users/esedlar a les privilèges DAV:read, DAV:write, et DAV:read-acl.

ACE #2 : le privilège DAV:read est refusé aux principaux identifiés par l'URL http://www.example.com/groups/mrktng. Dans cet exemple, L'URL principal identifie un groupe.

ACE #3 : dans cet ACE, le principal est une propriété principale, spécifiquement, la propriété DAV:owner. Pour évaluer cet ACE, la valeur de la propriété DAV:owner est restituée, et est examinée pour voir si elle contient un élément XML DAV:href. Si oui, l'URL au sein de l'élément DAV:href est lu, et il identifie un principal. Dans cet ACE, le propriétaire a les privilèges DAV:read-acl, et DAV:write-acl.

ACE #4 : cet ACE accorde aux principaux DAV:all (tous les utilisateurs) le privilège DAV:read. Cet ACE est hérité de la ressource http://www.example.com/top, la collection parente de cette ressource.

6. Évaluation d'ACL

Les ACL WebDAV sont évaluées de la même manière que les ACL sur Windows NT et dans NFSv4 [RFC3530]). Une ACL est évaluée pour déterminer si l'accès sera ou non accordé pour une demande WebDAV. Les ACE sont conservés dans un certain

ordre, et sont évalués jusqu'à ce que les permissions requises par la demande en cours aient été accordées, point auquel l'évaluation d'ACL se termine et où l'accès est accordé. Si, durant l'évaluation d'ACL, un ACE <deny> (correspondant à l'utilisateur actuel) est rencontré pour un privilège qui n'a pas encore été accordé, l'évaluation d'ACL se termine et l'accès est refusé. L'échec à avoir tous les privilèges demandés accordés résulte en un refus d'accès.

Noter que la sémantique de nombreux autres systèmes d'ACL existants peut être représentée via ce mécanisme, en mêlant les ACE de refus et d'octroi. Par exemple, considérons le schéma de privilège standard "rwx" utilisé par UNIX. Dans ce schéma, si l'utilisateur actuel est le propriétaire du fichier, l'accès est accordé si le bit de privilège correspondant est établi et refusé si il n'est pas établi, sans considération des permissions établies sur le groupe du fichier et pour le reste du monde. Une ACL pour les permissions UNIX de "r--rw-r--" pourrait être construite de la façon suivante :

```
<D:acl>
  <D:ace>
    <D:principal>
      <D:property><D:owner/></D:property>
    </D:principal>
    <D:grant>
      <D:privilege><D:read/></D:privilege>
    </D:grant>
  </D:ace>
  <D:ace>
    <D:principal>
      <D:property><D:owner/></D:property>
    </D:principal>
    <D:deny>
      <D:privilege><D:all/></D:privilege>
    </D:deny>
  </D:ace>
  <D:ace>
    <D:principal>
      <D:property><D:group/></D:property>
    </D:principal>
    <D:grant>
      <D:privilege><D:read/></D:privilege>
      <D:privilege><D:write/></D:privilege>
    </D:grant>
  </D:ace>
  <D:ace>
    <D:principal>
      <D:property><D:group/></D:property>
    </D:principal>
    <D:deny>
      <D:privilege><D:all/></D:privilege>
    </D:deny>
  </D:ace>
  <D:ace>
    <D:principal><D:all></D:principal>
    <D:grant>
      <D:privilege><D:read/></D:privilege>
    </D:grant>
  </D:ace>
</D:acl>
```

et les <acl-restrictions> seraient définie comme :

```
<D:no-invert/>
<D:required-principal>
  <D:all/>
  <D:property><D:owner/></D:property>
  <D:property><D:group/><D:group/>
</D:required-principal>
```

Noter que le client obtient quand même des erreurs d'un serveur UNIX en dépit de l'observation des <acl-restrictions>, incluant

<D:allowed-principal> (ajoutant un ACE qui spécifie un principal autre que ceux de l'ACL ci-dessus) ou <D:ace-conflict> (en essayant de réordonner les ACE dans l'exemple ci-dessus) car ces sémantiques de mise en œuvre particulières sont trop complexes pour être capturées par de simples (mais générales) restrictions déclaratives.

7. Contrôle d'accès et méthodes existantes

La présente section définit l'impact des fonctions de contrôle d'accès sur les méthodes existantes.

7.1 Toute méthode HTTP

7.1.1 Traitement d'erreur

Le mécanisme d'ACL WebDAV exige l'utilisation de la méthode HTTP "preconditions" comme décrit au paragraphe 1.6 de la RFC3253 pour TOUTES les méthodes HTTP. Toutes les méthodes HTTP ont une précondition supplémentaire appelée DAV:need-privileges. Si une méthode HTTP échoue à cause de privilèges insuffisants, le corps de réponse à l'erreur "403 Interdit" DOIT contenir l'élément <DAV:error>, qui à son tour contient l'élément <DAV:need-privileges>, qui contient un ou plusieurs éléments <DAV:resource> indiquant quelle ressource a des privilèges insuffisants, et quels sont les privilèges manquants :

```
<!ELEMENT need-privileges (resource)* >
<!ELEMENT resource ( href , privilege ) >
```

Comme certaines méthodes exigent plusieurs permissions sur de multiples ressources, ces informations sont nécessaires pour résoudre toute ambiguïté. Il n'est pas exigé que toutes les violations de privilège soient rapportées - pour des raisons de mise en œuvre, certains serveurs peuvent ne rapporter que la première violation de privilège. Par exemple :

>> Demande <<

```
MOVE /a/b/ HTTP/1.1
Host: www.example.com
Destination: http://www.example.com/c/d
```

>> Réponse <<

```
HTTP/1.1 403 Interdit
Content-Type: text/xml; charset="utf-8"
Content-Length: xxx

<D:error xmlns:D="DAV:">
  <D:need-privileges>
    <D:resource>
      <D:href>/a</D:href>
      <D:privilege><D:unbind/></D:privilege>
    </D:resource>
    <D:resource>
      <D:href>/c</D:href>
      <D:privilege><D:bind/></D:privilege>
    </D:resource>
  </D:need-privileges>
</D:error>
```

7.2 OPTIONS

Si le serveur prend en charge le contrôle d'accès, il DOIT retourner "access-control" comme champ dans l'en-tête de réponse DAV d'une demande OPTIONS sur toute ressource mise en œuvre par ce serveur. Une valeur de "access-control" dans l'en-tête DAV DOIT indiquer que le serveur prend en charge toutes les exigences de niveau DOIT et les caractéristiques EXIGÉES spécifiées dans le présent document.

7.2.1 Exemple - OPTIONS

>> Demande <<

```
OPTIONS /foo.html HTTP/1.1
Host: www.example.com
Content-Length: 0
```

>> Réponse <<

```
HTTP/1.1 200 OK
DAV: 1, 2, access-control
Allow: OPTIONS, GET, PUT, PROPFIND, PROPPATCH, ACL
```

Dans cet exemple, la réponse OPTIONS indique que le serveur supporte le contrôle d'accès et que /foo.html peut avoir sa liste de contrôle d'accès modifiée par la méthode ACL.

7.3 MOVE

Lorsque une ressource est déplacée d'une localisation à une autre à cause d'une demande MOVE, les ACE non hérités et non protégés dans la propriété DAV:acl de la ressource NE DOIVENT PAS être modifiés, ou la demande MOVE échoue. Le traitement des ACE hérités et protégés est intentionnellement indéfini pour donner aux mises en œuvre de serveur de la souplesse pour appliquer l'héritage et la protection d'ACE.

7.4 COPY

La propriété DAV:acl sur la ressource à la destination d'un COPY DOIT être la même que si la ressource était créée par une demande individuelle de création de ressource (par exemple, MKCOL, PUT). Les clients qui souhaitent préserver la propriété DAV:acl dans une copie doivent lire la propriété DAV:acl avant de faire le COPY, puis effectuer une opération ACL sur la nouvelle ressource à la destination pour restaurer, pour autant que ce soit possible, la liste originale de contrôle d'accès.

7.5 LOCK

Un verrou sur une ressource assure que seul le possesseur du verrou peut modifier les ACE qui ne sont pas hérités ni protégés (ce sont les seuls ACE qu'un client puisse modifier avec une demande ACL). Un verrou ne protège pas les ACE hérités ou protégés, car un client ne peut pas les modifier avec une demande ACL sur cette ressource.

