

本文的英文原文来自 [XEP-0045](#)

XEP-0045: 多用户聊天

摘要: 本文定义了一个 XMPP 协议扩展用于多用户文本会议. 即多个 XMPP 可以在一个房间或频道互相交流信息, 类似互联网中继聊天系统 (IRC). 还有标准聊天室功能如聊天室的主题和邀请, 本协议定义了一个强有力的房间控制模型, 包括能够踢和禁止用户, 任命主持人和管理员, 要求会员或密码才能加入房间, 等等。

作者: Peter Saint-Andre

XMPP 扩展协议的版权(1999-2008)归 XMPP 标准化基金会 (XSF) 所有

版权: © 1999 - 2010 XMPP 标准化基金会 (XSF). 参见[法律通告](#).

状态: 草案

类型: 标准跟踪

版本: 1.24

最后更新日期: 2008-07-16

注意: 这里定义的协议是 XMPP 标准化基金会有一个草案标准. 对本协议的执行是被鼓励的, 也适于部署到生产系统, 但是在它成为最终标准之前可能还会有一些变动.

目录

- [1 绪论](#)
- [2 范围](#)
- [3 需求](#)
- [4 术语](#)
 - [4.1 通用术语](#)
 - [4.2 房间类型](#)
 - [4.3 登场人物](#)
- [5 角色\(Roles\), 岗位\(Affiliations\)和权限\(Privileges\)](#)
 - [5.1 角色](#)
 - [5.1.1 权限](#)
 - [5.1.2 默认角色](#)
 - [5.1.3 变更角色](#)
 - [5.2 岗位](#)
 - [5.2.1 权限](#)
 - [5.2.2 变更岗位](#)

- [6 实体用例](#)
 - [6.1 MUC 的发现组件支持](#)
 - [6.2 发现房间](#)
 - [6.3 查询房间信息](#)
 - [6.4 查询房间条目](#)
 - [6.5 查询一个房间的房客](#)
 - [6.6 发现客户端对 MUC 的支持](#)
- [7 房客用例](#)
 - [7.1 进入一个房间](#)
 - [7.1.1 Groupchat 1.0 协议](#)
 - [7.1.2 基本 MUC 协议](#)
 - [7.1.3 出席信息广播](#)
 - [7.1.4 缺省角色](#)
 - [7.1.5 非匿名房间](#)
 - [7.1.6 半匿名房间](#)
 - [7.1.7 密码保护房间](#)
 - [7.1.8 仅限会员房间](#)
 - [7.1.9 被禁止的用户](#)
 - [7.1.10 昵称冲突](#)
 - [7.1.11 最大用户数](#)
 - [7.1.12 锁住的房间](#)
 - [7.1.13 不存在的房间](#)
 - [7.1.14 房间记录](#)
 - [7.1.15 讨论历史](#)
 - [7.1.16 管理讨论历史](#)
 - [7.2 退出一个房间](#)
 - [7.3 更改昵称](#)
 - [7.4 更改可用性状态](#)
 - [7.5 邀请其他用户进入一个房间](#)
 - [7.5.1 直接邀请](#)
 - [7.5.2 间接邀请](#)
 - [7.6 把一对一聊天转为多用户会议](#)
 - [7.7 房客修改房间标题](#)
 - [7.8 发送私有消息](#)
 - [7.9 发送消息给所有房客](#)
 - [7.10 注册到房间](#)
 - [7.11 获取成员列表](#)
 - [7.12 发现保留的房间昵称](#)
 - [7.13 申请发言权](#)
- [8 主持人用例](#)
 - [8.1 修改房间主题](#)
 - [8.2 踢出房客](#)
 - [8.3 授予游客发言权](#)
 - [8.4 撤销与会者发言权](#)

- [8.5 修改发言权列表](#)
 - [8.6 批准发言权申请](#)
- [9 管理员用例](#)
 - [9.1 禁止用户](#)
 - [9.2 修改黑名单](#)
 - [9.3 授予成员资格](#)
 - [9.4 撤销成员资格](#)
 - [9.5 修改成员列表](#)
 - [9.6 授予主持人权限](#)
 - [9.7 撤销主持人权限](#)
 - [9.8 修改主持人列表](#)
 - [9.9 批准注册申请](#)
- [10 所有者用例](#)
 - [10.1 新建房间](#)
 - [10.1.1 一般注意事项](#)
 - [10.1.2 新建即时房间](#)
 - [10.1.3 新建保留房间](#)
 - [10.1.4 申请唯一房间名](#)
 - [10.2 随后的房间配置](#)
 - [10.2.1 配置变更通知](#)
 - [10.3 授予所有者权限](#)
 - [10.4 撤销所有者权限](#)
 - [10.5 修改所有者列表](#)
 - [10.6 授予管理员权限](#)
 - [10.7 撤销管理员权限](#)
 - [10.8 修改管理员列表](#)
 - [10.9 销毁房间](#)
- [11 错误和状态码](#)
 - [11.1 错误码](#)
 - [11.2 状态码](#)
- [12 国际化事项](#)
- [13 安全事项](#)
 - [13.1 用户验证和授权](#)
 - [13.2 端到端加密](#)
 - [13.3 隐私](#)
 - [13.4 匿名](#)
 - [13.5 拒绝服务](#)
 - [13.6 其它事项](#)
- [14 IANA 事项](#)
- [15 XMPP 登记事项](#)
 - [15.1 协议名字空间](#)
 - [15.2 服务发现种类/类型](#)
 - [15.3 服务发现特性](#)
 - [15.4 知名服务发现节点](#)

- [15.5 字段标准化](#)
 - [15.5.1 muc#register FORM TYPE](#)
 - [15.5.2 muc#request FORM TYPE](#)
 - [15.5.3 muc#roomconfig FORM TYPE](#)
 - [15.5.4 muc#roominfo FORM TYPE](#)
- [15.6 状态码登记处](#)
 - [15.6.1 过程](#)
 - [15.6.2 初始提交](#)
- [15.7 URI 查询类型](#)
 - [15.7.1 join](#)
 - [15.7.2 invite](#)
- [16 商业规则](#)
 - [16.1 Addresses](#)
 - [16.2 Message](#)
 - [16.3 Presence](#)
 - [16.4 IQ](#)
- [17 实现注意事项](#)
 - [17.1 服务端](#)
 - [17.1.1 允许的通讯](#)
 - [17.2 客户端](#)
 - [17.2.1 IRC 命令映射](#)
- [18 XML 架构](#)
 - [18.1 <http://jabber.org/protocol/muc>](#)
 - [18.2 <http://jabber.org/protocol/muc#user>](#)
 - [18.3 <http://jabber.org/protocol/muc#admin>](#)
 - [18.4 <http://jabber.org/protocol/muc#owner>](#)
 - [18.5 <http://jabber.org/protocol/muc#unique>](#)
- [19 鸣谢](#)
- [20 附录](#)
 - [20.1 附录 A: 文档信息](#)
 - [20.2 附录 B: 作者信息](#)
 - [20.3 附录 C: 法律通告](#)
 - [20.4 附录 D: 和 XMPP 的关系](#)
 - [20.5 附录 E: 讨论地点](#)
 - [20.6 附录 F: 需求一致性](#)
 - [20.7 附录 G: 备注](#)
 - [20.8 附录 H: 修订历史](#)

绪论

传统上，即时消息被视为由一对一的聊天构成而不是多对多聊天(即所谓“群聊”或“文本会议”)。群聊功能常见于一些系统如 Internet Relay Chat (IRC) 和

流行的 IM 服务所提供的聊天室功能。Jabber 社区早在 1999 年开发和实施了一个基本的群聊协议。这个“groupchat 1.0”协议为聊天室提供了一个最小功能集但是范围很有限。本协议(多用户聊天或简称 MUC)建立在向后兼容旧的“groupchat 1.0”协议的基础上但是提供高级功能如邀请, 房间主持和管理, 以及专门的房间类型。

范围

本文着重于和配置, 参与以及管理一个独立的基于文本的会议室相关的通用需求。这里所指出的需求是应用于单个房间级别的并且是“通用的”, 某种意义上它们是在 Jabber 社区广泛讨论的或在现有的 Jabber 之外的基于文本的会议环境(例如, 定义在 [RFC 1459 1](#) 中的 Internet Relay Chat 和它的继承者: [RFC 2810 2](#), [RFC 2811 3](#), [RFC 2812 4](#), [RFC 2813 5](#))中已经存在的。

本文明确地不涉及以下需求:

- 房间之间的关系(例如, 房间的层次结构)
- 多用户聊天服务的管理(例如, 管理跨越整个服务级别的权限或注册一个全局可用的房间昵称); 这些用例定义在 [Service Administration 6](#)
- 个别消息的主持
- 通过房间发送的消息的加密
- 高级特性, 如附加文件给一个房间, 集成白板, 以及和语音或视频聊天服务的接口
- MUC 部署和外来的聊天系统(例如, 和 IRC 网关或现有的其他 IM 系统)之间的交互
- 在多个 MUC 部署之间进行镜像或复制

这一受限的范围并非蔑视这些都很有用的主题; 无论如何, 这意味着本文专注于讨论和介绍一个易于理解的协议能够被类似的 Jabber 客户端和组件开发者实现。将来的协议当然可能涉及以上提到的这些主题。

需求

本文描述了由 Jabber 现有的多用户聊天服务提供的最小功能集。为了向后兼容性起见, 本文使用原来的“groupchat 1.0”协议作为基本功能, 包括以下这些:

- 每个房间被标识为 `<room@service>` (例如, `<jdev@conference.jabber.org>`), 这里 “room” 是房间的名称而 “service” 是多用户聊天服务运行所在的主机名.
- 在一个房间里每个房客被标识为 `<room@service/nick>`, 这里 “nick” 是这个房客在这个房间里的昵称, 定义于刚加入这个房间的时候, 也可以在房客驻留改房间期间修改.
- 一个用户通过发送出席信息给 `<room@service/nick>` 来加入一个房间 (也就是成为房客).
- 在多用户聊天房间里发送的消息使用特殊的类型 “groupchat” 并且被寻址于房间本身 (`room@service`), 然后反映给所有房客.
- 通过发送出席信息给 `<room@service/newnick>`, 一个房客可以改变他或她的房间昵称以及在房间中的可用性状态 .
- 通过发送一个类型为 “unavailable” 的出席信息给当前的 `<room@service/nick>`, 一个房客可以退出房间.

本文追加的特性和功能包括以下这些:

1. 本地会话日志 (不需要房间内的机器人)
2. 允许用户申请房间成员
3. 在一个非匿名房间里, 允许房客可以察看 (另) 一个房客的全 JID
4. 在一个半匿名房间里, 允许主持人可以察看一个房客的全 JID
5. 允许只有主持人修改房间主题
6. 允许主持人从房间里踢出与会者和游客
7. 在一个被主持的房间里, 主持人可以授予和撤销发言权 (也就是说, 发言的权力), 并且管理发言权列表
8. 允许管理员授权和取消主持人权力, 并且管理主持人列表
9. 允许管理员在房间禁止用户, 并管理黑名单
10. 允许管理员授予和撤销成员权力, 并且管理一个仅限成员的房间的成员列表
11. 允许所有者限制房客的数量
12. 允许所有者指定其他的所有者 (们)
13. 允许所有者授予或撤销管理特权, 并管理管理员列表
14. 允许所有者销毁房间

另外, 本文提供了协议元素用于支持以下房间类型:

1. 公共的或隐藏的
2. 持久的或临时的
3. 密码保护的或不安全的
4. 仅限成员的或开放的

5. 主持的或非主持的
6. 非匿名的或半匿名的

为了实现这些需求, 本扩展协议需要满足 '<http://jabber.org/protocol/muc>' 名字空间(以及 在主名字空间 URI 加上 #owner, #admin, 和 #user 片断).

术语

通用术语

Affiliation(岗位) -- 一个长期存在的和房间之间的联系或连接; 可能的岗位有 "owner"(所有者), "admin"(管理者), "member"(成员), 以及 "outcast"(被排斥者) (当然也可能没有岗位); 岗位(affiliation)和角色(role)是有区别的. 一个岗位跨越了用户对一个房间的访问期间.

Ban(禁止) -- 从一个房间移除一个用户以使这个用户不能够再进入这个房间 (直到这个禁令被废除为止). 一个被禁止的用户的岗位(affiliation)为 "outcast"(被排斥者).

Bare JID(纯 JID) -- 一个用户的标识符 <user@host>, 不同于任何已有会话或资源的上下文, 与之相对的是全 JID 和房间 JID.

Full JID(全 JID) -- 一个在线用户的标识符 <user@host/resource>, 不同于一个房间的上下文; 与之相对的是纯 JID 和房间 JID.

GC -- 最小的 "groupchat 1.0" 协议[7], Jabber 社区于 1999 年开发; MUC 向后兼容 GC.

History(历史) -- 有限数量的消息节, 由当前讨论的上下文提供发送给一个新的房客.

Invitation(邀请) -- 从一个用户发出的特殊消息给另一个用户, 邀请对方加入房间.; the invitation can be sent directly (see Direct MUC Invitations [8]) or mediated through the room (as described under Inviting Another User to a Room).

IRC -- Internet Relay Chat.

Kick(踢人) -- 临时从一个房间移除一个与会者或游客; 这个用户任何时候都可以再次进入这个房间. 一个被踢的用户角色是 "none".

Logging(记录) -- 存储发生在一个房间的讨论内容用于公开发布到房间上下文之外的地方.

Member(成员) -- 一个用户在一个仅限会员的房间内处于“white list”(白名单)内, 或已经注册到一个公开的房间. 一个成员的岗位是“member”.

Moderator(主持人) -- 一个房间角色, 通常和房间的管理有关但是这个角色可以被赋予非管理员; 可以踢人, 可以授予和撤销发言权, 等等. 一个主持人的角色是“moderator”.

MUC -- 本文所定义的基于文本会议的多用户聊天协议.

Occupant(房客) -- 一个房间里的任何 Jabber 用户 (这是一个“抽象类”并且不对应任何特定的角色).

Outcast(被排斥者) -- 一个被某个房间禁止的用户. 一个被排斥者的岗位是“outcast”.

Participant(与会者) -- 一个没有管理权限的房客; 在一个被主持的房间里, 参与者更多地被定义为有发言权的 (与之相反的是游客). 一个与会者的角色是“participant”.

Private Message(私有消息) -- 从一个房客直接发给另一个房间 JID 的消息 (不是房间本身广播给所有房客的消息).

Role(角色) -- 在一个房间里的一个临时的地位或者权限级别, 对于这个房间中的用户的长期岗位来说是唯一的; 可能的角色有 “moderator”(主持人), “participant”(与会者), 和 “visitor”(游客) (也可能没有预定义的角色). 一个角色仅仅存在于一个房客访问一个房间的期间.

Room(房间) -- 一个虚拟的地方, Jabber 用户象征性地加入它, 来和其他用户一起参与一个实时的基于文本的会议.

Room Administrator(房间管理员) -- 一个由房间所有者授权的用户, 可以执行管理功能, 如禁止用户等等; 无论如何, 不允许改变定义的房间特性. 一个管理员的岗位是“admin”.

Room ID(房间 ID) -- 一个房间 JID 的节点标识符部分, 它可以是不透明的因而对人类用户没有什么含义 (见 语法的商业规则 Business Rules for syntax); 与之相对的是房间名.

Room JID(房间 JID) -- 在一个房间上下文中的一个房客, 以 <room@service/nick> 来标识; 与之相对的是纯 JID 和全 JID.

Room Name(房间名) -- 一个用户友好的, 自然语言的房间名字, 由房间所有者配置并在服务查询中展示; 与之相对的是房间 ID.

Room Nickname(房间昵称) -- 房间 JID 的资源标识符部分 (见语法的商业规则); 这是一个房客在这个房间中所呈现的“友好的名字”.

Room Owner(房间所有者) -- 建立某个房间的 Jabber 用户或一个被房间创建者或所有者指派拥有所有者权限(如果允许的话)的 Jabber 用户; 它被允许改变定义好的房间特性, 也可以执行全部的管理功能. 一个所有者的岗位为"owner".

Room Roster(房间名册) -- 一个房间中的所有房客在一个 Jabber 客户端的展现.

Server(服务器) -- 一个 Jabber 服务器, 可以关联或不关联一个基于文本的会议服务.

Service(服务) -- 一个主机, 提供基于文本的会议的能力; 通常但不必须是一个 Jabber 服务器的子域(例如, conference.jabber.org).