8. Méthodes de contrôle d'accès

8.1 ACL

La méthode ACL modifie la liste de contrôle d'accès (qui peut être lue via la propriété DAV:acl) d'une ressource. Précisément, la méthode ACL ne permet que la modification des ACE qui ne sont pas hérités, et non protégés. Une invocation de méthode ACL modifie tous les ACE non hérités et non protégés dans la liste de contrôle d'accès d'une ressource pour correspondre exactement aux ACE contenus au sein de l'élément XML DAV:acl (spécifié au paragraphe 5.5) du corps de la demande. Un corps de demande ACL DOIT contenir seulement un élément XML DAV:acl. Sauf si les ACE non hérités et non protégés de la propriété DAV:acl de la ressource peuvent être mis à jour pour avoir exactement la valeur spécifiée dans la demande ACL, la demande ACL DOIT échouer.

Il est possible que les ACE visibles pour l'utilisateur actuel dans la propriété DAV:acl ne soient qu'une portion de l'ensemble complet des ACE sur cette ressource. Si c'est le cas, une demande ACL ne modifie que l'ensemble d'ACE visibles à l'utilisateur actuel, et n'affecte aucun ACE non visible.

Afin d'éviter d'écraser les changements de DAV:acl par un autre client, un client DEVRAIT acquérir un verrou WebDAV sur la ressource avant de restituer la propriété DAV:acl d'une ressource qu'il a l'intention de mettre à jour.

Note de mise en œuvre : Deux opérations courantes sont d'ajouter ou retirer un ACE d'une liste de contrôle d'accès existante. Pour ce faire, un client utilise la méthode PROPFIND pour restituer la valeur de la propriété DAV:acl, puis il analyse la liste de contrôle d'accès retournée pour retirer tous les ACE hérités et protégés (ces ACE sont étiquetés avec les éléments XML DAV:inherited et DAV:protected). Dans l'ensemble restant d'ACE non hérités, non protégés, le client peut ajouter ou supprimer un ou plusieurs ACE avant de soumettre l'ensemble d'ACE final dans le corps de demande de la méthode ACL.

8.1.1 Préconditions d'ACL

Une mise en œuvre DOIT appliquer les contraintes suivantes sur une demande ACL. Si la contrainte est violée, une réponse 403 (Interdit) ou 409 (Conflit) DOIT être retournée et l'élément XML indiqué DOIT être retourné comme fils d'un élément DAV:error de niveau supérieur dans un corps de réponse XML.

Bien que ces éléments d'état soient généralement exprimés comme des éléments XML vides (et soient définis comme EMPTY dans le DTD) les mises en œuvre PEUVENT retourner des éléments descriptifs XML supplémentaires comme fils de l'élément d'état. Les clients DOIVENT être capables d'accepter les fils de ces éléments d'état. Les clients qui ne comprennent pas les éléments XML supplémentaires devraient les ignorer.

(DAV:no-ace-conflict) : les ACE soumis dans la demande ACL NE DOIVENT PAS être en conflit les uns avec les autres. C'est un code d'erreur fourre-tout qui indique qu'une restriction d'ACL spécifique d'une mise en œuvre a été violée.

(DAV:no-protected-ace-conflict) : les ACE soumis dans la demande ACL NE DOIVENT PAS être en conflit avec les ACE protégés sur la ressource. Par exemple, si la ressource a un ACE protégé qui accorde DAV:write à un certain principal, ce ne serait alors pas cohérent si la demande ACL soumettait un ACE refusant DAV:write au même principal.

(DAV:no-inherited-ace-conflict) : les ACE soumis dans la demande ACL NE DOIVENT PAS être en conflit avec les ACE hérités sur la ressource. Par exemple, si la ressource hérite un ACE de sa collection parente qui accorde DAV:write à un certain principal, il ne serait alors pas cohérent que la demande ACL soumette un ACE refusant DAV:write au même principal. Noter que rapporter cette erreur va dépendre de la mise en œuvre. Les mises en œuvre DOIVENT soit rapporter cette erreur, soit permettre l'établissement de l'ACE, et laisser ensuite les règles normales d'évaluation d'ACE déterminer si le nouvel ACE a un impact sur les privilèges disponibles à un certain principal.

(DAV:limited-number-of-aces) : le nombre d'ACE soumis dans la demande ACL NE DOIT PAS excéder le nombre d'ACE permis sur cette ressource. Cependant, les serveurs conformes à ACL DOIVENT accepter au moins un ACE accordant des privilèges sur un seul principal, et un ACE accordant des privilèges à un groupe.

(DAV:deny-before-grant) : tous les ACE de refus non hérités DOIVENT précéder tous les ACE d'octroi non hérités.

(DAV:grant-only) : les ACE soumis dans la demande ACL NE DOIVENT PAS inclure d'ACE de refus. Cette précondition ne s'applique que lorsque les restrictions d'ACL de la ressource incluent la contrainte DAV:grant-only (définie au paragraphe 5.6.1).

(DAV:no-invert) : la demande ACL NE DOIT PAS inclure d'élément DAV:invert. Cette précondition ne s'applique que lorsque la sémantique d'ACL de la ressource inclut la contrainte DAV:no-invert (définie au paragraphe 5.6.2).

(DAV:no-abstract) : la demande ACL NE DOIT PAS tenter d'accorder ou refuser un privilège abstrait (voir le paragraphe 5.3).

(DAV:not-supported-privilege) : les ACE soumis dans la demande ACL DOIVENT être pris en charge par la ressource.

(DAV:missing-required-principal) : le résultat de la demande ACL DOIT avoir au moins un ACE pour chaque principal identifié dans un élément XML DAV:required-principal dans la sémantique d'ACL de cette ressource (voir le paragraphe 5.5).

(DAV:recognized-principal) : Chaque URL principal dans la demande ACL DOIT identifier une ressource principale.

(DAV:allowed-principal) : les principaux spécifiés dans les ACE soumis dans la demande ACL DOIVENT être permis comme principaux pour la ressource. Par exemple, un serveur auquel seuls des principaux authentifiés peuvent accéder aux ressources ne permettrait pas que soient utilisés dans un ACE des principaux DAV:all ou DAV:unauthenticated, car cela permettrait un accès non authentifié aux ressources.

8.1.2 Exemple : méthode ACL

Dans l'exemple qui suit, l'utilisateur "fielding", authentifié par des informations de l'en-tête Authorization, accorde au principal identifié par l'URL http://www.example.com/users/esedlar (c'est-à-dire, l'utilisateur "esedlar") les privilèges de lecture et d'écriture, accorde au propriétaire de la ressource les privilèges read-acl et write-acl, et accorde à tous le privilège de lecture.

>> Demande <<

```
ACL /top/container/ HTTP/1.1
Host: www.example.com
Content-Type: text/xml; charset="utf-8"
```

```
Content-Length: xxxx
Authorization: Digest username="fielding",
 realm="users@example.com", nonce="...",
 uri="/top/container/", response="...", opaque="..."

<?xml version="1.0" encoding="utf-8" ?>
<D:acl xmlns:D="DAV:">
 <D:ace>
  <D:principal>
   <D:href>http://www.example.com/users/esedlar</D:href>
  </D:principal>
  <D:grant>
   <D:privilege><D:read/></D:privilege>
   <D:privilege><D:write/></D:privilege>
  </D:grant>
 </D:ace>
 <D:ace>
  <D:principal>
   <D:property><D:owner/></D:property>
  </D:principal>
  <D:grant>
   <D:privilege><D:read-acl/></D:privilege>
   <D:privilege><D:write-acl/></D:privilege>
  </D:grant>
 </D:ace>
 <D:ace>
  <D:principal><D:all/></D:principal>
  <D:grant>
   <D:privilege><D:read/></D:privilege>
  </D:grant>
 </D:ace>
</D:acl>

>> Réponse <<

HTTP/1.1 200 OK
```