Subject(主题) -- 一个房间的临时讨论标题.

Visit(访问) -- 一个房间的一个用户的"session"(会话), 当用户进入这个房间时开始(也就是说, 成为一个房客), 结束于用户离开房间之时.

Visitor(游客) -- 在一个被主持的房间里一个没有发言权的房客(相反则是一个与会者). 一个游客的角色是"visitor".

Voice(发言权) -- 在一个被主持的房间里, 发送消息给全部房客的权限.

房间类型

Hidden Room(隐藏房间) -- 一个无法被任何用户以普通方法如搜索和服务查询来发现的房间; 反义词: 公开(public)房间.

Members-Only Room(仅限会员的房间) -- 如果一个用户不在成员列表中则无法加入的一个房间; 反义词: 开放(open)房间.

Moderated Room(被主持的房间) -- 只有有"发言权"的用户才可以发送消息给所有房客的房间; 反义词: 非主持的(Unmoderated)房间.

Non-Anonymous Room(非匿名房间) -- 一个房客的全 JID 会暴露给所有其他房客的房间, 尽管房客可以选择任何期望的房间昵称; 相对的是半匿名(Semi-Anonymous)房间.

Open Room(开放房间) -- 任何人可以加入而不需要在成员列表中的房间; 反义词: 仅限会员的房间.

Password-Protected Room(密码保护房间) -- 一个用户必须提供正确密码才能加入的房间; 反义词: 非保密房间.

Persistent Room(持久房间) -- 如果最后一个房客退出也不会被销毁的房间；反义词：临时房间.

Public Room(公开房间) -- 用户可以通过普通方法如搜索和服务查询来发现的房间；反义词：隐藏房间.

Semi-Anonymous Room(半匿名房间) -- 一个房客的全 JID 只能被房间管理员发现的房间；相对的是非匿名 (Non-Anonymous) 房间.

Temporary Room(临时房间) -- 如果最后一个房客退出就会被销毁的房间；反义词：持久房间.

Unmoderated Room(非主持的房间) -- 任何房客都被允许发送消息给所有房客的房间；反义词：被主持的房间.

Unsecured Room(非保密房间) -- 任何人不需要提供密码就可以进入的房间；反义词：密码保护房间.

登场人物

本文的大部分例子使用了莎士比亚的《麦克白》中第四幕第一场开头女巫在黑洞中开会的场景，在这里使用“coven@chat.shakespeare.lit”代表聊天室. 人物如下：

表 1: 剧中人

Room Nickname	Full JID	Affiliation
firstwitch	cronel@shakespeare.lit/desktop	Owner
secondwitch	wiccarocks@shakespeare.lit/laptop	Admin
thirdwitch	hag66@shakespeare.lit/pda	None

角色(Roles)，岗位(Affiliations)和权限(Privileges)

A user might be allowed to perform any number of actions in a room, from joining or sending a message to changing configuration options or destroying the room altogether. We call each permitted action a “privilege”. There are two ways we might structure privileges:

1. Define each privilege atomically and explicitly define each user’s particular privileges; this is flexible but can be confusing to manage.

2. Define bundles of privileges that are generally applicable and assign a user-friendly "shortcut" to each bundle (e.g., "moderator" or "admin").

MUC 使用第 2 种方式。

有两个尺度我们可以用来衡量一个用户的连接或在一个房间的地位。一个是用户和一个房间的长期的联系 -- 例如，用户的状态是一个所有者或一个被排斥者。另一个是当用户驻留于一个聊天室的时候的角色 -- 例如，一个房客的地位是主持人，有权踢出游客和与会者。这两个尺度各自都是唯一的，因为一个岗位是跨越访问的，而一个角色只存在于一次访问期间。另外，在角色和岗位之间没有一对一的对应关系；例如，某个不从属于某房间的人可能成为一个(临时的)主持人，一个成员可能在一个被主持的房间中是一个与会者或游客者。这些概念以下全面解释。

角色

以下是已定义的角色：

表 2：角色

名称	支持
主持人 Moderator	必需的
无 None	缺少角色
与会者 Participant	必需的
游客 Visitor	推荐的

角色是临时的，它不一定要在用户对房间的访问中持久化，它可以 (MAY) 在一个房客访问房间期间改变。一个实现可以 (MAY) 在一次访问期间持久化角色并且应该 (SHOULD) 在被主持的房间这样做（因为在游客和与会者之间，唯一性对一个被主持的房间是很关键的）。

在角色和岗位之间没有一对一的映射（例如，一个成员可以是一个与会者或一个游客）。

在房间会话中，一个主持人是最有权力的房客，它能在某种程度走上管理房间的其他房客的角色。一个与会者的权力小于一个主持人，尽管他或她有权发言。在一个被主持的房间会话中游客是一个更受限制的角色，因为访问者不允许发送消息给所有房客。

角色的授予，撤销，和维护是基于房客的房间昵称或全 JID，而不是纯 JID。和这些角色相关的权限，还有角色改变触发的动作，定义在下文中。

所有在房间中生成或反射的出席信息中关于角色的信息必须(MUST)被发送, 从而发送给房客们.

权限

大部分情况下, 角色存在于一个层次中. 例如, 一个与会者可以做任何游客能做的事, 而一个主持人可以做任何与会者能做的事. 每个角色拥有下一级角色所没有的权限; 这些权限定义于下表作为缺省值(一个实现可以(MAY)提供配置选项来重载这些缺省值).

表 3: 和角色相关的权限

权限	无	游客	与会者	主持人
在房间中出席	否	是	是	是
接收消息	否	是	是	是
接收房客出席信息	否	是*	是	是
出席信息广播到房间	否	是*	是	是
改变可用性状态	否	是	是	是
改变房间昵称	否	是*	是	是
发送私人消息	否	是*	是	是
邀请其他用户	否	是*	是*	是
发送消息给所有人	否	否**	是	是
修改标题	否	否*	是*	是
踢出与会者和游客	否	否	否	是
授予发言权	否	否	否	是
撤销发言权	否	否	否	是***

- 缺省; 设定配置时可以(MAY)修改这个权限.
- - 一个实现可以(MAY)在非主持的房间里缺省地授予发言权给游客.
- - - 主持人不能(MUST NOT)从一个管理员或所有者收回发言权.

默认角色

The following table summarizes the initial default roles that a service SHOULD set based on the user's affiliation (there is no role associated with the "outcast" affiliation, since such users are not allowed to enter the room).

Table 4: Initial Role Based on Affiliation

Room Type	None	Member	Admin	Owner
Moderated	Visitor	Participant	Moderator	Moderator
Unmoderated	Participant	Participant	Moderator	Moderator
Members-Only	N/A *	Participant	Moderator	Moderator
Open	Participant	Participant	Moderator	Moderator

- Entry is not permitted.

变更角色

一个房客的角色变更方法是定义好的。有时候房客自己的动作导致变更（例如，加入或退出房间），反之有时候由主持人、管理员或所有者的动作导致变更。如果一个房客的角色改变了，一个 MUC 服务实现必须 (MUST) 变更这个房客的角色来反映这个变更并且传达这个变更给所有房客。角色的变更和它们触发的动作定义于下表。

表 4：角色状态表

>	无	游客	与会者	主持人
无	--	进入被主持的房间	进入非主持的房间	管理员或所有者进入房间
游客	退出房间或被主持人踢出房间	--	主持人授予发言权	管理员或所有者授予主持人权限
与会者	退出房间或被主持人踢出房间	主持人撤销发言权	--	管理员或所有者授予主持人权限
主持人	退出房间	管理员或所有者改变角色成为游客*	管理员或所有者改变角色成为与会者或撤销主持人权限*	--

- 一个主持人不能 (MUST NOT) 从一个岗位等于或高于主持人的房客那里收回主持人权限。

注意：特定的角色一般暗含特定的权限。例如，一个管理员或所有者自动成为一个主持人，所以如果一个房客被授予管理员地位那么这个房客事实上将被授予主持人权限；类似的，当一个房客成为一个被主持的房的成员，这个房客自动拥有一个与会者的角色。无论如何，失去管理员地位并不足以意味这个房客不再是主持人（因为只要是与会者就可能成为一个主持人）。因此，当一个房客被授予特定的岗位的时候所拥有的角色是固定的，反之当一个房客失去一个特定的岗位时它的角色是不确定的并取决于（服务的）实现。因为一个客户端无

法预料是否在撤销某个岗位之后这个角色成为什么，如果它不想同时移除管理员/所有者权限和主持人角色，那么除了岗位变更之外它还必须特意请求角色变更。

岗位

已定义了以下岗位：

1. 所有者
2. 管理员
3. 成员
4. 被排斥者
5. 无（缺少岗位）

必须支持“所有者”这个岗位，推荐支持“管理员”，“成员”，“被排斥者”的岗位。（“无”表示缺少岗位）

这些岗位是长时间的跨越一个用户对这个房间的访问期间的并且不受房间里事件的影响。而且，这些岗位和一个房客在房间中的角色之间没有一对一的映射关系。岗位被授予，撤销，和维护都是基于这个用户的纯 JID。

如果一个没有已定义的岗位的用户进入一个房间，这个用户的岗位被定义为“无”；无论如何，这个岗位不能跨越（多次的）访问（也就是说，一个服务不会跨越访问维护一个“无 列表”）。

“成员”岗位为房间所有者或管理员提供了一个方法来指定一个“白名单”，其中的用户被允许加入一个仅供会员的房间。当一个成员加入了一个仅供会员的房间，他或她的岗位不会改变，无论他或她的角色是什么。成员岗位也为用户提供一个方法来高效地注册一个开放的房间并在某种方式意义上保持和那个房间的联系（例如可能在房间里预留那个用户的昵称）。

一个被排斥者就是一个被从房间踢出来并且不允许进入那个房间的用户。

关于岗位的信息必须(MUST)由房间生成或反射到所有的出席信息节之中发送给房客们。

权限

大部分情况下，岗位存在一个层次结构。例如，一个所有者可以做任何管理员能做的事情，而一个管理员可以做任何成员能做的事情。每个岗位拥有其下一级岗位所没有的权限；这些权限定义在下表中。

表 5：和岗位相关的权限

权限	Outcast (被排斥者)	None (无)	Member (成员)	Admin (管理员)	Owner (所有者)
进入房间	否	是*	是	是	是
注册一个开放的房间	否	是	N/A	N/A	N/A
接收成员列表	否	否**	是	是	是
加入一个仅限会员的房间	否	否	是*	是	是
禁止成员并把用户的岗位删除	否	否	否	是	是
编辑成员列表	否	否	否	是	是
编辑主持人列表	否	否	否	是**	是**
编辑管理员列表	否	否	否	否	是
编辑所有者列表	否	否	否	否	是
变更房间定义	否	否	否	否	是
销毁房间	否	否	否	否	是

- 作为缺省值，一个无岗位的用户进入一个被主持的房间的角色是一个游客，而进入一个开放的房间的角色是一个与会者。一个成员进入一个房间的角色是与会者。一个管理员或所有者进入房间的角色是一个主持人。
- - 一个管理员或所有者不能 (MUST NOT) 撤销另一个管理员或所有者的权限。

变更岗位

一个用户的岗位变更方法已经定义得很完善。有时用户自己的动作导致这些变更(例如，注册为一个房间的新成员)，反之有时候一个管理员或所有者的动作导致了这些变更。如果一个用户的岗位改变了，一个 MUC 服务实现必须 (MUST) 变更这个用户的岗位来反射这一变更并通知所有房客。岗位变更和他们触发的动作定义在下表中。

表 6：岗位状态表

	被排斥者 (Outcast)	无 (None)	成员 (Member)	管理员 (Admin)	所有者 (Owner)
被排斥者	--	管理员或	管理员或所有者增	所有者增	所有者增

(Outcast)		所有者移除屏蔽	加用户到成员列表	加用户到管理员列表	加用户到所有者列表
无 (None)	管理员或所有者使用屏蔽	--	管理员或所有者增加用户到成员列表, 或用户注册一个成员 (如果允许)	所有者增加用户到管理员列表	所有者增加用户到所有者列表
成员 (Member)	管理员或所有者使用屏蔽	管理员或所有者变更岗位为 "none"	--	所有者增加用户到管理员列表	所有者增加用户到所有者列表
管理员 (Admin)	所有者使用屏蔽	所有者变更岗位为 "none"	所有者变更岗位为 "member"	--	所有者增加用户到所有者列表
所有者 (Owner)	所有者使用屏蔽	所有者变更岗位为 "none"	所有者变更岗位为 "member"	所有者变更岗位为 "admin"	--

实体用例

一个 MUC 实现必须 (MUST) 支持[服务发现 7](#).

MUC 的发现组件支持

一个 Jabber 实体可能希望发现是否一个服务实现了多用户聊天协议; 为了达到这个目的, 它发送一个服务发现信息 ("disco#info") 查询给这组件的 JID:

例子 1. 用户通过 Disco 查询聊天服务是否支持 MUC

```
<iq from='hag66@shakespeare.lit/pda'
  id='disco1'
  to='macbeth.shakespeare.lit'
  type='get'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

服务必须 (MUST) 返回它的的身份和它所支持的特性:

例子 2. 服务返回 Disco Info 结果

```
<iq from='macbeth.shakespeare.lit'
```



```

    id='disco1'
    to='hag66@shakespeare.lit/pda'
    type='result' >
<query xmlns='http://jabber.org/protocol/disco#info' >
  <identity
    category='conference'
    name='Macbeth Chat Service'
    type='text' />
    <feature var='http://jabber.org/protocol/muc' />
  </query>
</iq>

```

注意：因为 MUC 是旧的“groupchat 1.0”协议的超集，一个 MUC 服务不应该 (SHOULD NOT) 返回一个<feature var='gc-1.0' />条目在一个 disco#info 结果中。

发现房间

发现服务条目 (“disco#items”) 协议使得一个用户可以向一个服务查询相关的条目列表，在一个聊天服务中这包含这个服务所承载的所有特定房间的集合。

例子 3. 用户向聊天服务查询房间

```

<iq from='hag66@shakespeare.lit/pda'
  id='disco2'
  to='macbeth.shakespeare.lit'
  type='get' >
  <query xmlns='http://jabber.org/protocol/disco#items' />
</iq>

```

服务应该 (SHOULD) 返回它承载的所有房间的列表。

例子 4. 服务返回 Disco Item 结果

```

<iq from='macbeth.shakespeare.lit'
  id='disco2'
  to='hag66@shakespeare.lit/pda'
  type='result' >
  <query xmlns='http://jabber.org/protocol/disco#items' >
    <item jid='heath@macbeth.shakespeare.lit'
      name='A Lonely Heath' />
    <item jid='darkcave@macbeth.shakespeare.lit'
      name='A Dark Cave' />
    <item jid='forres@macbeth.shakespeare.lit'

```

```

        name='The Palace' />
    <item jid='inverness@macbeth.shakespeare.lit'
        name='Macbeth&apos;s Castle' />
</query>
</iq>

```

如果全部房间的列表太大(详见 [XEP-0030](#)), 服务可以(MAY)只返回部分的房间列表. 如果这样做了, 它应该 SHOULD 包含一个 <set/> 元素 (定义在 [Result Set Management](#) 8) 以表明这个列表不是全部的结果集.

例子 5. 服务返回 Disco Item 结果的部分列表

```

<iq from='rooms.shakespeare.lit'
    id='disco-rsm-1'
    to='hag66@shakespeare.lit/pda'
    type='result'>
    <query xmlns='http://jabber.org/protocol/disco#items'>
        <item jid='alls-well-that-ends-well@rooms.shakespeare.lit' />
        <item jid='as-you-like-it@rooms.shakespeare.lit' />
        <item jid='cleopatra@rooms.shakespeare.lit' />
        <item jid='comedy-of-errors@rooms.shakespeare.lit' />
        <item jid='coriolanus@rooms.shakespeare.lit' />
        <item jid='cymbeline@rooms.shakespeare.lit' />
        <item jid='hamlet@rooms.shakespeare.lit' />
        <item jid='henry-the-fourth-one@rooms.shakespeare.lit' />
        <item jid='henry-the-fourth-two@rooms.shakespeare.lit' />
        <item jid='henry-the-fifth@rooms.shakespeare.lit' />
        <set xmlns='http://jabber.org/protocol/rsm'>
            <first
index='0'>alls-well-that-ends-well@rooms.shakespeare.lit</first>
            <last>henry-the-fifth@rooms.shakespeare.lit</last>
            <count>37</count>
        </set>
    </query>
</iq>

```

查询房间信息

使用 disco#info 协议, 一个用户也可以查询一个特定房间的详情. 为了在进入房间之前确定这个房间的隐私和安全配置用户应该(SHOULD)这样做(详见[安全事项](#)).

例子 6. 用户查询特定聊天室的信息

```
<iq from='hag66@shakespeare.lit/pda'
  id='disco3'
  to='darkcave@macbeth.shakespeare.lit'
  type='get'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

房间必须(MUST)返回它的标识并且应该(SHOULD)返回它支持的特性:

例子 7. 房间返回查询信息结果

```
<iq from='darkcave@macbeth.shakespeare.lit'
  id='disco3'
  to='hag66@shakespeare.lit/pda'
  type='result'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    <identity
      category='conference'
      name='A Dark Cave'
      type='text' />
    <feature var='http://jabber.org/protocol/muc' />
    <feature var='muc_passwordprotected' />
    <feature var='muc_hidden' />
    <feature var='muc_temporary' />
    <feature var='muc_open' />
    <feature var='muc_unmoderated' />
    <feature var='muc_nonanonymous' />
  </query>
</iq>
```

注意: 因为 MUC 是旧的 “groupchat 1.0” 协议的超集, 一个 MUC 房间不应该 (SHOULD NOT) 在一个 disco#info 结果中返回<feature var='gc-1.0' />条目. 房间应该 (SHOULD) 返回它支持的实质的有意义的特性, 例如密码保护和房间主持 (这些特性被完整地列入了特性注册, 由 [XMPP Registrar](#) 维护; 也见于本文的 [XMPP 注册](#) 章节).

一个聊天室可以(MAY)使用[服务查询扩展 9](#)在它的 disco#info 应答中返回更详细的信息, 通过包含一个隐含的 FORM_TYPE 属性值 “[http://jabber.org/protocol/muc#roominfo](#)”来标识. 这些信息可能包括关于一个房间的更详细的描述, 当前的房间标题, 以及这个房间当前的房客数量:

例子 8. 房间返回扩展的查询信息结果

```
<iq from='darkcave@macbeth.shakespeare.lit'
  id='disco3a'
```

```

    to='hag66@shakespeare.lit/pda'
    type='result'>
<query xmlns='http://jabber.org/protocol/disco#info'>
  <identity
    category='conference'
    name='A Dark Cave'
    type='text' />
  <feature var='http://jabber.org/protocol/muc' />
  <feature var='muc_passwordprotected' />
  <feature var='muc_hidden' />
  <feature var='muc_temporary' />
  <feature var='muc_open' />
  <feature var='muc_unmoderated' />
  <feature var='muc_nonanonymous' />
  <x xmlns='jabber:x:data' type='result'>
    <field var='FORM_TYPE' type='hidden'>
      <value>http://jabber.org/protocol/muc#roominfo</value>
    </field>
    <field var='muc#roominfo_description' label='Description'>
      <value>The place for all good witches!</value>
    </field>
    <field var='muc#roominfo_changesubject' label='Whether Occupants
May Change the Subject'>
      <value>true</value>
    </field>
    <field var='muc#roominfo_contactjid' label='Contact Addresses'>
      <value>cronel@shakespeare.lit</value>
    </field>
    <field var='muc#roominfo_subject' label='Subject'>
      <value>Spells</value>
    </field>
    <field var='muc#roominfo_occupants' label='Number of occupants'>
      <value>3</value>
    </field>
    <field var='muc#roominfo_lang' label='Language of discussion'>
      <value>en</value>
    </field>
    <field var='muc#roominfo_logs' label='URL for discussion logs'>
      <value>http://www.shakespeare.lit/chatlogs/darkcave/</value>
    </field>
    <field var='muc#roominfo_pubsub' label='Associated pubsub node'>
      <value>xmpp:pubsub.shakespeare.lit?node=chatrooms/darkcave</value>
    </field>
  </x>
</query>

```

```
</x>
</query>
</iq>
```

某些扩展的房间信息可能是动态生成的(例如, 讨论记录的 URL 地址, 它可能取决于服务器那一层的配置); 反之另一些信息则可能基于房间那一层的配置, 任何定义在 [muc#roomconfig FORM TYPE](#) 里的字段都可以用于扩展服务发现的字段(如上文所示的 `muc#roomconfig_changesubject` 字段).

注意: 前述 '<http://jabber.org/protocol/muc#roominfo>' FORM_TYPE 的扩展服务发现字段将来还可以扩充(通过本文的[字段标准化](#)章节描述的机制).

查询房间条目

一个用户也可以(MAY)向一个特定的聊天室查询和它相关的条目:

例子 9. 用户查询和一个特定聊天室相关的条目

```
<iq from='hag66@shakespeare.lit/pda'
  id='disco4'
  to='darkcave@macbeth.shakespeare.lit'
  type='get'>
  <query xmlns='http://jabber.org/protocol/disco#items' />
</iq>
```

一个实现可以(MAY)返回现有房客的列表(如果那信息是可公开的), 或不返回列表(如果那信息是私有的).

例子 10. 房间返回查询条目结果(条目是公开的)

```
<iq from='darkcave@macbeth.shakespeare.lit'
  id='disco4'
  to='hag66@shakespeare.lit/pda'
  type='result'>
  <query xmlns='http://jabber.org/protocol/disco#items'>
    <item jid='darkcave@macbeth.shakespeare.lit/firstwitch' />
    <item jid='darkcave@macbeth.shakespeare.lit/secondwitch' />
  </query>
</iq>
```

注意: 这些 `<item/>` 元素由 `disco#items` 名字空间限定, 而不是 `muc` 名字空间; 这意味着他们不能拥有 '`affiliation`' 或 '`role`' 属性, 例如.

例子 11. 房间返回空的查询条目结果(条目是私有的)

```
<iq from='darkcave@macbeth.shakespeare.lit'
    id='disco4'
    to='hag66@shakespeare.lit/pda'
    type='result'>
  <query xmlns='http://jabber.org/protocol/disco#items' />
</iq>
```

查询一个房间的房客

如果一个非房客试图发送一个查询请求给一个<room@service/nick>类型的地址, 一个 MUC 服务应该(SHOULD)返回这个请求给这个实体并指明一个<bad-request/>错误条件. 如果一个房客发送这样一个请求, 服务可以(MAY)把它传递给指定的接收者; 详见本文的 [实现注意事项](#) 章节.

发现客户端对 MUC 的支持

一个 Jabber 用户可能想发现这个用户的某个联系人是否支持多用户聊天协议. 这可以使用服务发现(协议)来完成.

例子 12. 用户查询联系人对于 MUC 的支持

```
<iq from='hag66@shakespeare.lit/pda'
    id='disco5'
    to='wiccarocks@shakespeare.lit/laptop'
    type='get'>
  <query xmlns='http://jabber.org/protocol/disco#info' />
</iq>
```

客户端应该(SHOULD)返回它的标识和它支持的特性:

例子 13. 联系人返回发现信息结果

```
<iq from='wiccarocks@shakespeare.lit/laptop'
    id='disco5'
    to='hag66@shakespeare.lit/pda'
    type='result'>
  <query xmlns='http://jabber.org/protocol/disco#info'>
    <identity
      category='client'
      type='pc' />
    ...
    <feature var='http://jabber.org/protocol/muc' />
    ...
  </query>
```

```
</query>
</iq>
```

一个用户也可能查询一个联系人在哪个房间。这可以通过特定服务发现节点'<http://jabber.org/protocol/muc#rooms>' 查询联系人的全 JID(<user@host/resource>) 来完成：

例子 14. 用户在当前房间查询联系人

```
<iq from='hag66@shakespeare.lit/pda'
  id='rooms1'
  to='wicarocks@shakespeare.lit/laptop'
  type='get'>
  <query xmlns='http://jabber.org/protocol/disco#items'
    node='http://jabber.org/protocol/muc#rooms' />
</iq>
```

例子 15. 联系人返回房间查询结果

```
<iq from='wicarocks@shakespeare.lit/laptop'
  id='rooms1'
  to='hag66@shakespeare.lit/pda'
  type='result'>
  <query xmlns='http://jabber.org/protocol/disco#items'
    node='http://jabber.org/protocol/muc#rooms' />
    <item jid='darkcave@macbeth.shakespeare.lit' />
    <item jid='characters@conference.shakespeare.lit' />
  </query>
</iq>
```

可选的，联系人可以 (MAY) 把它的房间昵称作为 'name' 属性的值返回：

```
...
<item jid='darkcave@macbeth.shakespeare.lit'
  name='secondwitch' />
...
```

房客用例

在一个多用户聊天环境中主要的行为者是房客，它可以被认为存在于一个多用户聊天室“之内”并且参与那个房间的讨论（在本协议中，与会者和游客“仅仅”被认为是房客，因为他们不拥有管理员权限）。为了更加清晰起见，本文中的协议元素中涉及到驻留者的用例分为以下三类：

1. 现存于 “groupchat 1.0” 协议的最小功能集
2. 对于 “groupchat 1.0” 协议直接的应用，如处理一些和新房间类型有关的错误
3. 用来处理 “groupchat 1.0” 协议未涉及的功能的额外的协议元素 (房间邀请，房间密码，和房间角色及岗位相关的扩展出席信息)；在
'<http://jabber.org/protocol/muc#user>' 名字空间

注意：这里所有客户端生成的例子是从服务的角度来展示的，所以所有由服务收到的节都包含一个 'from' 属性来表达发送者的全 JID (这个 from 属性是由一个通用的 Jabber 路由或会话管理者加入的)。另外，通常的表示请求已被完成的 IQ 结果节 (如 [RFC 3920](#) [10] 中所要求的) 未显示在这里。

进入一个房间

Groupchat 1.0 协议

为了参加一个多用户聊天室的讨论，一个 Jabber 用户必须 (MUST) 首先进入一个房间成为一个房客。在旧的 “groupchat 1.0” 协议中，这是通过发送出席信息 `<room@service/nick>` 来实现的，这里 “room” 是房间 ID，“service” 是聊天服务的主机名，“nick” 是这个用户在这房间里预期的昵称：

例子 16. Jabber 用户进入一个房间 (Groupchat 1.0)

```
<presence
  from='hag66@shakespeare.lit/pda'
  to='darkcave@macbeth.shakespeare.lit/thirdwitch' />
```

在这个例子中，一个全 JID 为 “hag66@shakespeare.lit/pda” 的用户请求用昵称 “thirdwitch” 进入位于 “macbeth.shakespeare.lit” 聊天服务的房间 “darkcave”。

如果用户未指定一个房间昵称，服务应该 (SHOULD) 返回一个 `<jid-malformed/>` 错误：

例子 17. Jabber 用户进入一个房间 (Groupchat 1.0)

```
<presence
  from='darkcave@macbeth.shakespeare.lit'
  to='hag66@shakespeare.lit/pda'
  type='error' >
  <error code='400' type='modify' >
    <jid-malformed xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</presence>
```


基本 MUC 协议

兼容的多用户聊天服务必须 (MUST) 接受知道“groupchat 1.0” (GC) 协议或 multi-user chat (MUC) 协议的任何客户端发出上述请求进入会议室；无论如何，MUC 客户端应该 (SHOULD) 声明他们的有能力支持 MUC 协议，方法是在出席信息节里面包含一个空的 `<x/>` 元素，满足名字空间

`'http://jabber.org/protocol/muc'` (注意不需要 `'#user'` 部分)：

例子 18. Jabber 用户准备进入一个房间 (Multi-User Chat)

```
<presence
  from="hag66@shakespeare.lit/pda"
  to='darkcave@macbeth.shakespeare.lit/thirdwitch'>
  <x xmlns='http://jabber.org/protocol/muc' />
</presence>
```

注意：如果发生了一个和加入房间有关的错误，服务应该 SHOULD 返回一个包含 MUC 子元素 (i.e., `<x xmlns='http://jabber.org/protocol/muc' />`) 的 `<presence/>` 节，其 `type` 为 “error”。

在尝试进入房间之间，一个兼容 MUC 的客户端应该 (SHOULD) 首先查询它的保留的房间昵称 (如果有的话)，接下来的协议本文中的 [发现保留的房间昵称](#) 章节对此作了定义。

出席信息广播

如果服务能够添加用户到房间，它必须 (MUST) 从所有现存的房客的房间 JID 发送出席信息给新的房客的全 JID，包括扩展的关于角色的出席信息，一个满足 `'http://jabber.org/protocol/muc#user'` 名字空间的 `<x/>` 元素并包含一个 `<item/>` 子元素，这个子元素的 `'role'` 属性值设为 “moderator”，“participant”，或 “visitor”，这个子元素的 `'affiliation'` 属性值设为 “owner”，“admin”，“member”，或 “none” 中的一个：

例子 19. 服务从现有的房客发送出席信息给新的房客

```
<presence
  from='darkcave@macbeth.shakespeare.lit/firstwitch'
  to='hag66@shakespeare.lit/pda'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='owner' role='moderator' />
  </x>
</presence>
<presence
  from='darkcave@macbeth.shakespeare.lit/secondwitch'
  to='hag66@shakespeare.lit/pda'>
```

```
<x xmlns='http://jabber.org/protocol/muc#user'>
  <item affiliation='admin' role='moderator' />
</x>
</presence>
```

这个示例中，用户已从前一个例子进入房间，有两个人已经在房间里：一个是昵称为“firstwitch”的(房间拥有者)，另一个是昵称为“secondwitch”的(房间管理员)。

服务也必须(MUST)从新进入的房客的房间JID向所有房客的全JID发送出席信息(含新房客)：

例子 20. 服务发送新房客的出席信息给所有房客

```
<presence
  from='darkcave@macbeth.shakespeare.lit/thirdwitch'
  to='cronel@shakespeare.lit/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='member' role='participant' />
  </x>
</presence>

<presence
  from='darkcave@macbeth.shakespeare.lit/thirdwitch'
  to='wiccarocks@shakespeare.lit/laptop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='member' role='participant' />
  </x>
</presence>

<presence
  from='darkcave@macbeth.shakespeare.lit/thirdwitch'
  to='hag66@shakespeare.lit/pda'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='member' role='participant' />
    <status code='110' />
  </x>
</presence>
```

在这个例子里，初始的房间出席信息从新房客(thirdwitch)发送给所有房客，包括这个新房客自己。看看上面最后一个节，由房间以房客的名义发送给用户自己的出席信息，应该 SHOULD 包含一个 110 状态码，这样用户就知道这个出席信息来自于作为房客的那个他自己。

服务可以 MAY 重写新房客的房间昵称（例如，如果房间昵称被锁定）。如果服务不接受新房客请求的房间昵称，而是分配一个新的房间昵称，它必须 MUST 包含一个“210”状态码在发送给这个新房客的出席信息广播里。

例子 21. 服务发送新房客的出席信息给新房客

```
<presence
  from='darkcave@macbeth.shakespeare.lit/thirdwitch'
  to='hag66@shakespeare.lit/pda'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='member' role='participant' />
    <status code='110' />
    <status code='210' />
  </x>
</presence>
```

注意：发送给新房客的出席信息的顺序是很重要的。服务必须 MUST 首先发送现有房客的完整列表给这个新房客，然后只发送新房客自己的出席信息给新房客。这有助于客户端知道什么时候它收到了完整的房间名册（“room roster”）。

发送出席信息广播之后（并且只在这之后），服务可以发送讨论历史，即时消息，出席信息更新，以及其他房间内的流量。

缺省角色

下表总结了初始缺省的角色，一个服务应该根据用户的岗位来设置它们（没有和被排斥者“outcast”岗位相关的角色，因为这些用户不允许进入房间）。

表 7：基于岗位的初始角色

房间类型	无	成员	管理员	所有者
被主持的	游客	与会者	主持人	主持人
非主持的	与会者	与会者	主持人	主持人
仅限会员的	N/A *	与会者	主持人	主持人
开放的	与会者	与会者	主持人	主持人

- 实体不被允许.