8.1.3 Exemple : Échec de méthode ACL due à un conflit d'ACE protégé

Dans la demande qui suit, l'utilisateur "fielding", authentifié par les informations de l'en-tête Authorization, tente de refuser au principal identifié par l'URL http://www.example.com/users/esedlar (c'est-à-dire, l'usager "esedlar") les privilèges d'écriture. Avant la demande, la propriété DAV:acl sur la ressource contenait un ACE protégé (voir au paragraphe 5.5.3) accordant au DAV:owner les privilèges DAV:read et DAV:write. Le principal identifié par l'URL http://www.example.com/users/esedlar est le propriétaire de la ressource. L'invocation de la méthode ACL échoue parce que l'ACE soumis est en conflit avec l'ACE protégé, violant donc la sémantique de la protection d'ACE.

```
>> Demande <<

ACL /top/container/ HTTP/1.1
Host: www.example.com
Content-Type: text/xml; charset="utf-8"
Content-Length: xxxx
Authorization: Digest username="fielding",
 realm="users@example.com", nonce="...",
 uri="/top/container/", response="...", opaque="..."

<?xml version="1.0" encoding="utf-8" ?>
<D:acl xmlns:D="DAV:">
 <D:ace>
  <D:principal>
   <D:href>http://www.example.com/users/esedlar</D:href>
  </D:principal>
```

```
    <D:deny>
      <D:privilege><D:write/></D:privilege>
    </D:deny>
  </D:ace>
</D:acl>
```

>> Réponse <<

```
HTTP/1.1 403 Forbidden
Content-Type: text/xml; charset="utf-8"
Content-Length: xxx

<?xml version="1.0" encoding="utf-8" ?>
<D:error xmlns:D="DAV:">
  <D:no-protected-ace-conflict/>
</D:error>
```

8.1.4 Exemple : Échec de méthode ACL due à un conflit d'ACE hérité

Dans la demande suivante, l'utilisateur "ejw", authentifié par les information de l'en-tête Authorization, essaye de changer la liste de contrôle d'accès sur la ressource http://www.example.com/top/index.html. Cette ressource a deux ACE hérités.

L'ACE hérité n° 1 accorde au principal identifié par l'URL http://www.example.com/users/ejw (c'est-à-dire, l'usager "ejw") les privilèges http://www.example.com/privs/write-all et DAV:read-acl. Sur ce serveur, http://www.example.com/privs/write-all est un privilège agrégé qui contient DAV:write, et DAV:write-acl.

L'ACE hérité n° 2 accorde au principal DAV:all le privilège DAV:read.

La demande tente d'établir un ACE (non hérité) ACE, refusant au principal identifié par l'URL http://www.example.com/users/ejw (c'est-à-dire, l'usager "ejw") la permission DAV:write. Ceci est en conflit avec l'ACE hérité n° 1. Noter que la décision de rapporter un conflit d'ACE hérité est spécifique de la mise en œuvre de serveur. Une autre mise en œuvre de serveur pourrait permettre l'établissement du nouvel ACE, et utiliser ensuite les règles normales d'évaluation d'ACE pour déterminer si le nouvel ACE a un impact sur les privilèges disponibles pour un principal.

>> Demande <<

```
ACL /top/index.html HTTP/1.1
Host: www.example.com
Content-Type: text/xml; charset="utf-8"
Content-Length: xxxx
Authorization: Digest username="ejw",
  realm="users@example.com", nonce="...",
  uri="/top/index.html", response="...", opaque="..."

<?xml version="1.0" encoding="utf-8" ?>
<D:acl xmlns:D="DAV:" xmlns:F="http://www.example.com/privs/">
  <D:ace>
    <D:principal>
      <D:href>http://www.example.com/users/ejw</D:href>
    </D:principal>
    <D:grant><D:write/></D:grant>
  </D:ace>
</D:acl>
```

>> Réponse <<

```
HTTP/1.1 403 Forbidden
Content-Type: text/xml; charset="utf-8"
Content-Length: xxx

<?xml version="1.0" encoding="utf-8" ?>
<D:error xmlns:D="DAV:">
```

```
  <D:no-inherited-ace-conflict/>
</D:error>
```

8.1.5 Exemple : Échec de méthode ACL dû à une tentative d'établir un accord et un refus dans un seul ACE

Dans cet exemple, l'usager "ygoland", authentifié par les informations de l'en-tête Authorization, essaye de changer la liste de contrôle d'accès sur la ressource http://www.example.com/diamond/engagement-ring.gif. La demande ACL comporte un seul ACE syntaxiquement et sémantiquement incorrect, qui tente d'accorder au groupe identifié par l'URL http://www.example.com/users/friends le privilège DAV:read et de refuser au principal identifié par l'URL http://www.example.com/users/ygoland-so (c'est-à-dire, l'usager "ygoland-so") le privilège DAV:read. Cependant, il est illégal d'avoir plusieurs éléments principaux, ainsi qu'à la fois un octroi et un refus dans le même ACE, donc la demande échoue à cause de sa mauvaise syntaxe.

>> Demande <<

```
ACL /diamond/engagement-ring.gif HTTP/1.1
Host: www.example.com
Content-Type: text/xml; charset="utf-8"
Content-Length: xxxx
Authorization: Digest username="ygoland",
 realm="users@example.com", nonce="...",
 uri="/diamond/engagement-ring.gif", response="...",
 opaque="..."

<?xml version="1.0" encoding="utf-8" ?>
<D:acl xmlns:D="DAV:">
  <D:ace>
    <D:principal>
      <D:href>http://www.example.com/users/friends</D:href>
    </D:principal>
    <D:grant><D:read/></D:grant>
    <D:principal>
      <D:href>http://www.example.com/users/ygoland-so</D:href>
    </D:principal>
    <D:deny><D:read/></D:deny>
  </D:ace>
</D:acl>
```

>> Réponse <<

```
HTTP/1.1 400 Mauvaise demande
Content-Length: 0
```

Noter que si la demande avait été divisée en deux ACE, un pour accorder, et un pour refuser, la demande aurait été syntaxiquement bien formée.

9. Rapports de contrôle d'accès

9.1 Méthode REPORT

La méthode REPORT (définie au paragraphe 3.6 de la [RFC3253]) fournit un mécanisme extensible pour obtenir des informations sur une ressource. À la différence de la méthode PROPFIND, qui retourne la valeur d'une ou plusieurs propriétés désignées, la méthode REPORT peut impliquer des traitements plus complexes. REPORT est valable dans des cas où le serveur a accès à toutes les informations nécessaires pour effectuer une demande complexe (comme une interrogation) et où cela exigerait de multiples demandes du client pour restituer les informations nécessaires pour effectuer la même demande.

Un serveur qui prend en charge le protocole de contrôle d'accès WebDAV DOIT prendre en charge le rapport DAV:expand-property (défini au paragraphe 3.8 de la [RFC3253]).

9.2 Rapport DAV:acl-principal-prop-set

Le rapport DAV:acl-principal-prop-set retourne, pour tous les principaux dans la propriété DAV:acl (de l'URI de demande) qui sont identifiés par des URL http(s) ou par un principal DAV:property, la valeur des propriétés spécifiées dans le corps de la demande REPORT. Dans le cas où un URL principal apparaît plusieurs fois, le rapport DAV:acl-principal-prop-set DOIT retourner les propriétés pour ce principal une seule fois. La prise en charge de ce rapport est EXIGÉE.

Une utilisation prévue pour ce rapport est de restituer le nom lisible par l'homme (dans la propriété DAV:displayname) de chaque principal trouvé dans l'ACL. C'est utile pour construire des interfaces d'utilisateur qui montrent chaque ACE sous une forme lisible.

Conduite à tenir : Le corps de demande DOIT être un élément XML DAV:acl-principal-prop-set.

<!ELEMENT acl-principal-prop-set ANY>
Valeur de ANY : une séquence d'un ou plusieurs éléments, avec au plus un élément DAV:prop.
prop : voir le paragraphe 12.11 de la RFC2518

Ce rapport n'est défini que lorsque l'en-tête Depth a la valeur "0" ; les autres valeurs résultent en une réponse d'erreur 400 (Mauvaise demande). Noter que le paragraphe 3.6 de la [RFC3253] déclare que si l'en-tête Depth n'est pas présent, il prend la valeur de "0" par défaut.

Le corps de réponse pour une demande réussie DOIT être un élément XML DAV:multistatus (c'est-à-dire que la réponse utilise le même format que la réponse pour PROPFIND). Dans le cas où il n'y a pas d'élément de réponse, l'élément XML multistatus retourné est vide.

multistatus : voir le paragraphe 12.9 de la RFC 2518.