非匿名房间

如果房间是非匿名的，服务必须 MUST 发送新房客的全 JID 给所有房客，使用满足 '<http://jabber.org/protocol/muc#user>' 名字空间的扩展出席信息，其中带有 <x/> 元素并包含一个 <item/> 子元素，其 'jid' 属性值为这个房客的全 JID：

例子 22. 服务发送全 JID 给所有房客

```
<presence
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='cronel@shakespeare.lit/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='none'
      jid='hag66@shakespeare.lit/pda'
      role='participant' />
  </x>
</presence>

[ ... ]
```

如果这个用户正在进入一个非匿名房间(即, 它如上所示, 向所有房客通报每个房客的全 JID), 服务应该 SHOULD 允许该用户加入本房间, 但是必须 MUST 同时警告该用户本房间是非匿名的. 应该 SHOULD 在房间发送给这个新房客的初始出席信息种包含状态码 "100" 来实现这一点:

例子 23. 服务发送新房客的出席信息给新房客

```
<presence
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='hag66@shakespeare.lit/pda'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='member' role='participant' />
    <status code='100' />
    <status code='110' />
    <status code='210' />
  </x>
</presence>
```

无论如何, 也可以 MAY 发送一个 "groupchat" 类型的消息给新房客来达到上述目的, 这个消息应该包含一个 <x/> 子元素, 并拥有 <status/> 子元素, 并且其 'code' 属性值为 "100":

例子 24. 服务警告新房客(该房间)非匿名

```
<message
  from='darkcave@chat.shakespeare.lit'
  to='hag66@shakespeare.lit/pda'
  type='groupchat'>
  <body>This room is not anonymous.</body>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <status code='100' />
```

```
</x>
</message>
```

附带的状态码协助客户端展示它们自己的通知消息（例如，和用户所在地方有关的信息）。

半匿名房间

如果房间是半匿名的，服务必须 MUST 如上文所述从新房客发送出席信息给所有房客，但是必须 MUST 只在发给“主持人”的时候发送新房客的全 JID, 而非主持人则不发(全 JID)。

(注意：所有随后的例子中，涉及的<item/>元素都带有‘jid’属性，即使这个信息在半匿名房间里不被发送给非主持人.)

密码保护房间

如果房间要求密码验证而用户不能提供(或密码错误)，服务必须 MUST 拒绝访问这个房间并且通知该用户它们是未被授权的；具体方法是返回一个类型为“error”的出席信息节并标明 <not-authorized/> 错误：

例子 25. 服务拒绝访问, 因为(用户)未提供密码

```
<presence
  from='darkcave@chat.shakespeare.lit'
  to='hag66@shakespeare.lit/pda'
  type='error'>
  <x xmlns='http://jabber.org/protocol/muc' />
  <error type='auth'>
    <not-authorized xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</presence>
```

密码应该 SHOULD 通过进入房间时发送的出席信息节来提供，包含在满足‘<http://jabber.org/protocol/muc>’名字空间的 <x/> 元素的<password/>子元素里。密码以明码方式发送；目前不支持其它验证方法，而且任何这类的验证或授权方法都将会定义在一个独立的协议里(参见本文的[安全事项](#)章节)。

例子 26. 用户进入房间时提供密码

```
<presence
  from='hag66@shakespeare.lit/pda'
  to='darkcave@chat.shakespeare.lit/thirdwitch'>
  <x xmlns='http://jabber.org/protocol/muc'>
    <password>cauldronburn</password>
```

```
</x>
</presence>
```

仅限会员房间

如果房间是仅限会员的, 但用户不是(该房间的)成员, 服务必须 MUST 拒绝访问这个房间并通知用户它们不被允许进入房间; 具体方法是返回一个“error”类型的出席信息节, 并包含一个 <registration-required/> 错误条件:

例子 27. 服务拒绝访问, 因为用户不在成员列表中

```
<presence
  from='darkcave@chat.shakespeare.lit'
  to='hag66@shakespeare.lit/pda'
  type='error' >
  <x xmlns='http://jabber.org/protocol/muc' />
  <error type='auth' >
    <registration-required
xmlns='urn:iETF:params:xml:ns:xmpp-stanzas' />
  </error>
</presence>
```

被禁止的用户

如果用户已经被房间禁止(即, 其岗位为被排斥者 “outcast”), 服务必须 MUST 拒绝访问这个房间并通知用户他(她)被禁止了; 具体方法是返回一个出席信息节, 类型为“error”, 标明 <forbidden/> 错误条件:

例子 28. 服务拒绝访问, 因为用户被禁止了

```
<presence
  from='darkcave@chat.shakespeare.lit'
  to='hag66@shakespeare.lit/pda'
  type='error' >
  <x xmlns='http://jabber.org/protocol/muc' />
  <error type='auth' >
    <forbidden xmlns='urn:iETF:params:xml:ns:xmpp-stanzas' />
  </error>
</presence>
```

昵称冲突

如果房间里已经有别的用户使用了准备进入房间的新用户预期的昵称(或如果这个昵称被保留给另一个成员列表里面的用户), 服务必须 MUST 拒绝访问这个房

间并通知用户这个冲突；具体方法是返回一个出席信息节，类型为“error”，标明 <conflict/> 错误条件：

例子 29. 服务拒绝访问，因为昵称冲突

```
<presence
  from='darkcave@chat.shakespeare.lit'
  to='hag66@shakespeare.lit/pda'
  type='error' >
  <x xmlns='http://jabber.org/protocol/muc' />
  <error type='cancel' >
    <conflict xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</presence>
```

无论如何，如果现有房客的纯 JID <localpart@domain.tld> 和准备进入房间的用户纯 JID 相同，那么服务应该 SHOULD 允许这个用户的进入，所以这个用户就有两个(或更多)房间内的会话“sessions”使用同一个房间昵称，每一个对应一个资源。如果一个服务允许相同纯 JID 可以同时存在多个房客并使用同一个房间的房间昵称，它应该 SHOULD 路由房间内的消息给该用户的所有资源并允许用户的所有资源发送消息给房间；视实现而定，服务来决定如何适当的处理从用户的资源发送的出席信息以及如何路由私有消息到所有或某个资源(基于出席信息优先级或其他机制)。

如何确定昵称冲突取决于实现(例如，该服务是否应用于一个特定的惯例，一个 stringprep 规则如 Resourceprep 或 Nodeprep，等等)。

最大用户数

如果房间达到它的最大房客数量，服务应该 SHOULD 拒绝访问这个房间并通知该用户这个限制；方法是返回一个出席信息节，类型为“error”，标明 <service-unavailable/> 错误条件：

例子 30. 服务通知用户该房间已达到房客数量极限

```
<presence
  from='darkcave@chat.shakespeare.lit'
  to='hag66@shakespeare.lit/pda'
  type='error' >
  <x xmlns='http://jabber.org/protocol/muc' />
  <error type='wait' >
    <service-unavailable
      xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</presence>
```

另外，房间可以踢出空闲用户（“idle user”）以腾出空间。

如果房间的房客数量已达到最大值但是一个房间管理员或所有者试图进入，该房间应该允许管理员或所有者加入，为了使得额外的房客达到一个合理的数目，该数量可以 MAY 做成可配置的。

锁住的房间

如果一个用户尝试进入一个房间而该房间是锁住的“locked”（即，在房间创建者提供初始的配置之前以及也就是在房间正式存在之前），服务必须 MUST 拒绝进入并返回一个 <item-not-found/> 错误给该用户：

例子 31. 服务拒绝访问，因为房间不存在

```
<presence
  from='darkcave@chat.shakespeare.lit'
  to='hag66@shakespeare.lit/pda'
  type='error' >
  <x xmlns='http://jabber.org/protocol/muc' />
  <error type='cancel' >
    <item-not-found xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</presence>
```

不存在的房间

如果用户准备进入房间时，该房间已经不存在了，服务应该 SHOULD 建立它；无论如何，这不是必需的，因为一个实现或部署可以 MAY 选择限制建立房间的权限。详见本文的[新建房间](#)章节。

房间记录

如果用户进入一个房间，该房间的讨论是被记录到一个公开的存档里面（经常可以通过 HTTP 访问的），服务应该 SHOULD 允许该用户加入该房间但是必须 MUST 同时警告该用户讨论已被记录。方法是应该 SHOULD 在房间发送给该新房客的初始出席信息中包含一个状态码“170”：

例子 32. 服务发送新房客的出席信息给新房客

```
<presence
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='hag66@shakespeare.lit/pda' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
    <item affiliation='member' role='participant' />
    <status code='100' />
  </x>
</presence>
```



```
<status code=' 110' />
<status code=' 170' />
<status code=' 210' />
</x>
</presence>
```

讨论历史

如上发送完初始出席信息之后，一个房间可以 MAY 发送讨论历史给这个新房客。（在完成按照本文[出席信息广播](#)章节规定的发送房间出席信息之前，该房间不能 MUST NOT 发送任何讨论历史。）是否这个历史要被发送，以及这个历史里面包含多少条消息，将由聊天服务实现或特定的部署来决定。

例子 33. 讨论历史的发送

```
<message
  from='darkcave@chat.shakespeare.lit/firstwitch'
  to='hecate@shakespeare.lit/broom'
  type='groupchat'>
  <body>Thrice the brinded cat hath mew'd.</body>
  <delay xmlns='urn:xmpp:delay'
    from='cronel@shakespeare.lit/desktop'
    stamp='2002-10-13T23:58:37Z' />
</message>

<message
  from='darkcave@chat.shakespeare.lit/secondwitch'
  to='hecate@shakespeare.lit/broom'
  type='groupchat'>
  <body>Thrice and once the hedge-pig whined.</body>
  <delay xmlns='urn:xmpp:delay'
    from='wiccarocks@shakespeare.lit/laptop'
    stamp='2002-10-13T23:58:43Z' />
</message>

<message
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='hecate@shakespeare.lit/broom'
  type='groupchat'>
  <body>Harpier cries 'Tis time, 'tis time.</body>
  <delay xmlns='urn:xmpp:delay'
    from='hag66@shakespeare.lit/pda'
    stamp='2002-10-13T23:58:49Z' />
</message>
```

讨论历史消息必须 MUST 标为 [Delayed Delivery 11](#) 信息，满足 'urn:xmpp:delay' 名字空间，以表明它们是被延迟发送的并且标明它们最初发出的时间。（注意：'urn:xmpp:delay' 名字空间定义在 XEP-0203 里面，取代了旧的定义在 [Legacy Delayed Delivery 12](#) 里的 'jabber:x:delay' 名字空间；XEP-0091 状态更改为已过时之前，实现应该 SHOULD 包含两种日期时间 (datetime) 格式.）。在非匿名房间里，'from' 属性应该 SHOULD 是原始发送者的全 JID，但不能 MUST NOT 在半匿名房间里（在那里 'from' 属性应该 SHOULD 设置为房间本身的 JID）。服务应该 SHOULD 在进入该房间之后，发送任何即时（“live”）消息之前，发送完所有讨论历史消息。

管理讨论历史

用户可能 MAY 希望管理进入房间时（由房间）提供的讨论历史（可能因为用户带宽比较低或正在使用迷你客户端）。他必须 MUST 在加入房间时发出的初始出席信息节里包含一个 <history/> 子元素。这个元素有四个可用的属性：

表 8：历史管理属性

属性	数据类型	含义
maxchars	int	限制历史中的字符总数为“X”（这里的字符数量是全部 XML 节的字符数，不只是它们的 XML 字符数据）。
maxstanzas	int	限制历史中的消息总数为“X”。
seconds	int	仅发送最后 “X” 秒收到的消息。
since	dateTime	仅发送从指定日期时间 datetime 之后收到的消息（这个 datetime 必须 MUST 符合 XMPP Date and Time Profiles 13 定义的 DateTime 规则，）。

服务必须 MUST 发送满足以上条件组合的最小数量的消息，还要顾及服务级别和房间级别的缺省设置。服务必须 MUST 只发送完整的消息节（即，它不能 MUST not 按特定字符数把历史从字面上截断，但是必须 MUST 发送最大数量的完整节，这使得字符数小于或等于 'maxchars' 属性的值）。如果客户端不希望收到历史，它必须 MUST 把 'maxchars' 属性值设为“0”（zero）。

以下例子展示如何使用这个协议。

例子 34. 用户请求在历史中限制消息数量

```
<presence
  from='hag66@shakespeare.lit/pda'
  to='darkcave@chat.shakespeare.lit/thirdwitch'>
  <x xmlns='http://jabber.org/protocol/muc'>
    <history maxstanzas='20' />
  </x>
```

```
</presence>
```

例子 35. 用户请求最后三分钟的历史

```
<presence
  from='hag66@shakespeare.lit/pda'
  to='darkcave@chat.shakespeare.lit/thirdwitch'>
  <x xmlns='http://jabber.org/protocol/muc'>
    <history seconds='180' />
  </x>
</presence>
```

例子 36. 用户请求从 Unix 时代到现在的所有历史

```
<presence
  from='hag66@shakespeare.lit/pda'
  to='darkcave@chat.shakespeare.lit/thirdwitch'>
  <x xmlns='http://jabber.org/protocol/muc'>
    <history since='1970-01-01T00:00:00Z' />
  </x>
</presence>
```

服务绝对不应该 SHOULD NOT 返回从 Unix 时代开始到现在的所有消息，而应该 SHOULD 基于服务或房间的缺省值返回适当的有限数量的历史给用户。

例子 37. 用户请求不发送历史

```
<presence
  from='hag66@shakespeare.lit/pda'
  to='darkcave@chat.shakespeare.lit/thirdwitch'>
  <x xmlns='http://jabber.org/protocol/muc'>
    <history maxchars='0' />
  </x>
</presence>
```

退出一个房间

为了退出一个多用户聊天房间，一个房客发送一个类型为“unavailable”的出席信息节给正在使用这个房间的 <room@service/nick> 。

例子 38. 房客退出一个房间

```
<presence
  from='hag66@shakespeare.lit/pda'
```

```
to='darkcave@chat.shakespeare.lit/thirdwitch'  
type='unavailable' />
```

服务必须 MUST 接着从要离开的房客的房间 JID 发送“unavailable”类型的出席信息节给这个要离开的房客的全 JID 们以及留在房间的房客们：

例子 39. 服务发送和离开的房客有关的出席信息

```
<presence  
  from='darkcave@chat.shakespeare.lit/thirdwitch'  
  to='hag66@shakespeare.lit/pda'  
  type='unavailable' >  
  <x xmlns='http://jabber.org/protocol/muc#user' >  
    <item affiliation='member' role='none' />  
    <status code='110' />  
  </x>  
</presence>  
<presence  
  from='darkcave@chat.shakespeare.lit/thirdwitch'  
  to='cronel@shakespeare.lit/desktop'  
  type='unavailable' >  
  <x xmlns='http://jabber.org/protocol/muc#user' >  
    <item affiliation='member' role='none' />  
  </x>  
</presence>  
<presence  
  from='darkcave@chat.shakespeare.lit/thirdwitch'  
  to='wiccarocks@shakespeare.lit/laptop'  
  type='unavailable' >  
  <x xmlns='http://jabber.org/protocol/muc#user' >  
    <item affiliation='member' role='none' />  
  </x>  
</presence>
```

由房间反射的类型为“unavailable”的出席信息节必须 MUST 包含扩展的关于角色和岗位的出席信息；‘role’属性值应该 SHOULD 被设为“none”以表示这个人不再是一个房客了。

房客可以 MAY 在出席信息节包含一个常规的 <status/> 信息；这使房客能在必要的情况下提供一个自定的退出消息：

例子 40. 自定的退出消息

```
<presence  
  from='wiccarocks@shakespeare.lit/laptop'
```

```
to='darkcave@chat.shakespeare.lit/oldhag'  
type='unavailable'>  
<status>gone where the goblins go</status>  
</presence>
```

常规的出席信息节生成规则定义在 [XMPP IM 14](#)，所以如果用户发送一个一般的不可用出席信息节，用户的服务器将广播那个节到 <room@service/nick>，而该用户之前曾经发送过直接出席信息给这个<room@service/nick>。

有可能一个用户不能正常地通过直接发送不可用信息给一个房间来退出该房间。如果该用户没有发送不可用出席信息就下线了，用户的服务器负责代替该用户发送不可用出席信息（依据 [RFC 3921](#)）。如果该用户的服务器下线或该用户的服务器和该用户连接的MUC服务失去连接（例如，在联邦通信），这个MUC服务负责监视它收到的错误信息节以确定该用户是否下线。如果该MUC服务确定该用户已下线，它必须 `must` 当成该用户自己发送了不可用信息一样地处理这个用户。

注意：如果房间不是持久的并且该房客是最后一个退出的，服务负责销毁这个房间。

更改昵称

多用户聊天室的一个常用功能是一个房客能修改自己在房间里的昵称。在MUC里这需要发送一个更新出席信息给房间，具体来说是在相同的房间里发送出席信息给一个新的房间JID（变更的只是这个房间JID的资源）。

例子 41. 房客修改昵称

```
<presence  
  from='hag66@shakespeare.lit/pda'  
  to='darkcave@chat.shakespeare.lit/oldhag' />
```

服务接着发送两个出席信息节给每个房客的全JID（包括修改自己昵称的房客本身），一个是类型为“unavailable”的用于旧的昵称另一个指明新昵称可用了。

这个不可用出席信息必须 `MUST` 在一个满足 ['http://jabber.org/protocol/muc#user'](http://jabber.org/protocol/muc#user) 名字空间的 <x/> 子元素里面包含以下扩展的出席信息：

- 新昵称（在这个例子中，`nick='oldhag'`）
- 一个状态码 303

这使接受者能从旧昵称关联到新昵称。

例子 42. 服务更新昵称

```
<presence
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='cronel@shakespeare.lit/desktop'
  type='unavailable' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
    <item affiliation='member'
      jid='hag66@shakespeare.lit/pda'
      nick='oldhag'
      role='participant' />
    <status code='303' />
  </x>
</presence>
<presence
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='wiccarocks@shakespeare.lit/laptop'
  type='unavailable' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
    <item affiliation='member'
      jid='hag66@shakespeare.lit/pda'
      nick='oldhag'
      role='participant' />
    <status code='303' />
  </x>
</presence>
<presence
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='hag66@shakespeare.lit/pda'
  type='unavailable' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
    <item affiliation='member'
      jid='hag66@shakespeare.lit/pda'
      nick='oldhag'
      role='participant' />
    <status code='303' />
    <status code='110' />
  </x>
</presence>
<presence
  from='darkcave@chat.shakespeare.lit/oldhag'
  to='cronel@shakespeare.lit/desktop' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
```

```

    <item affiliation='member'
        jid='hag66@shakespeare.lit/pda'
        role='participant' />
</x>
</presence>
<presence
    from='darkcave@chat.shakespeare.lit/oldhag'
    to='wiccarocks@shakespeare.lit/laptop' >
    <x xmlns='http://jabber.org/protocol/muc#user' >
        <item affiliation='member'
            jid='hag66@shakespeare.lit/pda'
            role='participant' />
    </x>
</presence>
<presence
    from='darkcave@chat.shakespeare.lit/oldhag'
    to='hag66@shakespeare.lit/pda' >
    <x xmlns='http://jabber.org/protocol/muc#user' >
        <item affiliation='member'
            jid='hag66@shakespeare.lit/pda'
            role='participant' />
        <status code='110' />
    </x>
</presence>

```

如果该用户尝试修改他或她的房间昵称, 但这个昵称已经被其他用户使用了 (或者这个昵称是被这房间的其他用户岗位保留的, 例如, 一个成员或者所有者), 服务必须 MUST 拒绝这次昵称修改并通知该用户这一冲突; 也就是返回一个类型为 “error” 的出席信息节指明 <conflict/> 错误条件:

例子 43. 服务拒绝昵称修改, 因为昵称冲突

```

<presence
    from='darkcave@chat.shakespeare.lit'
    to='hag66@shakespeare.lit/pda'
    type='error' >
    <x xmlns='http://jabber.org/protocol/muc' />
    <error type='cancel' >
        <conflict xmlns='urn:iETF:params:xml:ns:xmpp-stanzas' />
    </error>
</presence>

```

无论如何, 如果现有房客纯 JID <localpart@domain.tld> 和尝试变更昵称的房客的纯 JID 相同, 那么服务可以 MAY 允许昵称变更. 详见本文的[昵称冲突](#)章节.

如果该用户尝试变更自己的昵称但是房间昵称被锁定了(“locked down”), 服务必须 MUST 拒绝这个昵称变更请求并返回一个“error”类型的出席信息节, 指明一个 <not-acceptable/> 错误条件:

例子 44. 服务拒绝昵称变更, 因为房间昵称被锁定

```
<presence
  from='darkcave@chat.shakespeare.lit'
  to='hag66@shakespeare.lit/pda'
  type='error' >
  <x xmlns='http://jabber.org/protocol/muc' />
  <error type='cancel' >
    <not-acceptable xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</presence>
```

用户应该 SHOULD 接着发现它的保留昵称, 如本文的 [发现保留的房间昵称](#) 章节所述.

更改可用性状态

在一个多用户聊天系统里例如 IRC, 一个常用的修改某人房间昵称的行为也意味着变更某人的可用性(例如, 变更某人的房间昵称为“thirdwitch|away”). 在 Jabber 里面, 可用性当然是通过出席信息(中 <show/> 和 <status/> 元素)的变更来通知的, 这能提供重要的上下文给聊天室. 一个房客通过发送更新的出席信息给它自己的<room@service/nick>来改变他在房间内的可用性状态.

例子 45. 房客变更可用性状态

```
<presence
  from='wiccarocks@shakespeare.lit/laptop'
  to='darkcave@chat.shakespeare.lit/oldhag' >
  <show>xa</show>
  <status>gone where the goblins go</status>
</presence>
```

服务然后从该房客发送一个出席信息节来修改他或她的出席信息给每个房客的全 JID, 包含扩展的出席信息, 包括这个房客的角色和全 JID(给那些有权知道的人):

例子 46. 服务传递修改的出席信息给所有房客

```
<presence
  from='darkcave@chat.shakespeare.lit/secondwitch'
```



```

    to='cronel@shakespeare.lit/desktop'>
<show>xa</show>
<status>gone where the goblins go</status>
<x xmlns='http://jabber.org/protocol/muc#user'>
  <item affiliation='admin'
        jid='wiccarocks@shakespeare.lit/laptop'
        role='moderator' />
</x>
</presence>

[ ... ]

```

邀请其他用户进入一个房间

直接邀请

一个办法是发送一个直接的邀请(而不是由房间本身来间接邀请), 定义在 [Direct MUC Invitations 15](#). 直接发送邀请有助于适应被邀请者那一边的通信阻塞(对方可能拒绝和和不在好友名单中的实体通信).

间接邀请

邀请别的用户到一个房间成为房客是很有用的. 为了做到这一点, 一个 MUC 客户端必须 MUST 发送以下格式的 XML 给 <room@service> 本身(原因(reason)是可选的 OPTIONAL 而消息(message)的类型必须 MUST 是显式或隐式的 "normal"类型):

例子 47. 房客通过房间发送一个邀请

```

<message
  from='cronel@shakespeare.lit/desktop'
  to='darkcave@chat.shakespeare.lit'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <invite to='hecate@shakespeare.lit'>
      <reason>
        Hey Hecate, this is the place for all good witches!
      </reason>
    </invite>
  </x>
</message>

```

<room@service> 本身必须 MUST 接着增加一个 'from' 地址到 <invite/> 元素, 其值为邀请者的纯 JID, 全 JID, 或房间 JID, 并发送邀请给 'to' 地址所指明的被邀请者(为了旧的客户端, 服务可以 MAY 包含一个消息主体"message

body”解释这个邀请或包含一个原因”reason”(子元素)；另外，房间应该 SHOULD 增加 password 如果该房间是密码保护的)：

例子 48. 房间代表邀请者发送邀请给被邀请者

```
<message
  from='darkcave@chat.shakespeare.lit'
  to='hecate@shakespeare.lit'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <invite from='cronel@shakespeare.lit/desktop'>
      <reason>
        Hey Hecate, this is the place for all good witches!
      </reason>
    </invite>
    <password>cauldronburn</password>
  </x>
</message>
```

如果房间是仅限成员的，服务可以 MAY 同时把这个被邀请者加入成员列表。(注意：在仅限成员的房间邀请的权力应该 SHOULD 由房间管理员限定；如果一个没有权限的成员修改成员列表试图邀请别的用户，服务应该 SHOULD 返回一个 <forbidden/> 错误给该房客；详见本文的[修改成员列表](#)章节。)

如果邀请者提供了一个不存在的 JID，房间应该 SHOULD 返回一个 <item-not-found/> 错误给邀请者。

被邀请者可以 MAY 选择正式地拒绝（反之则忽略）邀请；这是发送者希望看到的正式的通知。为了拒绝这个邀请，被邀请者必须 MUST 发送以下格式的消息给 <room@service> 本身：

例子 49. 被邀请者谢绝邀请

```
<message
  from='hecate@shakespeare.lit/broom'
  to='darkcave@chat.shakespeare.lit'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <decline to='cronel@shakespeare.lit'>
      <reason>
        Sorry, I'm too busy right now.
      </reason>
    </decline>
  </x>
</message>
```

例子 50. 房间通知邀请者邀请被拒绝了

```

<message
  from='darkcave@chat.shakespeare.lit'
  to='cronel@shakespeare.lit/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <decline from='hecate@shakespeare.lit'>
      <reason>
        Sorry, I'm too busy right now.
      </reason>
    </decline>
  </x>
</message>

```

可能(有人)想知道为什么被邀请者不直接发送拒绝消息给访问者. 主要原因是特定的实现可能 MAY 选择让邀请基于房间 JIDs 而不是纯 JIDs (所以, 例如, 一个房客可能从一个房间邀请某人到另一个房间而不需要知道这个人的纯 JID). 因而服务必须 MUST 同时处理邀请和拒绝.

把一对一聊天转为多用户会议

有时候人们需要把一个一对一的聊天转成一个多用户的会议. 以下例子展示了这个流程.

首先, 两个用户开始一个一对一聊天.

例子 51. 一个一对一聊天

```

<message
  from='cronel@shakespeare.lit/desktop'
  to='wiccarocks@shakespeare.lit/laptop'
  type='chat'>
  <thread>e0ffe42b28561960c6b12b944a092794b9683a38</thread>
  <body>Thrice the brinded cat hath mew'd.</body>
</message>

<message
  from='wiccarocks@shakespeare.lit/laptop'
  to='cronel@shakespeare.lit/desktop'
  type='chat'>
  <thread>e0ffe42b28561960c6b12b944a092794b9683a38</thread>
  <body>Thrice and once the hedge-pig whined.</body>
</message>

```

现在第一个用户决定加入第三个人到这个讨论, 所以她 (或, 更准确地说, 她的客户端) 做以下事情:

1. 新建一个多用户聊天室
2. 可选地发送一对一聊天的历史到房间
3. 发送一个邀请给第二个人和第三个人，包含一个 `<continue/>` 元素（可选地包含一个 `'thread'` 属性）。

注意：新房间应该 SHOULD 是非匿名的，可以 MAY 是一个即时房间（定义于本文的[新建即时房间](#)章节），也可以 MAY 有一个从服务接收的唯一房间名（定义于本文的[请求唯一的房间名](#)章节）。

注意：如果这个一对一的聊天消息包含了一个 `<thread/>` 元素，这个新建房间的人应该 SHOULD 在历史消息中包含这个 ThreadID，在邀请中把这个 ThreadID 的值赋予 `<continue/>` 元素的 `'thread'` 属性，并把这 ThreadID 包含在任何新的消息中发送到房间。ThreadIDs 的使用是推荐的 RECOMMENDED，因为它帮助提供一对一聊天和多用户聊天的连续性。

例子 52. 继续讨论 I: 用户新建房间

```
<presence
  from='cronel@shakespeare.lit/desktop'
  to='darkcave@chat.shakespeare.lit/firstwitch'>
  <x xmlns='http://jabber.org/protocol/muc' />
</presence>

<presence
  from='darkcave@chat.shakespeare.lit/firstwitch'
  to='cronel@shakespeare.lit/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='owner' role='moderator' />
    <status code='110' />
  </x>
</presence>
```

例子 53. 继续讨论 II: 所有者发送历史到房间

```
<message
  from='cronel@shakespeare.lit/desktop'
  to='darkcave@chat.shakespeare.lit'
  type='groupchat'>
  <thread>e0ffe42b28561960c6b12b944a092794b9683a38</thread>
  <body>Thrice the brinded cat hath mew'd.</body>
  <delay xmlns='urn:xmpp:delay'
    from='cronel@shakespeare.lit/desktop'
    stamp='2004-09-29T01:54:37Z' />
</message>
```

```

<message
  from='cronel@shakespeare.lit/desktop'
  to='darkcave@chat.shakespeare.lit'
  type='groupchat'>
  <thread>e0ffe42b28561960c6b12b944a092794b9683a38</thread>
  <body>Thrice and once the hedge-pig whined.</body>
  <delay xmlns='urn:xmpp:delay'
    from='wiccarocks@shakespeare.lit/laptop'
    stamp='2004-09-29T01:55:21Z' />
</message>

```

注意：使用 Delayed Delivery 协议使房间创建者能够从他一对一聊天历史指明每个消息的日期时间 datetime（通过 'stamp' 属性），以及每个消息的原始发送者的 JID（通过 'from' 属性）。房间创建者应该 SHOULD 在邀请额外的用户到房间之前发送完整的一对一聊天历史，并且也应该 SHOULD 把第二个人加入该房间之前和第一个人在一对一聊天界面中出现的任何消息当成历史来发送；如果这个一对一历史特别的大，发送的客户端可能希望在数秒内发送这个历史而不是一次性发送所有历史（以 to 避免触发频率限制）。服务不应该 SHOULD NOT 在从房间所有者接收的历史消息之前添加它自己的延迟元素“delay elements”（见本文的[讨论历史](#)章节）。

例子 54. 继续讨论 III：所有者发送邀请(们)，包含 Continue 标志

```

<message
  from='cronel@shakespeare.lit/desktop'
  to='darkcave@chat.shakespeare.lit'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <invite to='wiccarocks@shakespeare.lit/laptop'>
      <reason>This coven needs both wiccarocks and hag66.</reason>
      <continue thread='e0ffe42b28561960c6b12b944a092794b9683a38' />
    </invite>
    <invite to='hag66@shakespeare.lit'>
      <reason>This coven needs both wiccarocks and hag66.</reason>
      <continue thread='e0ffe42b28561960c6b12b944a092794b9683a38' />
    </invite>
  </x>
</message>

```

注意：当邀请者的客户端一知道和它一对一聊天的那个人的全 JID 之后，它就应该 SHOULD 在邀请中包含这个全 JID（而不是纯 JID）。

邀请被递送到被邀请者：

例子 55. 邀请被递送

```

<message
  from='darkcave@chat.shakespeare.lit'>
  to='wiccarocks@shakespeare.lit/laptop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <invite from='cronel@shakespeare.lit'>
      <reason>This coven needs both wiccarocks and hag66.</reason>
      <continue thread='e0ffe42b28561960c6b12b944a092794b9683a38' />
    </invite>
  </x>
</message>

```

```

<message
  from='darkcave@chat.shakespeare.lit'>
  to='hag66@shakespeare.lit'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <invite from='cronel@shakespeare.lit'>
      <reason>This coven needs both wiccarocks and hag66.</reason>
      <continue thread='e0ffe42b28561960c6b12b944a092794b9683a38' />
    </invite>
  </x>
</message>

```

当客户端被 <wiccarocks@shakespeare.lit/laptop> 用来接收邀请，它应该 SHOULD 自动加入或提示用户是否加入（取决于用户的选项配置）并且随后无缝地把现有的一对一聊天窗口转到一个多用户会议的窗口：

例子 56. 被邀请者接受邀请，加入房间，并接收出席信息和历史

```

<presence
  from='wiccarocks@shakespeare.lit/laptop'
  to='darkcave@chat.shakespeare.lit/secondwitch'>
  <x xmlns='http://jabber.org/protocol/muc' />
</presence>

```

```

<presence
  from='darkcave@chat.shakespeare.lit/firstwitch'
  to='wiccarocks@shakespeare.lit/laptop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='owner' role='moderator' />
  </x>
</presence>

```

```

<presence
  from='darkcave@chat.shakespeare.lit/secondwitch'
  to='wiccarocks@shakespeare.lit/laptop'>

```

```

<x xmlns='http://jabber.org/protocol/muc#user'>
  <item affiliation='member' role='participant' />
</x>
</presence>

<message
  from='darkcave@chat.shakespeare.lit'
  to='wiccarocks@shakespeare.lit/laptop'
  type='groupchat'>
  <thread>e0ffe42b28561960c6b12b944a092794b9683a38</thread>
  <body>Thrice the brinded cat hath mew'd.</body>
  <delay xmlns='urn:xmpp:delay'
    from='cronel@shakespeare.lit/desktop'
    stamp='2004-09-29T01:54:37Z' />
</message>

<message
  from='darkcave@chat.shakespeare.lit'
  to='wiccarocks@shakespeare.lit/laptop'
  type='groupchat'>
  <thread>e0ffe42b28561960c6b12b944a092794b9683a38</thread>
  <body>Thrice and once the hedge-pig whined.</body>
  <delay xmlns='urn:xmpp:delay'
    from='wiccarocks@shakespeare.lit/laptop'
    stamp='2004-09-29T01:55:21Z' />
</message>

```

注意：事实上，这些消息从 <room@service> 本身而不是 <room@service/nick> 发出，告诉这些接收的客户端这些消息是优先的聊天历史，因为任何来自房客的消息的 'from' 地址应该等于发送者的房间 JID.