Le corps de réponse pour une demande réussie de rapport DAV:acl-principal-prop-set DOIT contenir un élément DAV:response pour chaque principal identifié par un URL http(s) de la liste d'un élément XML DAV:principal d'un ACE au sein de la propriété DAV:acl de la ressource identifiée par l'URI de demande.

Postconditions : (DAV:number-of-matches-within-limits) : Le nombre de principaux qui correspondent doit tomber dans les limites prédéfinies spécifiques du serveur. Par exemple, cette condition peut être déclanchée si une spécification de recherche devait causer le retour d'un très grand nombre de réponses.

9.2.1 Exemple : Rapport DAV:acl-principal-prop-set

La ressource http://www.example.com/index.html a une ACL avec trois ACE :

ACE n° 1 : tous les principaux (DAV:all) ont l'accès DAV:read et DAV:read-current-user-privilege-set.

ACE n° 2 : le principal identifié par http://www.example.com/people/gstein (l'usager "gstein") a reçu les privilèges DAV:write, DAV:write-acl, et DAV:read-acl.

ACE n° 3 : le groupe identifié par http://www.example.com/groups/authors (le groupe "authors") a reçu les privilèges DAV:write et DAV:read-acl.

L'exemple qui suit montre un rapport DAV:acl-principal-prop-set qui demande la propriété DAV:displayname. Il retourne la valeur de DAV:displayname pour les ressources http://www.example.com/people/gstein et http://www.example.com/groups/authors, mais pas pour DAV:all, car ce n'est pas un URL http(s).

>> Demande <<

```
REPORT /index.html HTTP/1.1
Host: www.example.com
Content-Type: text/xml; charset="utf-8"
Content-Length: xxxx
Depth: 0

<?xml version="1.0" encoding="utf-8" ?>
<D:acl-principal-prop-set xmlns:D="DAV:">
  <D:prop>
    <D:displayname/>
```

```
  </D:prop>
</D:acl-principal-prop-set>
>> Réponse <<

HTTP/1.1 207 Multi-Status
Content-Type: text/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<D:multistatus xmlns:D="DAV:">
  <D:response>
    <D:href>http://www.example.com/people/gstein</D:href>
    <D:propstat>
      <D:prop>
        <D:displayname>Greg Stein</D:displayname>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
  <D:response>
    <D:href>http://www.example.com/groups/authors</D:href>
    <D:propstat>
      <D:prop>
        <D:displayname>Site authors</D:displayname>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
</D:multistatus>
```

9.3 Rapport DAV:principal-match

Le rapport DAV:principal-match est utilisé pour identifier tous les membres (toute profondeur) de la collection identifiée par l'URI de demande qui sont des principaux et qui correspondent à l'utilisateur actuel. En particulier, si la collection contient des principaux, le rapport peut être utilisé pour identifier tous les membres de la collection qui correspondent à l'utilisateur actuel. Autrement, si la collection contient des ressources avec une propriété qui identifie un principal (par exemple, DAV:owner) le rapport peut être utilisé pour identifier tous les membres de la collection dont une propriété identifie un principal qui correspond à l'utilisateur actuel. Par exemple, ce rapport peut retourner toutes les ressources dans une hiérarchie de collections qui sont la propriété de l'utilisateur actuel. La prise en charge de ce rapport est EXIGÉE.

Conduite à tenir :

Le corps de demande DOIT être un élément XML DAV:principal-match.
<!ELEMENT principal-match ((principal-property | self), prop?)>
<!ELEMENT principal-property ANY>

Valeur ANY : un élément dont la valeur identifie une propriété. Ce qu'on attend est que la valeur de la propriété désignée contienne normalement un élément href qui contient l'URI d'un principal <!ELEMENT self EMPTY>

prop : voir le paragraphe 12.11 de la RFC2518.

Ce rapport n'est défini que lorsque l'en-tête Depth a la valeur "0" ; les autres valeurs résultent en une réponse d'erreur 400 (Mauvaise demande). Noter que le paragraphe 3.6 de la [RFC3253] déclare que si l'en-tête Depth n'est pas présent, sa valeur par défaut est de "0". Le corps de réponse pour une demande réussie DOIT être un élément XML DAV:multistatus. Dans le cas où il n'y a pas d'éléments de réponse, l'élément XML multistatus retourné est vide.

multistatus : voir le paragraphe 12.9 de la RFC 2518.

Le corps de réponse pour une demande réussie de rapport DAV:principal-match DOIT contenir un élément DAV:response pour chaque membre de la collection qui correspond à l'utilisateur actuel. Lorsque l'élément DAV:principal-property est utilisé, une correspondance se produit si l'utilisateur actuel correspond au principal identifié par l'URI trouvé dans l'élément DAV:href de la propriété identifiée par l'élément DAV:principal-property. Lorsque l'élément DAV:self est utilisé dans le rapport DAV:principal-

match produit contre un groupe, il correspond au groupe si un membre identifie le même principal que l'utilisateur actuel.

Si DAV:prop est spécifié dans le corps de demande, les propriétés spécifiées dans l'élément DAV:prop DOIVENT être rapportées dans les éléments DAV:response.

9.3.1 Exemple : Rapport DAV:principal-match

L'exemple suivant identifie les membres de la collection identifiée par l'URL http://www.example.com/doc qui sont la propriété de l'utilisateur actuel. L'utilisateur actuel ("gclemm") est authentifié en utilisant l'authentification par résumé.

>> Demande <<

```
REPORT /doc/ HTTP/1.1
Host: www.example.com
Authorization: Digest username="gclemm",
  realm="users@example.com", nonce="...",
  uri="/papers/", response="...", opaque="..."
Content-Type: text/xml; charset="utf-8"
Content-Length: xxxx
Depth: 0
<?xml version="1.0" encoding="utf-8" ?>
<D:principal-match xmlns:D="DAV:">
  <D:principal-property>
    <D:owner/>
  </D:principal-property>
</D:principal-match>
```

>> Réponse <<

```
HTTP/1.1 207 Multi-Status
Content-Type: text/xml; charset="utf-8"
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<D:multistatus xmlns:D="DAV:">
  <D:response>
    <D:href>http://www.example.com/doc/foo.html</D:href>
    <D:status>HTTP/1.1 200 OK</D:status>
  </D:response>
  <D:response>
    <D:href>http://www.example.com/doc/img/bar.gif</D:href>
    <D:status>HTTP/1.1 200 OK</D:status>
  </D:response>
</D:multistatus>
```

9.4 Rapport DAV:principal-property-search

Le rapport DAV:principal-property-search effectue une recherche de tous les principaux dont les propriétés contiennent des données de caractères qui correspondent aux critères de recherche spécifiés dans la demande. Une utilisation attendue de ce rapport est de découvrir l'URL d'un principal associé à une certaine personne ou groupe en les recherchant par noms. Cela est fait en cherchant sur DAV:displayname, qui est défini sur tous les principaux.

La méthode de recherche réelle (correspondance exacte contre correspondance de sous chaîne ou correspondance de préfixe, ou sensibilité à la casse) est délibérément laissée à la mise en œuvre de serveur pour permettre à l'utilisateur la mise en œuvre d'un large ensemble de systèmes de gestion possibles. Dans les cas où la mise en œuvre de DAV:principal-property-search n'est pas contrainte par la sémantique d'un répertoire sous-jacent de gestion d'utilisateur, la sémantique préférée par défaut est celle de correspondance de sous chaînes sans souci de la casse.

Pour l'efficacité de la mise en œuvre, les serveurs ne prennent normalement pas en charge la recherche sur toutes les propriétés. Une recherche qui demande des propriétés qui ne sont pas cherchables pour un certain principal ne va pas correspondre à ce principal. La prise en charge du rapport DAV:principal-property-search est EXIGÉE.

Note de mise en œuvre : la valeur d'une propriété WebDAV est une séquence de XML bien formés, et peut donc inclure tout caractère de la norme Unicode/ISO-10646, c'est-à-dire, la plupart des caractères connus dans les langages humains. Du fait des idiosyncrasies de transposition de casse à travers les langues humaines, la mise en œuvre de correspondance insensible à la casse n'est pas triviale. Les mises en œuvre de serveurs qui effectuent des correspondances de sous chaînes sont vivement encouragées à consulter la "Norme Unicode" [UNICODE4], en particulier le paragraphe 5.18, sous section "Correspondance sans casse", pour des lignes directrices sur la mise en œuvre d'algorithmes de correspondance insensibles à la casse.