房客修改房间标题

如果房间配置允许，一个房客可以 MAY 被允许修改一个房间的主题。详见本文的[修改房间主题](#)章节。

发送私有消息

因为每个房客有一个唯一的房间 JID，一个房客可以 MAY 发送一个私有消息 “private message” 给选定的房客，即通过服务发送一个消息给那房客的房间 JID. 这个消息类型应该 SHOULD 是 “chat” 并且不能 MUST NOT 是 “groupchat”，但是可以 MAY 不表明（即，一个常规 “normal” 消息）。这个权力应该 SHOULD 被任何房客允许（甚至在一个被主持的房间里的游客）。

例子 57. 房客发送私有消息

```
<message
  from='wiccarocks@shakespeare.lit/laptop'
  to='darkcave@chat.shakespeare.lit/firstwitch'
  type='chat'>
  <body>I'll give thee a wind.</body>
</message>
```

服务负责把 'from' 地址改为发送者的房间 JID 并递送这个消息到预期的接收者的全 JID.

例子 58. 接收者接收私有消息

```
<message
  from='darkcave@chat.shakespeare.lit/secondwitch'
  to='cronel@shakespeare.lit/desktop'
  type='chat'>
  <body>I'll give thee a wind.</body>
</message>
```

如果发送者尝试发送一个类型为 "groupchat" 的私有消息给特定的房客，服务必须 MUST 拒绝递送这个消息（因为接收者的客户端期望的房间内的消息类型为 "groupchat"）并且返回一个 <bad-request/> 错误给发送者：

例子 59. 房客尝试发送类型为 "Groupchat" 的私有消息给特定的房客

```
<message
  from='wiccarocks@shakespeare.lit/laptop'
  to='darkcave@chat.shakespeare.lit/firstwitch'
  type='groupchat'>
  <body>I'll give thee a wind.</body>
</message>

<message
  from='darkcave@chat.shakespeare.lit'
  to='wiccarocks@shakespeare.lit/laptop'
  type='error'>
  <body>I'll give thee a wind.</body>
  <error type='modify'>
    <bad-request xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</message>
```


如果发送者尝试发送一个私有消息给一个不存在的房间 JID，服务必须 MUST 返回一个 `<item-not-found/>` 错误给发送者。

如果发送者不是预期的接收者正在访问的那个房间的房客，服务必须 MUST 返回一个 `<not-acceptable/>` 错误给发送者。

发送消息给所有房客

房客发送一个消息给所有房间内的房客的方法，是发送一个类型为 “groupchat” 的消息到 `<room@service>` 本身（服务可以 MAY 忽略或拒绝类型不是 “groupchat” 的消息）。在一个被主持的房间，这个权力限于角色为与会者或更高的房客拥有。

例子 60. 房客发送一个消息给所有房客

```
<message
  from='hag66@shakespeare.lit/pda'
  to='darkcave@chat.shakespeare.lit'
  type='groupchat'>
  <body>Harpier cries: 'tis time, 'tis time.</body>
</message>
```

如果发送者在这个房间有发言权（在被主持的房间里缺省是这样期望），服务必须 MUST 修改发送者的 ‘from’ 属性成为房间 JID 并反射这个消息到每个房客的全 JID。

例子 61. 服务反射消息给所有房客

```
<message
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='cronel@shakespeare.lit/desktop'
  type='groupchat'>
  <body>Harpier cries: 'tis time, 'tis time.</body>
</message>
<message
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='wiccarocks@shakespeare.lit/laptop'
  type='groupchat'>
  <body>Harpier cries: 'tis time, 'tis time.</body>
</message>
<message
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='hag66@shakespeare.lit/pda'
  type='groupchat'>
```

```
<body>Harpier cries: 'tis time, 'tis time.</body>
</message>
```

如果发送者是个游客 即，在一个被主持的房间里没有发言权)，服务可以 MAY 返回一个 <forbidden/> 错误给发送者并且不能 MUST NOT 反射这个消息给所有房客。如果发送者不是该房间的房客，服务应该 SHOULD 返回一个 <not-acceptable/> 错误给发送者并且不应该 SHOULD NOT 反射这个消息给所有房客；这个规则的唯一例外是，一个实现可以 MAY 允许用户们拥有特定的权限（例如，一个房间拥有者，房间管理员，或服务级别的管理员）发送消息到这个房间，即使那些用户不是房客。

注册到房间

一个实现可以 MAY 允许一个无岗位的用户（在一个被主持的房间里，通常是一个与会者）注册一个房间从而成为该房间的一个成员（反之，一个实现也可以 MAY 限制这个权力并且只允许房间管理员添加新的成员）。特别是，不在成员列表的人是无法加入一个仅限会员的房间的，所以为了加入这样一个房间，实体需要申请会籍。

如果允许，这个功能应该 SHOULD 这样被实现。让用户使用 'jabber:iq:register' 名字空间[带内注册 16](#)提出注册申请给房间，：

例子 62. 用户提出注册申请

```
<iq from='hag66@shakespeare.lit/pda'
  id='reg1'
  to='darkcave@chat.shakespeare.lit'
  type='get'>
  <query xmlns='jabber:iq:register' />
</iq>
```

如果用户提出的注册申请不被允许注册该房间（例如，因为那个权限被限制了），该房间必须 MUST 返回一个 <not-allowed/> 错误给该用户。如果该用户已经注册过了，房间必须 MUST 返回一个类型为“result”的 IQ 节并包含一个空的 <register/>元素（定义于 [XEP-0077](#)）。如果该房间不存在，服务必须 MUST 返回一个 <item-not-found/> 错误。

否则，房间必须 MUST 接着返回一个数据表单“Data Form”给该用户（定义于[数据表单 17](#)）。注册需要的信息可以 MAY 根据实现和部署的不同而不同并且没有完全定义在本文中（例如，本文根据 'http://jabber.org/protocol/muc#register' 名字空间采用的注册字段 FORM_TYPE 可能将来会根据[字段标准化](#)章节里描述的得到补充，）。以下是一个典型的例子：

例子 63. 服务返回注册表单

```
<iq from='darkcave@chat.shakespeare.lit'
  id='reg1'
  to='hag66@shakespeare.lit/pda'
  type='result'>
  <query xmlns='jabber:iq:register'>
    <instructions>
      To register on the web, visit http://shakespeare.lit/
    </instructions>
    <x xmlns='jabber:x:data' type='form'>
      <title>Dark Cave Registration</title>
      <instructions>
        Please provide the following information
        to register with this room.
      </instructions>
      <field
        type='hidden'
        var='FORM_TYPE'>
        <value>http://jabber.org/protocol/muc#register</value>
      </field>
      <field
        label='Given Name'
        type='text-single'
        var='muc#register_first'>
        <required/>
      </field>
      <field
        label='Family Name'
        type='text-single'
        var='muc#register_last'>
        <required/>
      </field>
      <field
        label='Desired Nickname'
        type='text-single'
        var='muc#register_roomnick'>
        <required/>
      </field>
      <field
        label='Your URL'
        type='text-single'
        var='muc#register_url' />
      <field
```

```

        label='Email Address'
        type='text-single'
        var='muc#register_email' />
    <field
        label='FAQ Entry'
        type='text-multi'
        var='muc#register_faentry' />
    </x>
</query>
</iq>

```

用户应该 SHOULD 接着提交这个表单:

例子 64. 用户提交注册表单

```

<iq from='hag66@shakespeare.lit/pda'
    id='reg2'
    to='darkcave@chat.shakespeare.lit'
    type='set'>
    <query xmlns='jabber:iq:register'>
        <x xmlns='jabber:x:data' type='submit'>
            <field var='FORM_TYPE'>
                <value>http://jabber.org/protocol/muc#register</value>
            </field>
            <field var='muc#register_first'>
                <value>Brunhilde</value>
            </field>
            <field var='muc#register_last'>
                <value>Entwhistle-Throckmorton</value>
            </field>
            <field var='muc#register_roomnick'>
                <value>thirdwitch</value>
            </field>
            <field var='muc#register_url'>
                <value>http://witchesonline/~hag66/</value>
            </field>
            <field var='muc#register_email'>
                <value>hag66@witchesonline</value>
            </field>
            <field var='muc#register_faentry'>
                <value>Just another witch.</value>
            </field>
        </x>
    </query>
</iq>

```

如果期望的房间昵称已经被那个房间保留，房间必须 MUST 返回一个 `<conflict/>` 错误给该用户：

例子 65. 房间返回冲突错误给用户

```
<iq from='darkcave@chat.shakespeare.lit'
  id='reg2'
  to='hag66@shakespeare.lit/pda'
  type='error' >
  <error type='cancel' >
    <conflict xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>
```

如果该房间或服务不支持注册，它必须 MUST 返回一个 `<service-unavailable/>` 错误给用户：

例子 66. 房间返回服务不可用错误给用户

```
<iq from='darkcave@chat.shakespeare.lit'
  id='reg2'
  to='hag66@shakespeare.lit/pda'
  type='error' >
  <error type='cancel' >
    <service-unavailable
xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>
```

如果用户没有提交合法的数据表格，房间必须 MUST 返回一个 `<bad-request/>` 错误给用户：

例子 67. 房间返回“服务错误的请求”错误给用户

```
<iq from='darkcave@chat.shakespeare.lit'
  id='reg2'
  to='hag66@shakespeare.lit/pda'
  type='error' >
  <error type='modify' >
    <bad-request xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>
```

否则，该房间必须 MUST 通知用户注册请求被成功地接收到了：

例子 68. 房间通知用户注册请求已经被处理了

```
<iq from='darkcave@chat.shakespeare.lit'  
  id='reg2'  
  to='hag66@shakespeare.lit/pda'  
  type='result' />
```

用户提交表单之后，服务可以 MAY 向一个房间 管理员/所有者 请求批准该申请（参见本文的[批准注册申请](#)章节）或也可以 MAY 立刻把该用户的岗位从“none”变更为“member”来添加此用户到成员列表。如果服务变更了该用户的岗位并且该用户在房间里，它必须 MUST 从这个用户发送更新的出席信息给所有房客，声明岗位的变更，这个更新的出席信息应包含一个满足 ['http://jabber.org/protocol/muc#user'](http://jabber.org/protocol/muc#user) 名字空间 `<x/>` 元素并包含一个 `'affiliation'` 属性值设为“member”的 `<item/>` 子元素。

例子 69. 服务发送成员变更通知给所有房客

```
<presence  
  from='darkcave@chat.shakespeare.lit/thirdwitch'  
  to='cronel@shakespeare.lit/desktop'>  
  <x xmlns='http://jabber.org/protocol/muc#user'>  
    <item affiliation='member'  
      jid='hag66@shakespeare.lit/pda'  
      role='participant' />  
  </x>  
</presence>  
  
[ ... ]
```

如果一个用户已经注册到一个房间，该房间可以 MAY 选择限制这个用户在那个房间仅能使用已注册的昵称。如果它这样做，当用户尝试以不同于该用户之前已注册的房间昵称来加入该房间（这使房间锁定“lock down”房间昵称以保证房客身份的一致性）的时候，它应该 SHOULD 返回一个 `<not-acceptable/>` 错误给该用户。

获取成员列表

根据房间配置如果允许的话，一个房客可以 MAY 被允许接收房间成员的列表。详见本文的[修改成员列表](#)章节。

发现保留的房间昵称

一个用户可以 MAY 有一个保留的房间昵称，例如通过显式的房间注册，数据库集成，或昵称锁定 “lockdown”。用户应该 SHOULD 在尝试进入该房间之前发现自己的保留昵称。这可以通过发送一个发现服务信息请求并指定一个服务发现节点“x-roomuser-item”给房间 JID 来做到。

例子 70. 用户请求保留的昵称

```
<iq from='hag66@shakespeare.lit/pda'
  id='getnick1'
  to='darkcave@chat.shakespeare.lit'
  type='get'>
  <query xmlns='http://jabber.org/protocol/disco#info'
    node='x-roomuser-item' />
</iq>
```

对一个多用户聊天服务来说，对上述的服务发现节点的支持是可选的 OPTIONAL。如果房间或服务不支持上述的服务发现节点，它必须 MUST 返回一个 <feature-not-implemented/> 错误给用户。如果它支持这个特性并且该用户有一个已注册的昵称，它必须 MUST 返回这个昵称给这个用户，方法是发送一个服务发现的<identity/>元素，其‘name’属性值为这个昵称（此处 category/type 应该 SHOULD 是 “conference/text”）：

例子 71. 房间返回昵称

```
<iq from='darkcave@chat.shakespeare.lit'
  id='getnick1'
  to='hag66@shakespeare.lit/pda'
  type='result'>
  <query xmlns='http://jabber.org/protocol/disco#info'
    node='x-roomuser-item'>
    <identity
      category='conference'
      name='thirdwitch'
      type='text' />
  </query>
</iq>
```

如果该用户没有已注册的昵称，房间必须 MUST 返回一个空的服务发现 <query/> 元素（根据 XEP-0030）。

即使一个用户已经注册了一个房间昵称，服务应该 SHOULD 允许该用户在加入该房间时指定一个不同的昵称（例如，为了从不同的客户端资源加入），尽管该服务可以 MAY 选择通过一个 <not-acceptable/> 错误来锁定 “lock down” 昵称并拒绝该用户。如果该用户的客户端在加入该房间之后发送上述请求，服务不能 MUST NOT 返回一个错误给该用户，而应该 SHOULD 返回上文所述。

如果另一个用户尝试以第一个用户保留的房间昵称来加入房间，服务必须 MUST 拒绝第二个用户并返回一个前文所述的 <conflict/> 错误。

申请发言权

在一个被主持的房间里游客是不能发言的（即，发送一个消息给所有房客）。为了申请发言权，一个游客应该 SHOULD 发送包含一个数据表格的 <message/> 节给房间本身，这个数据表格仅仅是一个 'muc#role' 字段，值为 "participant"。

例子 72. 房客申请发言权

```
<message from='hag66@shakespeare.lit/pda'
         to='darkcave@chat.shakespeare.lit'>
  <x xmlns='jabber:x:data' type='submit'>
    <field var='FORM_TYPE'>
      <value>http://jabber.org/protocol/muc#request</value>
    </field>
    <field var='muc#role'
            type='text-single'
            label='Requested role'>
      <value>participant</value>
    </field>
  </x>
</message>
```

服务接着应该 SHOULD 转发这个请求给房间主持人(们)，定义于本文的[批准发言权申请](#)。

主持人用例

一个主持人有权在房间里执行特定的动作（例如，变更某些房客的角色）但无权变更岗位的持久信息（它只能被管理员或所有者）或定义关于这个房间的信息。具体哪些动作可由主持人执行，取决于配置。无论如何，对于 MUC 框架来说，主持人被规定有权执行以下动作：

1. 在一个半匿名的房间里发现一个房客的全 JID(如上文所述缺省会发生)
2. 修改主题
3. 从该房间踢出一个与会者或游客
4. 在一个被主持的房间里授予或撤销发言权
5. 在一个被主持的房间里修改拥有发言权的房客列表

这些特性将通过一个基于 `<iq/>` 元素的 请求/应答 交换来实现, 这个 IQ 元素包含一个满足 '<http://jabber.org/protocol/muc#admin>' 名字空间的子元素. 以下例子展示这个协议和实现如何互动达到期望的功能. (以下除非显式地提及, 任何接下来的管理请求必须 MUST 被拒绝, 如果该请求的 'from' 地址 `<user@host>` 和主持人的纯 JID 不符的话; 在这种情况下, 服务必须 MUST 返回一个 `<forbidden/>` 错误.)