Note de mise en œuvre : Certaines mises en œuvre de ce protocole vont utiliser un répertoire LDAP pour mémoriser des métadonnées de principaux. Le schéma qui décrit chaque attribut (autrement dit une propriété WebDAV) dans un répertoire LDAP spécifie si il prend en charge une recherche sensible ou insensible à la casse. Un des avantages de laisser la méthode de recherche à la discrétion de la mise en œuvre de serveur est que le comportement de recherche d'attribut LDAP par défaut peut être utilisé lors de la mise en œuvre du rapport DAV:principal-property-search.

Conduite à tenir : le corps de demande DOIT être un élément XML DAV:principal-property-search contenant une spécification de recherche et une liste facultative de propriétés. Pour chaque principal qui correspond à la spécification de recherche, la réponse va contenir la valeur des propriétés demandées sur ce principal.

<!ELEMENT principal-property-search ((property-search+), prop?, apply-to-principal-collection-set?) >

Par défaut, le rapport cherche tous les membres (à toutes les profondeurs) de la collection identifiée par l'URI de demande. Si DAV:apply-to-principal-collection-set est spécifié dans le corps de demande, la demande est appliquée au lieu de chaque collection identifiée par la propriété DAV:principal-collection-set de la ressource identifiée par l'URI de demande.

L'élément DAV:property-search contient un élément prop qui énumère les propriétés à rechercher et un élément match, qui contient la chaîne de recherche.

<!ELEMENT property-search (prop, match) >
prop : voir le paragraphe 12.11 de la RFC2518.
<!ELEMENT match #PCDATA >

Plusieurs éléments property-search ou plusieurs éléments au sein de l'élément DAV:prop seront interprétés avec un ET logique.

Ce rapport n'est défini que lorsque l'en-tête Depth a la valeur "0" ; les autres valeurs résultent en une réponse d'erreur 400 (Mauvaise demande). Noter que le paragraphe 3.6 de la [RFC3253] déclare que si l'en-tête Depth n'est pas présent, il prend par défaut une valeur de "0".

Le corps de réponse pour une demande réussie DOIT être un élément XML DAV:multistatus. Lorsque il n'y a pas d'élément de réponse, l'élément XML multistatus retourné est vide.

multistatus : voir le paragraphe 12.9 de la RFC2518.

Le corps de réponse pour une demande réussie de rapport DAV:principal-property-search DOIT contenir un élément DAV:response pour chaque principal dont les valeurs de propriété satisfont la spécification de recherches donnée dans DAV:principal-property-search.

Si DAV:prop est spécifié dans le corps de demande, les propriétés spécifiées dans l'élément DAV:prop DOIVENT être rapportées dans les éléments DAV:response.

Préconditions : aucune

Postconditions : (DAV:number-of-matches-within-limits) : le nombre de principaux qui correspondent doit tomber dans les limites prédéfinies spécifiques du serveur. Par exemple, cette condition peut être déclanchée si une spécification de recherche pourrait causer le retour d'un nombre extrêmement grand de réponses.

9.4.1 Confrontation

Plusieurs cas sont à considérer lorsque on confronte des chaînes. Le cas le plus facile est lorsque une valeur de propriété est "simple" et a seulement pour contenu des éléments d'information de caractères (voir [XML-INFOSET]). Par exemple, la chaîne de recherche "julian" va faire correspondre la propriété DAV:displayname avec la valeur "Julian Reschke". Noter que la conduite à tenir sur le réseau pour DAV:displayname dans ce cas est :

<D:displayname xmlns:D="DAV:">Julian Reschke</D:displayname>

Le nom de la propriété est codé en élément d'information d'élément XML et le contenu de l'élément d'information de caractère de la propriété est "Julian Reschke".

Un cas plus compliqué se présente lorsque les propriétés ont un contenu mixte (c'est-à-dire, des valeurs composites consistant en plusieurs morceaux d'éléments fils, des morceaux d'autres types d'information, et du contenu d'éléments d'informations de caractères). On considère que pour la propriété "aprop" dans l'espace de noms "http://www.example.com/props/", la conduite à tenir est :

<W:aprop xmlns:W="http://www.example.com/props/"> {cdata 0}<W:elem1>{cdata 1}</W:elem1>
<W:elem2>{cdata 2}</W:elem2>{cdata 3} </W:aprop>

Dans ce cas, la confrontation est effectuée sur chaque séquence individuelle contiguë d'éléments d'information de caractères. Dans l'exemple ci-dessus, une chaîne de recherche serait comparée aux quatre chaînes suivantes :
{cdata 0}
{cdata 1}
{cdata 2}
{cdata 3}

C'est-à-dire que quatre confrontations individuelles seraient effectuées, une pour chacun de {cdata 0}, {cdata 1}, {cdata 2}, et {cdata 3}.

9.4.2 Exemple : Rapport réussi DAV:principal-property-search

Dans cet exemple, le client demande les URL de principal de tous les utilisateurs dont la propriété DAV:displayname contient la sous chaîne "doE" et dont la propriété "titre" dans l'espace de noms "http://BigCorp.com/ns/" (c'est-à-dire, leur titre professionnel) contient "Ventes". De plus, le client demande que cinq propriétés soient retournées avec les principaux correspondants :
Dans DAV: namespace : displayname
Dans l'espace de noms http://www.example.com/ns/ : département, téléphone, bureau, salaire

La réponse montre que deux ressources de principal satisfont à la spécification de recherche, "John Doe" et "Zygdoebert Smith". Les propriétés "salaire" dans l'espace de noms "http://www.example.com/ns/" ne sont pas retournées, car le principal qui fait la demande n'a pas les permissions d'accès suffisantes pour lire cette propriété.

>> Demande <<

REPORT /users/ HTTP/1.1
Host: www.example.com
Content-Type: text/xml; charset=utf-8
Content-Length: xxxx
Depth: 0

```
<?xml version="1.0" encoding="utf-8" ?>
<D:principal-property-search xmlns:D="DAV:">
  <D:property-search>
   <D:prop>
     <D:displayname/>
   </D:prop>
   <D:match>doE</D:match>
  </D:property-search>
  <D:property-search>
   <D:prop xmlns:B="http://www.example.com/ns/">
     <B:titre/>
      </D:prop>
      <D:match>Ventes</D:match>
    </D:property-search>
  <D:prop xmlns:B="http://www.example.com/ns/">
   <D:displayname/>
   <B:département/>
   <B:téléphone/>
```

```
    <B:bureau/>
    <B:salaire/>
  </D:prop>
</D:principal-property-search>

>> Réponse <<

HTTP/1.1 207 Multi-Status
Content-Type: text/xml; charset=utf-8
Content-Length: xxxx

<?xml version="1.0" encoding="utf-8" ?>
<D:multistatus xmlns:D="DAV:" xmlns:B="http://BigCorp.com/ns/">
  <D:response>
    <D:href>http://www.example.com/users/jdoe</D:href>
    <D:propstat>
      <D:prop>
        <D:displayname>John Doe</D:displayname>
        <B:département>Ventes d'accessoires logiciels</B:département>
        <B:téléphone>234-4567</B:téléphone>
        <B:bureau>209</B:bureau>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
    <D:propstat>
      <D:prop>
        <B:salaire/>
      </D:prop>
      <D:status>HTTP/1.1 403 Interdit</D:status>
    </D:propstat>
  </D:response>
  <D:response>
    <D:href>http://www.example.com/users/zsmith</D:href>
    <D:propstat>
      <D:prop>
       <D:displayname>Zygdoebert Smith</D:displayname>
        <B:département>Ventes d'accessoires</B:département>
        <B:téléphone>234-7654</B:téléphone>
        <B:bureau>114</B:bureau>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
    <D:propstat>
      <D:prop>
        <B:salaire/>
      </D:prop>
      <D:status>HTTP/1.1 403 Interdit</D:status>
    </D:propstat>
  </D:response>
</D:multistatus>
```

9.5 Rapport DAV:principal-search-property-set

Le rapport DAV:principal-search-property-set identifie les propriétés qui peuvent être cherchées en utilisant le rapport DAV:principal-property-search (défini au paragraphe 9.4).

Les serveurs DOIVENT prendre en charge le rapport DAV:principal-search-property-set sur toutes les collections identifiées dans la valeur de la propriété DAV:principal-collection-set.

Un agent d'utilisateur de protocole de contrôle d'accès pourrait utiliser le résultat du rapport DAV:principal-search-property-set pour présenter une interface d'interrogation à l'utilisateur pour restituer les principaux.

La prise en charge de ce rapport est EXIGÉE.

Note de mise en œuvre : Certains clients vont avoir seulement un espace d'écran limité pour l'affichage des listes de propriétés cherchables. Dans ce cas, un utilisateur peut apprécier d'avoir les propriétés les plus fréquemment recherchées affichées à l'écran, plutôt que d'avoir à dérouler une longue liste de propriétés cherchables. Un mécanisme pour signaler les propriétés les plus fréquemment recherchées est de les retourner au début d'une liste de propriétés. Un client peut alors afficher de préférence la liste des propriétés dans l'ordre, augmentant la probabilité que les propriétés les plus fréquemment recherchées apparaissent à l'écran, et n'exigent pas de déroulement pour leur sélection.

Conduite à tenir : Le corps de demande DOIT être un élément XML DAV:principal-search-property-set.

Ce rapport n'est défini que lorsque l'en-tête Depth a la valeur "0" ; d'autres valeurs résultent en une réponse d'erreur 400 (Mauvaise demande). Noter que le paragraphe 3.6 de la [RFC3253] déclare que si l'en-tête Depth n'est pas présent, il prend par défaut la valeur de "0".

Le corps de réponse DOIT être un élément XML DAV:principal-search-property-set, contenant un élément XML DAV:principal-search-property pour chaque propriété qui peut être cherchée avec le rapport DAV:principal-property-search. Un serveur PEUT limiter sa réponse à juste un sous ensemble des propriétés cherchables, comme celles qui seront vraisemblablement utiles pour un client de contrôle d'accès interactif.

<!ELEMENT principal-search-property-set (principal-search-property*) >

Chaque élément XML DAV:principal-search-property contient exactement une propriété cherchable, et une description de la propriété.

<!ELEMENT principal-search-property (prop, description) >

L'élément DAV:prop contient une propriété principale sur laquelle le serveur est capable d'effectuer un rapport DAV:principal-property-search.

prop : voir le paragraphe 12.11 de la RFC2518.

L'élément de description est une description lisible par l'homme des informations que cette propriété représente. Les serveurs DOIVENT indiquer le langage de la description en utilisant l'attribut xml:lang et DEVRAIENT considérer l'en-tête de demande HTTP Accept-Language lors du choix parmi plusieurs langues disponibles.

<!ELEMENT description #PCDATA >

9.5.1 Exemple : Rapport DAV:principal-search-property-set

Dans cet exemple, le client détermine l'ensemble des propriétés principales cherchables en demandant le rapport DAV:principal-search-property-set sur la racine de l'ensemble de collection d'URL de principaux du serveur, identifiés par http://www.example.com/users/.

>> Demande <<

```
REPORT /users/ HTTP/1.1
Host: www.example.com
Content-Type: text/xml; charset="utf-8"
Content-Length: xxx
Accept-Language: en, de
Authorization: BASIC d2FubmFtYWs6cGFzc3dvcmQ=
Depth: 0

<?xml version="1.0" encoding="utf-8" ?>
<D:principal-search-property-set xmlns:D="DAV:"/>
```