修改房间主题

多用户聊天室的一个常用特性是变更房间主题的能力. 缺省地, 一个房间里只有角色为主持人 "moderator" 的用户应该 SHOULD 被允许变更主题 (尽管这应该 SHOULD 是可配置的, 结果是如果需要的话, 仅仅与会者或甚至游客都被允许修改主题). 主题变更是通过发送一个类型为 "groupchat" 的消息给 `<room@service>` 来实现的, 在这里 `<message/>` 必须 MUST 包含一个 `<subject/>` 元素以指定新的主题, 但不应该 SHOULD NOT 包含其他元素 (例如, 不应该有 `<body/>` 元素或 `<thread/>` 元素).

例子 73. 主持人变更主题

```
<message
  from='wiccarocks@shakespeare.lit/laptop'
  to='darkcave@chat.shakespeare.lit'
  type='groupchat'>
  <subject>Fire Burn and Cauldron Bubble!</subject>
</message>
```

如果一个 MUC 服务接收到这样一个消息, 它必须 MUST 以发送这个变更主题消息的那个用户的房间 JID 作为 'from' 地址来反射这个消息给所有其他房客:

例子 74. 服务通知所有房客主题变更

```
<message
  from='darkcave@chat.shakespeare.lit/secondwitch'
  to='cronel@shakespeare.lit/desktop'
  type='groupchat'>
  <subject>Fire Burn and Cauldron Bubble!</subject>
</message>
```

[...]

另外, 当一个新的房客加入房间时, 该房间应该 SHOULD 在被发送的讨论历史中包含最后的主题变更.

一个接收到这类信息的 MUC 客户端可以 MAY 选择显示一个房间内的消息，如下：

例子 75. 客户端显式房间主题变更消息

```
* secondwitch has changed the subject to: Fire Burn and Cauldron Bubble!
```

如果一些没有适当权限的人尝试变更房间主题，服务必须 MUST 返回一个 “error” 类型的消息指明一个 <forbidden/> 错误条件：

例子 76. 服务返回未被授权变更主题的错误

```
<message
  from='darkcave@chat.shakespeare.lit'
  to='hag66@shakespeare.lit/pda'
  type='error'>
  <subject>Fire Burn and Cauldron Bubble!</subject>
  <error type='auth'>
    <forbidden xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</message>
```

为了移除现有的主题而不是提供一个新主题（即，设置主题为空），客户端应该发送一个空的 <subject/> 元素（即，“<subject/>” 或 “<subject></subject>”）。

例子 77. 主持人设置空的主题

```
<message
  from='wiccarocks@shakespeare.lit/laptop'
  to='darkcave@chat.shakespeare.lit'
  type='groupchat'>
  <subject></subject>
</message>
```

踢出房客

主持人有权从一个房间踢出特定种类的房客（哪些房客是可被踢的 “kickable” 取决于服务规定，房间配置，以及主持人的岗位 -- 见下文）。踢人通常基于房客的房间昵称来执行（尽管可以 MAY 基于全 JID）并且完全是通过把与会者或游客的角色设为 “none” 来实现的。

例子 78. 主持人踢出房客

```

<iq from='fluellen@shakespeare.lit/pda'
  id='kick1'
  to='harfleur@henryv.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item nick='pistol' role='none'>
      <reason>Avaunt, you cullion!</reason>
    </item>
  </query>
</iq>

```

服务必须 MUST 移除被踢的用户，通过发送一个类型为“unavailable”的出席信息节给每个被踢的房客，这个出席信息应在其扩展出席信息中包含状态码 307，或(可选地)包含 reason 子元素(如果提供了)以及踢人的执行者的纯 JID.

例子 79. 服务移除被踢的房客

```

<presence
  from='harfleur@henryv.shakespeare.lit/pistol'
  to='pistol@shakespeare.lit/harfleur'
  type='unavailable'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='none' role='none'>
      <actor jid='fluellen@shakespeare.lit' />
      <reason>Avaunt, you cullion!</reason>
    </item>
    <status code='307' />
  </x>
</presence>

```

包含状态码可使客户端能够提交他们自己的通知消息(例如,适当的用户位置的信息). 可选的包含原因 reason 元素以及执行者 actor 使得被踢的用户能理解为什么他或她被踢了, 以及被踢的用户可以找谁去理论. [18](#)

移除被踢的房客(们)之后, 服务必须 MUST 接着通知主持人成功了:

例子 80. 服务通知主持人成功了

```

<iq from='harfleur@henryv.shakespeare.lit'
  id='kick1'
  to='fluellen@shakespeare.lit/pda'
  type='result' />

```

通知主持人之后，服务必须 MUST 接着通知剩余的房客那个被踢的房客已经不在房间里了，即从被踢者的房间昵称(<room@service/nick>)发送“unavailable”类型的出席信息节给所有剩余的房客（就像房客自愿退出房间时所做的一样），包含状态码 status 以及可选的原因 reason 和执行者 actor.

例子 81. 服务通知剩余的房客

```
<presence
  from='harfleur@henryv.shakespeare.lit/pistol'
  to='gower@shakespeare.lit/cell'
  type='unavailable'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='none' role='none' />
    <status code='307' />
  </x>
</presence>

[ ... ]
```

一个用户不能被比自己岗位低的主持人踢出。所以，如果一个身为与会者的主持人尝试踢出一个管理员，或一个身为与会者的主持人或管理员尝试踢出一个所有者，服务必须 MUST 拒绝这个请求并返回一个 <not-allowed/> 错误给发送者：

例子 82. 服务对于尝试踢出更高岗位的用户返回错误

```
<iq from='darkcave@chat.shakespeare.lit'
  id='kicktest'
  to='wiccarocks@shakespeare.lit/laptop'
  type='error'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item nick='firstwitch' role='none'>
      <reason>Be gone!</reason>
    </item>
  </query>
  <error type='cancel'>
    <not-allowed xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>
```

如果一个主持人尝试踢出他自己，服务可以 MAY 拒绝这个请求并返回一个 <conflict/> 错误给发送者。（尽管这个踢出自己的行为可能看起来怪异，它在 IRC 里很常见，用于在房间里为某人的行为道歉。）

授予游客发言权

在一个被主持的房间里，主持人可能希望管理房间内谁有水没有发言权 “voice”（即，发送消息给所有房客的能力）。发言权的授予是基于游客的房间昵称来的，服务将从内部把这个房间昵称转成游客的全 JID。主持人通过把游客的角色变更为与会者 “participant”来给一个游客授予权限。

例子 83. 主持人授予权限给一个游客

```
<iq from='cronel@shakespeare.lit/desktop'
  id='voicel'
  to='darkcave@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item nick='thirdwitch'
      role='participant' />
  </query>
</iq>
```

<reason/> 元素是可选的 OPTIONAL:

例子 84. 主持人授予权限给一个游客(包含一个原因 Reason)

```
<iq from='cronel@shakespeare.lit/desktop'
  id='voicel'
  to='darkcave@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item nick='thirdwitch'
      role='participant'>
      <reason>A worthy witch indeed!</reason>
    </item>
  </query>
</iq>
```

服务必须 MUST 接着通知主持人成功了:

例子 85. 服务通知主持人成功了

```
<iq from='darkcave@chat.shakespeare.lit'
  id='voicel'
  to='cronel@shakespeare.lit/desktop'
  type='result' />
```

服务必须 MUST 接着以这个人的<room@service/nick>发送更新的出席信息给所有房客，在这个出席信息里包含了一个满足'<http://jabber.org/protocol/muc#user>' 名字空间的<x/>元素，<x/>元素则包含一个<item/>子元素，其'role' 属性值为"participant"，指明添加了发言权。

例子 86. 服务发送发言权通知给所有房客

```
<presence
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='cronel@shakespeare.lit/desktop' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
    <item affiliation='member'
      nick='thirdwitch'
      role='participant' >
      <reason>A worthy witch indeed!</reason>
    </item>
  </x>
</presence>

[ ... ]
```

撤销与会者发言权

在一个被主持的房间里，主持人可能希望撤销一个与会者发言的权力，主持人通过把与会者的角色变更为游客 "visitor"来撤销一个游客的发言权：

例子 87. 主持人撤销一个与会者的发言权

```
<iq from='cronel@shakespeare.lit/desktop'
  id='voice2'
  to='darkcave@chat.shakespeare.lit'
  type='set' >
  <query xmlns='http://jabber.org/protocol/muc#admin' >
    <item nick='thirdwitch'
      role='visitor' />
  </query>
</iq>
```

<reason/> 元素是可选的 OPTIONAL：

例子 88. 主持人撤销一个与会者的发言权(包含一个原因 Reason)

```
<iq from='cronel@shakespeare.lit/desktop'
  id='voice2'
```

```

    to='darkcave@chat.shakespeare.lit'
    type='set'>
<query xmlns='http://jabber.org/protocol/muc#admin'>
  <item nick='thirdwitch'
        role='visitor'>
    <reason>Not so worthy after all!</reason>
  </item>
</query>
</iq>

```

服务必须 MUST 接着通知主持人成功了：

例子 89. 服务通知主持人成功了

```

<iq from='darkcave@chat.shakespeare.lit'
    id='voice2'
    to='cronel@shakespeare.lit/desktop'
    type='result' />

```

服务必须 MUST 接着以这个人的<room@service/nick>发送更新的出席信息给所有房客，在这个出席信息里包含了一个满足

'<http://jabber.org/protocol/muc#user>' 名字空间的<x/>元素，<x/>元素则包含一个<item/>子元素，其'role'属性值为"visitor"，指明移除了发言权。

例子 90. 服务通知失去发言权

```

<presence
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='cronel@shakespeare.lit/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='member'
          jid='hag66@shakespeare.lit/pda'
          role='visitor' />
  </x>
</presence>

[ ... ]

```

一个主持人不能 MUST NOT 从一个岗位等于或高于主持人岗位的用户撤销发言权。另外，服务不能 MUST NOT 允许一个管理员或所有者的发言权被任何人撤销。如果一个主持人尝试撤销这些人的发言权，服务必须 MUST 拒绝这个请求并返回一个 <not-allowed/> 的错误给发送者(通过以下的违规条目)：

例子 91. 服务对于尝试从管理员，所有者或更高岗位的用户撤销权限返回错误

```

<iq from='darkcave@chat.shakespeare.lit'
  id='voicetest'
  to='cronel@shakespeare.lit/desktop'
  type='error'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item nick='secondwitch' role='visitor' />
  </query>
  <error type='cancel'>
    <not-allowed xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>

```

修改发言权列表

在一个被主持的房间里主持人可能希望管理发言权列表。为了达到这个目的，主持人首先查询房间所有角色为 'participant' 的房客列表来请求发言权列表。

例子 92. 主持人请求发言权列表

```

<iq from='bard@shakespeare.lit/globe'
  id='voice3'
  to='goodfolk@chat.shakespeare.lit'
  type='get'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item role='participant' />
  </query>
</iq>

```

服务必须 MUST 接着返回发言权列表给主持人；每个条目必须 MUST 包含 'nick' 和 'role' 属性并且应该 SHOULD 包含 'affiliation' 和 'jid' 属性：

例子 93. 服务发送发言权列表给主持人

```

<iq from='goodfolk@chat.shakespeare.lit'
  id='voice3'
  to='bard@shakespeare.lit/globe'
  type='result'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='none'
      jid='polonius@hamlet/castle'
      nick='Polo'
      role='participant' />
    <item affiliation='none'

```



```

        jid='horatio@hamlet/castle'
        nick='horotoro'
        role='participant' />
    <item affiliation='member'
        jid='hecate@shakespeare.lit/broom'
        nick='Hecate'
        role='participant' />
</query>
</iq>

```

主持人可以 MAY 接着修改发言权列表。为了达到这个目的，主持人必须 MUST 发送变更了的条目（即，只有“delta”）给服务；每个条目必须 MUST 包含 'nick' 属性和 'role' 属性（通常设置值为“participant”或“visitor”）但是不应该 SHOULD NOT 包含 'jid' 属性并且不能 MUST NOT 包含 'affiliation' 属性（它用于管理如所有者那样的岗位而不是与会者那样的角色）：

例子 94. 主持人发送修改的发言权列表给服务

```

<iq from='bard@shakespeare.lit/globe'
    id='voice4'
    to='goodfolk@chat.shakespeare.lit'
    type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item nick='Hecate'
        role='visitor' />
    <item nick='rosencrantz'
        role='participant'>
      <reason>A worthy fellow.</reason>
    </item>
    <item nick='guildenstern'
        role='participant'>
      <reason>A worthy fellow.</reason>
    </item>
  </query>
</iq>

```

服务必须 MUST 接着通知主持人成功了：

例子 95. 服务通知主持人成功了

```

<iq from='goodfolk@chat.shakespeare.lit'
    id='voice1'
    to='bard@shakespeare.lit/globe'
    type='result' />

```

服务必须 MUST 接着为任何受影响的人发送更新的出席信息给所有房客，如前文的用例所述，发送适当的扩展出席信息来指明发言权的变更。

大家知道，不能撤销一个房间所有者或管理员的发言权，也不能撤销比发出请求的主持人岗位高的用户的发言权。如果一个房间管理员尝试通过修改发言权列表来撤销这类用户的发言权，服务必须 MUST 拒绝请求并返回一个 `<not-allowed/>` 错误给发送者：

例子 96. 服务返回错误给试图撤销管理员，所有者或比发送者岗位更高的用户的发言权的发送者

```
<iq from='goodfolk@chat.shakespeare.lit'
  id='voicetest'
  to='bard@shakespeare.lit/globe'
  type='error'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item jid='hecate@shakespeare.lit'
      nick='Hecate'
      role='visitor' />
  </query>
  <error type='cancel'>
    <not-allowed xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>
```

批准发言权申请

在本文的[申请发言权](#)章节提到，当服务接受到一个来自房客请求，它应该 SHOULD 转发那个请求给房间的主持人(们)。为了达到这个目的，服务应该 SHOULD 发送一个 `<message/>` 节给房间主持人(们)，这里 `<message/>` 节包含一个数据表格 data form 来批准或拒绝这个申请，如下所示。

’ 例子 97. 申请批准发言权表格

```
<message from='darkcave@chat.shakespeare.lit'
  id='approve'
  to='cronel@shakespeare.lit/pda'>
  <x xmlns='jabber:x:data' type='form'>
    <title>Voice request</title>
    <instructions>
      To approve this request for voice, select
      the &quot;Grant voice to this person?&quot;;
      checkbox and click OK. To skip this request,
      click the cancel button.
```

```

</instructions>
<field var='FORM_TYPE' type='hidden'>
  <value>http://jabber.org/protocol/muc#request</value>
</field>
<field var='muc#role'
  type='text-single'
  label='Requested role'>
  <value>participant</value>
</field>
<field var='muc#jid'
  type='text-single'
  label='User ID'>
  <value>hag66@shakespeare.lit/pda</value>
</field>
<field var='muc#roomnick'
  type='text-single'
  label='Room Nickname'>
  <value>thirdwitch</value>
</field>
<field var='muc#request_allow'
  type='boolean'
  label='Grant voice to this person?'>
  <value>>false</value>
</field>
</x>
</message>

```

为了批准这个申请，主持人将提交此表格：

例子 98. 批准发言权申请

```

<message from='cronel@shakespeare.lit/pda'
  id='approve'
  to='darkcave@chat.shakespeare.lit'>
  <x xmlns='jabber:x:data' type='submit'>
    <field var='FORM_TYPE' type='hidden'>
      <value>http://jabber.org/protocol/muc#request</value>
    </field>
    <field var='muc#role'>
      <value>participant</value>
    </field>
    <field var='muc#jid'>
      <value>hag66@shakespeare.lit/pda</value>
    </field>
    <field var='muc#roomnick'>

```

```
<value>thirdwitch</value>
</field>
<field var='muc#request_allow'>
  <value>true</value>
</field>
</x>
</message>
```

如果主持人批准了这个发言权申请，服务将授予发言权给该房客并发送一个出席信息更新，如本文[授予游客发言权](#)章节所述。

管理员用例

一个房间管理员有权修改用户岗位的持久信息（例如，通过禁止用户）并授予和撤销主持人权限，但是无权修改房间的定义，那是唯一属于房间所有者(们)的权力。具体哪些动作是管理员可以执行的则取决于配置。无论如何，在 MUC 框架中的用途，规定房间管理员最少拥有执行以下操作的权限：

1. 在房间里禁止一个用户
2. 在房间里修改黑名单
3. 授予或撤销成员资格
4. 修改成员列表
5. 授予或撤销主持人权力
6. 修改主持人列表

这些特性将由一个 请求/应答 request/response 式的交换来实现，使用 <iq/> 元素，包含满足 '<http://jabber.org/protocol/muc#admin>' 名字空间的子元素。以下例子展示协议如何与实现互动以得到期望的功能。（以下除非显示地声明，如果发送方的 'from' 地址中的 <user@host> 和任何房间管理员的纯 JID 都不同，接下来的任何管理请求必须 MUST 被拒绝；这种情况下，服务必须 MUST 返回一个 <forbidden/> 错误。）

禁止用户

在房间里一个管理员或所有者可以禁止一个或多个用户。这动作必须 MUST 基于房客的纯 JID 来执行。为了禁止一个用户，管理员必须 MUST 把该用户的岗位改为“outcast”。

例子 99. 管理员禁止用户

```
<iq from='kinghenryv@shakespeare.lit/throne'
  id='ban1'
  to='southampton@henryv.shakespeare.lit'
```

```

    type='set' >
    <query xmlns='http://jabber.org/protocol/muc#admin' >
      <item affiliation='outcast'
        jid='earlofcambridge@shakespeare.lit' />
    </query>
  </iq>

```

<reason/> 元素是可选的 OPTIONAL.