>> Réponse <<

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
```

```
Content-Length: xxx

<?xml version="1.0" encoding="utf-8" ?>
<D:principal-search-property-set xmlns:D="DAV:">
  <D:principal-search-property>
    <D:prop>
      <D:displayname/>
    </D:prop>
    <D:description xml:lang="en">Full name</D:description>
  </D:principal-search-property>
  <D:principal-search-property>
    <D:prop xmlns:B="http://BigCorp.com/ns/">
      <B:title/>
    </D:prop>
    <D:description xml:lang="en">Job title</D:description>  </D:principal-search-property>
</D:principal-search-property-set>
```

10. Traitement XML

Les mises en œuvre de la présente spécification DOIVENT prendre en charge la règle "ignore" d'élément XML, comme spécifiée au paragraphe 23.3.2 de la [RFC2518], et la recommandation sur l'espace de noms XML [XML-NAMES].

Noter que l'utilisation de l'espace de noms DAV est réservé aux éléments XML et aux noms de propriété définis dans une RFC en cours de normalisation ou expérimentale de l'IETF.

11. Considérations d'internationalisation

Dans la présente spécification, le seul contenu lisible par l'homme se trouve dans l'élément XML de description qu'on trouve dans la propriété DAV:supported-privilege-set. Cet élément contient une description lisible par l'homme des capacités contrôlées par un privilège. Par suite, l'élément de description doit être capable de représenter les descriptions dans plusieurs jeux de caractères. Comme l'élément de description se trouve dans une propriété WebDAV, il est représenté sur le réseau comme XML [REC-XML], et donc peut démultiplier les capacités de codage des étiquettes de langage et de jeu de caractères de XML. Spécifiquement, les processeurs XML doivent au minimum être capables de lire les éléments XML codés en utilisant le codage UTF-8 [RFC3629] du plan multilingue de la norme ISO 10646. Les exemples XML de la présente spécification montrent l'utilisation du paramètre charset de l'en-tête Content-Type, comme défini dans la [RFC3023], ainsi que de l'attribut XML "encoding", qui ensemble fournissent les informations d'identification de jeu de caractères pour les processeurs MIME et XML. De plus, la présente spécification exige que les mises en œuvre de serveur étiquettent les champs de description avec l'attribut xml:lang (voir le paragraphe 2.12 de la [REC-XML]) qui spécifie le langage humain de la description. De plus, les mises en œuvre de serveur devraient prendre en compte la valeur de l'en-tête Accept-Language HTTP pour déterminer quelle chaîne de description retourner.

Pour les éléments XML autres que l'élément de description, il est prévu que les mises en œuvre vont traiter les noms de propriété, les noms de privilège, et les valeurs comme des jetons, et convertir ces jetons en texte lisible par l'homme dans le langage et le jeu de caractères de l'utilisateur lors de l'affichage à une personne. Seul un utilitaire générique d'affichage de propriétés WebDAV afficherait ces valeurs dans leur forme brute à un utilisateur humain.

Pour le rapport des erreurs, on suit la convention des codes d'état HTTP/1.1, incluant avec chaque code d'état une brève description en anglais du code (par exemple, 200 (OK)). Bien qu'il existe une possibilité qu'un agent d'utilisateur peu élaboré affiche ce message à un utilisateur, les applications internationalisées vont ignorer ce message, et afficher un message approprié dans la langue et le jeu de caractères de l'utilisateur.

D'autres considérations d'internationalisation pour ce protocole sont décrites dans la spécification du protocole de collecte ordonnée des auteurs distribuée sur la Toile WebDAV [RFC2518].

12. Considérations sur la sécurité

Les applications et les utilisateurs de ce protocole de contrôle d'accès devraient être conscients de plusieurs considérations de sécurité, détaillées ci-dessous. En plus de la discussion du présent document, les considérations de sécurité détaillées dans la spécification HTTP/1.1 [RFC2616], la spécification du protocole de collecte ordonnée des auteurs distribuée sur la Toile

WebDAV [RFC2518], et la spécification des types de supports XML [RFC3023] devraient être considérées dans une analyse de la sécurité de ce protocole.

12.1 Risque accru d'utilisateurs compromis

En l'absence d'un mécanisme pour manipuler à distance les listes de contrôle d'accès, si des accréditifs d'authentification d'un seul utilisateur sont compromis, seules les ressources pour lesquelles l'utilisateur a une permission d'accès peuvent être lues, modifiées, déplacées ou supprimées. Avec l'introduction du présent protocole de contrôle d'accès, si un seul utilisateur compromis a la capacité de changer les ACL pour une large gamme d'autres utilisateurs (par exemple, un super utilisateur) le nombre de ressources qui pourraient être altérées par un seul utilisateur compromis augmente. Ce risque peut être atténué en limitant le nombre de personnes qui ont les privilèges write-acl sur une large gamme de ressources.

12.2 Risques des privilèges DAV:read-acl et DAV:current-user-privilege-set

La capacité de lire les privilèges d'accès (mémorisés dans la propriété DAV:acl) ou les privilèges permis à l'utilisateur actuellement authentifié (mémorisés dans la propriété DAV:current-user-privilege-set) sur une ressource peut sembler inoffensive, car lire une ACL ne peut pas affecter l'état de la ressource. Cependant, si toutes les ressources ont des ACL qui peuvent être lues par tout le monde, il est possible d'effectuer une recherche exhaustive des ressources qui se sont elles-mêmes laissées par inadvertance dans un état vulnérable, comme d'être en écriture pour tout le monde. La méthode de restitution de propriété PROPFIND, exécutée avec une profondeur infinie sur une hiérarchie entière, est un moyen très efficace pour restituer les propriétés DAV:acl ou DAV:current-user-privilege-set. Une fois trouvée, cette vulnérabilité peut être exploitée par une attaque de déni de service dans laquelle la ressource ouverte est réécrite de façon répétée. Autrement, les ressources accessibles en écriture peuvent être modifiées d'une façon indésirable.

Pour réduire ce risque, les privilèges read-acl ne devraient pas être accordés à des principaux non authentifiés, et des restrictions sur les privilèges read-acl et read-current-user-privilege-set pour les principaux authentifiés devraient être analysés avec soin lors du déploiement de ce protocole. L'accès à la propriété current-user-privilege-set va impliquer un compromis entre facilité d'utilisation et sécurité. Lorsque le current-user-privilege-set est visible, les interfaces d'utilisateur sont supposées fournir des informations améliorées concernant les opérations permises et interdites, et donc ces informations peuvent aussi indiquer une vulnérabilité qui pourrait être exploitée. Le déploiement de ce protocole devra évaluer ce compromis à la lumière des exigences de l'environnement de déploiement.

12.3 Pas de connaissance préalable de l'ACL initiale

Dans un effort pour réduire la complexité du protocole, la présente spécification ne traite intentionnellement pas la question de la façon de gérer ou découvrir l'ACL initiale qui est placée sur une ressource à sa création. La seule façon de découvrir l'ACL initiale est de créer une nouvelle ressource, puis de restituer la valeur de la propriété DAV:acl. Cela suppose que le principal qui crée la ressource a aussi reçu le privilège DAV:read-acl.

Par suite, il est possible qu'un principal puisse créer une ressource, et ensuite découvre que son ACL accorde des privilèges qui sont indésirables. De plus, le présent protocole rend possible (quoique improbable) que le principal créateur soit dans l'incapacité de modifier l'ACL, ou même de supprimer la ressource. Même lorsque l'ACL peut être modifiée, il y aura une brève période pendant laquelle la ressource va exister avec l'ACL initiale avant que sa nouvelle ACL puisse être établie.

Plusieurs facteurs atténuent ce risque. Les principaux humains qui sont conscients des permissions d'accès par défaut dans leurs environnements d'édition les prennent en compte lorsque ils écrivent des informations. De plus, les politiques de privilège par défaut sont généralement très prudentes, et limitent les privilèges accordés par l'ACL initiale.

13. Authentification

Les mécanismes d'authentification définis pour être utilisés avec HTTP et WebDAV s'appliquent aussi à ce protocole de contrôle d'accès WebDAV, en particulier les mécanismes d'authentification Basic et Digest définis dans la [RFC2617]. La mise en œuvre de la spécification de l'ACL exige que l'authentification de base, si elle est utilisée, DOIT être prise en charge seulement sur un transport sécurisé tel que TLS.

14. Considérations relatives à l'IANA

Le présent document utilise l'espace de noms défini par la [RFC2518] pour les éléments XML. C'est-à-dire que la présente spécification utilise l'espace de noms d'URI "DAV:", précédemment enregistré dans le registre des schémas d'URI. Toutes les

autres considérations relatives à l'IANA mentionnées dans la [RFC2518] sont aussi applicables à la présente spécification.

15. Remerciements

Le présent protocole est le produit de la collaboration de l'équipe de conception d'ACL de WebDAV : Bernard Chester, Geoff Clemm, Anne Hopkins, Barry Lind, Sean Lyndersay, Eric Sedlar, Greg Stein, et Jim Whitehead. Les auteurs expriment leur reconnaissance pour leur relecture détaillée et leurs commentaires à Jim Amsden, Dylan Barrell, Gino Basso, Murthy Chintalapati, Lisa Dusseault, Stefan Eissing, Tim Ellison, Yaron Goland, Dennis Hamilton, Laurie Harper, Eckehard Hermann, Ron Jacobs, Chris Knight, Remy Maucherat, Larry Masinter, Joe Orton, Peter Raymond, et Keith Wannamaker. Merci à Keith Wannamaker pour l'essai initial des paragraphes de recherche de propriétés de principal.

Des travaux avaient été effectués antérieurement sur les protocoles de contrôle d'accès de WebDAV par Yaron Goland, Paul Leach, Lisa Dusseault, Howard Palmer, et Jon Radoff. Nous tenons à remercier les auteurs des protocoles DeltaV, WebDAV et HTTP qui ont posé les fondations sur lesquelles le présent protocole s'appuie, et le groupe de travail WebDAV pour les retours fournis.

16. Références

16.1 Références normatives

[REC-XML] Bray, T., Paoli, J., Sperberg-McQueen, C. and E. Maler, "Extensible Markup Language (XML) 1.0 (Third ed)", Recommandation W3C REC-xml-20040204, février 2004, < http://www.w3.org/TR/2004/REC-xml-20040204 >.

[XML-INFOSET] Cowan, J. and R. Tobin, "XML Information Set (Second Edition)", Recommandation W3C REC-xml-infoset-20040204, février 2004, < http://www.w3.org/TR/2004/REC-xml-infoset-20040204/ >.

[XML-NAMES] Bray, T., Hollander, D. and A. Layman, "Namespaces in XML", Recommandation W3C REC-xml-names-19990114, janvier 1999, < http://www.w3.org/TR/1999/REC-xml-names-19990114 >.

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.

[RFC2518] Y. Goland, E. Whitehead, A. Faizi, S. Carter et D. Jensen, "Extensions HTTP pour la création répartie -- WEBDAV", février 1999. *(Obsolète, voir la* RFC4918)

[RFC2616] R. Fielding et autres, "Protocole de transfert hypertexte -- HTTP/1.1", juin 1999. *(D.S., MàJ par* 2817, 6585)

[RFC2617] J. Franks et autres, "Authentification HTTP : Authentification d'accès de base et par résumé", juin 1999. *(DS.)*

[RFC3023] M. Murata, S. St.Laurent et D. Kohn, "Types de support XML", janvier 2001. (*Obsolète, voir* RFC7303)

[RFC3253] G. Clemm et autres, "Extensions de versions à WebDAV (Protocole de collecte ordonnée des auteurs et des versions distribuée sur la Toile)", mars 2002. *(P.S.)*

[RFC3530] S. Shepler et autres, "Protocole de système de fichiers réseau (NFS) version 4", avril 2003. (*Obsolète, voir* RFC7530) *(P.S.)*

[RFC3629] F. Yergeau, "UTF-8, un format de transformation de la norme ISO 10646", STD 63, novembre 2003.

16.2 Références pour information

[RFC2251] M. Wahl, T. Howes et S. Kille, "Protocole léger d'accès à un répertoire (v3)", décembre 1997.

[RFC2255] T. Howes, M. Smith, "Format d'URL LDAP", décembre 1997. (*Obsolète, voir* RFC4510, RFC4516) *(P.S.)*

[UNICODE4] The Unicode Consortium, "The Unicode Standard - Version 4.0", Addison-Wesley , août 2003, <http://www.unicode.org/versions/Unicode4.0.0/>. ISBN 0321185781.

Appendice A. Addendum à la définition de type de document XML WebDAV

Tous les éléments XML définis dans la présente définition de type de document (DTD, *Document Type Definition*) appartiennent à l'espace de noms DAV. Le présent DTD devrait être vu comme un addendum au DTD fourni au paragraphe 23.1 de la [RFC2518].

```
<!-- Privilèges -- (Section 3)>
  <!ELEMENT read EMPTY>
  <!ELEMENT write EMPTY>
  <!ELEMENT write-properties EMPTY>
  <!ELEMENT write-content EMPTY>
  <!ELEMENT unlock EMPTY>
  <!ELEMENT read-acl EMPTY>
  <!ELEMENT read-current-user-privilege-set EMPTY>
  <!ELEMENT write-acl EMPTY>
  <!ELEMENT bind EMPTY>
  <!ELEMENT unbind EMPTY>
  <!ELEMENT all EMPTY>

<!-- Propriétés de principal (Section 4) -->
  <!ELEMENT principal EMPTY>
  <!ELEMENT alternate-URI-set (href*)>
  <!ELEMENT principal-URL (href)>
  <!ELEMENT group-member-set (href*)>
  <!ELEMENT group-membership (href*)>

<!-- Propriétés de contrôle d'accès (Section 5) -->
<!-- Propriété DAV:owner (paragraphe 5.1) -->
  <!ELEMENT owner (href?)>

<!-- Propriété DAV:group (paragraphe 5.2) -->
  <!ELEMENT group (href?)>

<!-- Propriété DAV:supported-privilege-set (paragraphe 5.3) -->
  <!ELEMENT supported-privilege-set (supported-privilege*)>
  <!ELEMENT supported-privilege (privilege, abstract?, description, supported-privilege*)>
  <!ELEMENT privilege ANY>
  <!ELEMENT abstract EMPTY>
  <!ELEMENT description #PCDATA>

<!-- Propriété DAV:current-user-privilege-set (paragraphe 5.4) -->
  <!ELEMENT current-user-privilege-set (privilege*)>

<!-- Propriétés DAV:acl (paragraphe 5.5) -->
  <!ELEMENT acl (ace)* >
  <!ELEMENT ace ((principal | invert), (grant|deny), protected?, inherited?)>
  <!ELEMENT principal (href) | all | authenticated | unauthenticated | property | self)>
  <!ELEMENT all EMPTY>
  <!ELEMENT authenticated EMPTY>
  <!ELEMENT unauthenticated EMPTY>
  <!ELEMENT property ANY>
  <!ELEMENT self EMPTY>
  <!ELEMENT invert principal>
  <!ELEMENT grant (privilege+)>
  <!ELEMENT deny (privilege+)>
  <!ELEMENT privilege ANY>
  <!ELEMENT protected EMPTY>
  <!ELEMENT inherited (href)>

<!-- Propriété DAV:acl-restrictions (paragraphe 5.6) -->
  <!ELEMENT acl-restrictions (grant-only?, no-invert?, deny-before-grant?, required-principal?)>
  <!ELEMENT grant-only EMPTY>
```

```
  <!ELEMENT no-invert EMPTY>
  <!ELEMENT deny-before-grant EMPTY>
  <!ELEMENT required-principal (all? | authenticated? | unauthenticated? | self? | href* |property*)>

<!—Propriété DAV:inherited-acl-set (paragraphe 5.7) -->
  <!ELEMENT inherited-acl-set (href*)>

<!-- Propriété DAV:principal-collection-set (paragraphe 5.8) -->
  <!ELEMENT principal-collection-set (href*)>

<!-- Contrôle d'accès et méthodes existantes (Section 7) -->
  <!ELEMENT need-privileges (resource)* >
  <!ELEMENT resource ( href, privilege )

<!-- Préconditions de méthode ACL (paragraphe 8.1.1) -->
  <!ELEMENT no-ace-conflict EMPTY>
  <!ELEMENT no-protected-ace-conflict EMPTY>
  <!ELEMENT no-inherited-ace-conflict EMPTY>
  <!ELEMENT limited-number-of-aces EMPTY>
  <!ELEMENT grant-only EMPTY>
  <!ELEMENT no-invert EMPTY>
  <!ELEMENT deny-before-grant EMPTY>
  <!ELEMENT no-abstract EMPTY>
  <!ELEMENT not-supported-privilege EMPTY>
  <!ELEMENT missing-required-principal EMPTY>
  <!ELEMENT recognized-principal EMPTY>
  <!ELEMENT allowed-principal EMPTY>

<!-- REPORT (Section 9) -->
  <!ELEMENT acl-principal-prop-set ANY>
```

Valeur ANY : séquence d'un ou plusieurs éléments, avec au plus un élément DAV:prop.

```
  <!ELEMENT principal-match ((principal-property | self), prop?)>
  <!ELEMENT principal-property ANY>
```

Valeur ANY : un élément dont la valeur identifie une propriété. On s'attend à ce que la valeur de la propriété nommée contienne normalement un élément href qui contient l'URI d'un principal.

```
  <!ELEMENT self EMPTY>
  <!ELEMENT principal-property-search ((property-search+), prop?) >
  <!ELEMENT property-search (prop, match) >
  <!ELEMENT match #PCDATA >
  <!ELEMENT principal-search-property-set ( principal-search-property*) >
  <!ELEMENT principal-search-property (prop, description) >
  <!ELEMENT description #PCDATA >
```

Appendice B. Tableau des privilèges des méthodes WebDAV (normatif)

Le tableau des méthodes de WebDAV suivant (tel que défini dans les RFC 2518, 2616, et 3253) précise quels privilèges sont requis pour l'accès pour chaque méthode. Noter que les privilèges cités, s'ils sont refusés, DOIVENT causer le refus d'accès. Cependant, étant donné qu'une mise en œuvre spécifique PEUT définir un privilege personnalisé supplémentaire pour contrôler l'accès aux méthodes existantes, avoir tous les privilèges indiqués ne signifie pas que l'accès sera accordé. Noter que le manque des privilèges indiqués n'implique pas que l'accès sera refusé, car une mise en œuvre particulière peut utiliser un sous privilège agrégé sous le privilège indiqué pour contrôler l'accès. Les privilèges requis se réfèrent à la ressource en cours de traitement sauf spécification contraire.

Méthode	**Privilèges**
GET	<D:read>
HEAD	<D:read>
OPTIONS	<D:read>
PUT (la cible existe)	<D:write-content> sur la ressource cible
PUT (il n'existe pas de cible)	<D:bind> sur la collection parente de la cible
PROPPATCH	<D:write-properties>
ACL	<D:write-acl>

PROPFIND	<D:read> (plus <D:read-acl> et <D:read-current-user-privilege-set> comme nécessaire)
COPY (la cible existe)	<D:read>, <D:write-content> et <D:write-properties> sur la ressource cible
COPY (il n'existe pas de cible)	<D:read>, <D:bind> sur la collection cible
MOVE (il n'existe pas de cible)	<D:unbind> sur la collection source et <D:bind> sur la collection cible
MOVE (la cible existe)	Comme ci-dessus, plus <D:unbind> sur la collection cible
DELETE	<D:unbind> sur la collection parente
LOCK (la cible existe)	<D:write-content>
LOCK (il n'existe pas de cible)	<D:bind> sur la collection parente
MKCOL	<D:bind> sur la collection parente
UNLOCK	<D:unlock>
CHECKOUT	<D:write-properties>
CHECKIN	<D:write-properties>
REPORT	<D:read> (sur toutes les ressources référencées)
VERSION-CONTROL	<D:write-properties>
MERGE	<D:write-content>
MKWORKSPACE	<D:write-content> sur la collection parente
BASELINE-CONTROL	<D:write-properties> et <D:write-content>
MKACTIVITY	<D:write-content> sur la collection parente

Index

Adresse des auteurs

Geoffrey Clemm
IBM
20 Maguire Road
Lexington, MA 02421
USA
mél : geoffrey.clemm@us.ibm.com

Julian F. Reschke
greenbytes GmbH
Salzmannstrasse 152
Muenster, NW 48159
Germany
mél : julian.reschke@greenbytes.de

Eric Sedlar
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065
USA
mél : eric.sedlar@oracle.com

Jim Whitehead
U.C. Santa Cruz, Dept. of Computer Science
1156 High Street
Santa Cruz, CA 95064
USA
mél : ejw@cse.ucsc.edu

Déclaration complète de droits de reproduction

Remerciement
Le financement de la fonction d'édition des RFC est actuellement fourni par l'Internet Society.