例子 100. 管理员禁止用户(包含一个原因 Reason)

```

<iq from='kinghenryv@shakespeare.lit/throne'
  id='ban1'
  to='southampton@henryv.shakespeare.lit'
  type='set' >
  <query xmlns='http://jabber.org/protocol/muc#admin' >
    <item affiliation='outcast'
      jid='earlofcambridge@shakespeare.lit' >
      <reason>Treason</reason>
    </item>
  </query>
</iq>

```

服务必须 MUST 把那个纯 JID 添加到黑名单, 应该 SHOULD 把被排斥者的昵称从已注册的昵称列表中移除, 并且必须 MUST 通知管理员或所有者成功了:

例子 101. 服务通知管理员或所有者成功了

```

<iq from='southampton@henryv.shakespeare.lit'
  id='ban1'
  to='kinghenryv@shakespeare.lit/throne'
  type='result' />

```

服务必须 MUST 也移除任何还在房间中的被禁止的用户, 通过发送 "unavailable" 类型的出席信息节给每个被禁止的房客, 在扩展的出席信息中包含一个状态码 301, 可选地带上 reason (如果服务提供的话) 以及执行这个禁止动作的用户的纯 JID.

例子 102. 服务移除被禁止的用户

```

<presence
  from='southampton@henryv.shakespeare.lit/cambridge'
  to='earlofcambridge@shakespeare.lit/stabber'
  type='unavailable' >
  <x xmlns='http://jabber.org/protocol/muc#user' >

```

```

    <item affiliation='outcast' role='none'>
      <actor jid='kinghenryv@shakespeare.lit' />
      <reason>Treason</reason>
    </item>
    <status code='301' />
  </x>
</presence>

```

包含状态码可使客户端能够提交他们自己的通知消息（例如，适当的用户位置的信息）。可选的包含原因 `reason` 元素以及执行者 `actor` 使得被踢的用户能理解为什么他或她被踢了，以及被踢的用户可以找谁去理论。

通知主持人之后，服务必须 **MUST** 接着通知剩余的房客那个被禁止的房客已经不在房间里了，即从被禁止用户发送“unavailable”类型的出席信息节给所有剩余的房客（就像房客自愿退出房间时所做的一样），包含状态码 `status` 以及可选的原因 `reason` 和执行者 `actor`。

例子 103. 服务通知剩余的房客

```

<presence
  type='unavailable'
  from='southampton@henryv.shakespeare.lit/cambridge'
  to='exeter@shakespeare.lit/pda'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='outcast'
      jid='earlofcambridge@shakespeare.lit/stabber'
      role='none' />
    <status code='301' />
  </x>
</presence>

[ ... ]

```

就像[踢出房客](#)一样，一个用户不能被自己岗位低的管理员禁止。所以，如果一个管理员尝试禁止一个所有者，服务必须 **MUST** 拒绝这个请求并返回一个 `<not-allowed/>` 错误给发送者：

例子 104. 服务对尝试禁止更高岗位用户返回错误

```

<iq from='kinghenryv@shakespeare.lit/throne'
  id='ban1'
  to='southampton@henryv.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='outcast'

```

```

        jid='earlofcambridge@shakespeare.lit'>
      <reason>Treason</reason>
    </item>
  </query>
  <error type='cancel'>
    <not-allowed xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>

```

如果一个管理员或所有者尝试禁止他自己，服务必须 MUST 拒绝这个请求并返回一个 <conflict/> 错误给发送者。（注意：这和踢出自己时推荐的服务行为不同，踢自己的行为服务是允许的。）

修改黑名单

房间管理员可能希望修改黑名单。注意：黑名单总是基于用户的纯 JID。要修改黑名单，管理员首先向房间查询所有岗位为 'outcast' 的用户以得到黑名单。

例子 105. 管理员请求黑名单

```

<iq from='kinghenryv@shakespeare.lit/throne'
  id='ban2'
  to='southampton@henryv.shakespeare.lit'
  type='get'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='outcast' />
  </query>
</iq>

```

服务必须 MUST 接着返回黑名单给管理员；每个条目必须 MUST 包含 'affiliation' 和 'jid' 属性但不应该 SHOULD NOT 包含 'nick' 和 'role' 属性：

例子 106. 服务发送黑名单给管理员

```

<iq from='southampton@henryv.shakespeare.lit'
  id='ban2'
  to='kinghenryv@shakespeare.lit/throne'
  type='result'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='outcast'
      jid='earlofcambridge@shakespeare.lit'>
      <reason>Treason</reason>
    </item>
  </query>
</iq>

```

```
</query>
</iq>
```

管理员可以 MAY 接着修改黑名单。为此，管理员必须 MUST 发送变更的条目（即，仅是“delta”）给服务；每个条目必须 MUST 包含‘affiliation’属性（通常设为“outcast”来禁止或“none”来取消禁止）和‘jid’属性，但不应该 SHOULD NOT 包含‘nick’属性，不能 MUST NOT 包含‘role’属性（它用来管理角色，例如与会者，而不是被排斥者岗位）；另外，reason 和 actor 元素是可选的 OPTIONAL：

例子 107. 管理员发送修改的黑名单给服务

```
<iq from='kinghenryv@shakespeare.lit/throne'
  id='ban3'
  to='southampton@henryv.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='outcast'
      jid='earlofcambridge@shakespeare.lit'>
      <reason>Treason</reason>
    </item>
    <item affiliation='outcast'>
      jid='lordscroop@shakespeare.lit'>
      <reason>Treason</reason>
    </item>
    <item affiliation='outcast'
      jid='sirthomasgrey@shakespeare.lit'>
      <reason>Treason</reason>
    </item>
  </query>
</iq>
```

更新黑名单之后，服务必须 MUST 通知管理员成功了：

例子 108. 服务通知管理员成功了

```
<iq from='southampton@henryv.shakespeare.lit'
  id='ban3'
  to='kinghenryv@shakespeare.lit/throne'
  type='result' />
```

服务必须 MUST 接着移除受影响的房客（如果他们在房间里）并从他们发送更新的出席信息（包含适当的状态码）给所有剩余的房客，如“禁止用户”用例所述。（服务应该 SHOULD 也移除从保留房间昵称列表中移除每个被禁止的用户的保留昵称，如果必要。）

当一个实体被一个房间禁止，实现应该 SHOULD 按以下顺序匹配 JIDs（这些匹配规则和 [RFC 3921](#) 中定义的隐私列表的匹配规则是相同的）：

1. <user@domain/resource>（仅匹配特定的资源）
2. <user@domain>（匹配任何资源）
3. <domain/resource>（仅匹配特定资源）
4. <domain>（匹配域名本身，就像任何 user@domain 或 domain/resource 一样）

一些管理员可能希望在一个 MUC 服务中的所有房间里禁止所有和特定域名相关的用户。这个功能是一个服务级的特性，所以超过了本文的范围，它定义在 XEP-0133 里。

授予成员资格

管理员可以授予成员资格给一个用户；方法是把用户的岗位改为 “member”（通常如果用户在房间里，基于昵称，如果用户不在房间里，则基于纯 JID；在这两种情况下如果提供了昵称，那么这个昵称就是用户在这个房间的缺省昵称，如果实现支持那个功能的话）：

例子 109. 管理员授予成员资格

```
<iq from='cronel@shakespeare.lit/desktop'
  id='member1'
  to='darkcave@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='member'
      jid='hag66@shakespeare.lit' />
  </query>
</iq>
```

<reason/> 元素是可选的 OPTIONAL.

例子 110. 管理员授予成员资格(包含一个原因 Reason)

```
<iq from='cronel@shakespeare.lit/desktop'
  id='member1'
  to='darkcave@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='member'
      jid='hag66@shakespeare.lit'>
      <reason>A worthy witch indeed!</reason>
    </item>
  </query>
</iq>
```

```
    </item>
  </query>
</iq>
```

服务必须 MUST 把这个用户添加到成员列表，然后通知管理员成功了：

例子 111. 服务通知管理员成功了

```
<iq from='darkcave@chat.shakespeare.lit'
  id='member1'
  to='cronel@shakespeare.lit/desktop'
  type='result' />
```

如果该用户在房间里，服务必须 MUST 接着以这个用户的名义发送更新的出席信息给所有房客，在这个出席信息里包含了一个满足'<http://jabber.org/protocol/muc#user>' 名字空间的<x/>元素，<x/>元素则包含一个<item/>子元素，其'affiliation' 属性值为"member"，指明授予了成员资格。

例子 112. 服务发送成员资格通知给所有房客

```
<presence
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='cronel@shakespeare.lit/desktop' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
    <item affiliation='member'
      jid='hag66@shakespeare.lit/pda'
      role='participant' />
  </x>
</presence>

[ ... ]
```

如果该用户不在房间里，服务可以 MAY 从房间本身发送一个消息给房间的房客们，在这个消息里包含了一个满足'<http://jabber.org/protocol/muc#user>' 名字空间的<x/>元素，<x/>元素则包含一个<item/>子元素，其'affiliation' 属性值为"member"，指明授予了成员资格。

例子 113. 服务发送成员资格通知给所有房客

```
<message
  from='chat.shakespeare.lit'
  to='cronel@shakespeare.lit/desktop' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
    <item affiliation='member'
```

```

        jid='hag66@shakespeare.lit'
        role='none' />
    </x>
</message>

[ ... ]

```

撤销成员资格

一个管理员可能想撤销一个用户的成员资格；通过把该用户的岗位改为“none”：

例子 114. 管理员撤销成员资格

```

<iq from='cronel@shakespeare.lit/desktop'
    id='member2'
    to='darkcave@chat.shakespeare.lit'
    type='set' >
  <query xmlns='http://jabber.org/protocol/muc#admin' >
    <item affiliation='none'
        jid='hag66@shakespeare.lit' />
  </query>
</iq>

```

<reason/> 元素是可选的 OPTIONAL.

例子 115. 管理员撤销成员资格(包含一个原因 Reason)

```

<iq from='cronel@shakespeare.lit/desktop'
    id='member2'
    to='darkcave@chat.shakespeare.lit'
    type='set' >
  <query xmlns='http://jabber.org/protocol/muc#admin' >
    <item affiliation='none'
        jid='hag66@shakespeare.lit' >
      <reason>Not so worthy after all!</reason>
    </item>
  </query>
</iq>

```

服务必须 MUST 从成员列表中移除该用户然后通知主持人成功了：

例子 116. 服务通知主持人成功了

```

<iq from='darkcave@chat.shakespeare.lit'

```

```
id='member2'
to='cronel@shakespeare.lit/desktop'
type='result' />
```

服务必须 MUST 接着以这个用户的名义发送更新的出席信息节给所有房客，在这个出席信息里包含了一个满足'<http://jabber.org/protocol/muc#user>' 名字空间的<x/>元素，<x/>元素则包含一个<item/>子元素，其'affiliation' 属性值为"none"，指明失去了成员资格。

例子 117. 服务通知失去成员资格

```
<presence
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='cronel@shakespeare.lit/desktop' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
    <item affiliation='none'
      jid='hag66@shakespeare.lit/pda'
      role='participant' />
  </x>
</presence>

[ ... ]
```

如果房间是仅限会员的，服务必须 MUST 从房间移除这个用户，包含一个状态码 321 来指明用户被移除是因为岗位变更，并通知所有剩余的房客：

例子 118. 服务移除非会员

```
<presence
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='cronel@shakespeare.lit/desktop' >
  type='unavailable' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
    <item affiliation='none' role='none' >
      <actor jid='bard@shakespeare.lit' />
    </item>
    <status code='321' />
  </x>
</presence>

<presence
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='cronel@shakespeare.lit/desktop' >
  type='unavailable' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
```

```

    <item affiliation='none' role='none' />
    <status code='321' />
  </x>
</presence>

[ ... ]

```

修改成员列表

在一个仅限会员的房间的上下文里，成员列表本质上是一个允许人们加入房间的白名单 “whitelist”。任何不是成员的人等于是被禁止加入该房间，即使他们的岗位不是 “outcast”。

在一个开放房间的上下文里，成员列表只是一个注册了这个房间的用户（纯 JID 和保留的昵称）的列表。这些用户可以出现在一个房间名册里，有他们自己的保留房间昵称，在搜索结果或类似 FAQ 里被返回给（查询者）。

推荐 RECOMMENDED 在仅限会员的房间里只让房间管理员拥有修改成员列表的权力。为此，管理员首先请求成员列表，通过查询房间里所有岗位为 “member” 的用户来实现：

例子 119. 管理员请求成员列表

```

<iq from='cronel@shakespeare.lit/desktop'
  id='member3'
  to='darkcave@chat.shakespeare.lit'
  type='get'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='member' />
  </query>
</iq>

```

注意：在一个仅限会员的房间里，服务也应该 SHOULD 返回成员列表给任何房客；即，当一个房间的成员请求房间列表时，它不应该 SHOULD NOT 生成一个 <forbidden/> 错误。这个功能可帮助客户端展示所有现有的成员，即使他们中的一些人不在房间里，例如，帮助成员确定是否另一个用户应该被邀请。服务也应该 SHOULD 允许任何成员接收成员列表，即使还不是一个房客（译注：即未进入房间）。

服务必须 MUST 接着返回全部的成员列表给管理员，遵循 ['http://jabber.org/protocol/muc#admin'](http://jabber.org/protocol/muc#admin) 名字空间；每个条目必须 MUST 包含 'affiliation' 和 'jid' 属性，每个正在房间里的成员可以 MAY 包含 'nick' 和 'role' 属性。

例子 120. 服务发送成员列表给管理员

```
<iq from='darkcave@chat.shakespeare.lit'
  id='member3'
  to='cronel@shakespeare.lit/desktop'
  type='result'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='member'
      jid='hag66@shakespeare.lit'
      nick='thirdwitch'
      role='participant' />
  </query>
</iq>
```

管理员可以 MAY 接着修改成员列表。为此，管理员必须 MUST 发送变更的条目（即，仅“delta”）给服务；每个条目必须 MUST 包含‘affiliation’属性（通常值设为“member”或“none”）和‘jid’属性，但不应该 SHOULD NOT 包含‘nick’属性，并且不能 MUST NOT 包含‘role’属性（它是用来管理角色的，例如与会者，而不是成员的岗位）：

例子 121. 管理员发送修改的成员列表给服务

```
<iq from='cronel@shakespeare.lit/desktop'
  id='member4'
  to='darkcave@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='none'
      jid='hag66@shakespeare.lit' />
    <item affiliation='member'
      jid='hecate@shakespeare.lit' />
  </query>
</iq>
```

服务必须 MUST 修改成员列表然后通知主持人成功了：

例子 122. 服务通知主持人成功了

```
<iq from='darkcave@chat.shakespeare.lit'
  id='member4'
  to='cronel@shakespeare.lit/desktop'
  type='result' />
```

服务必须 MUST 变更任何受影响的用户的岗位。如果该用户已经从成员列表中移除了，服务必须 MUST 把该用户的岗位从 “member” 变更为 “none”。如果该用户已经被加入到成员列表，服务必须 MUST 把该用户的岗位改成 “member”。

如果一个被移除的成员正在一个仅限会员的房间，服务应该 SHOULD 踢出这个房客，如前文所述，通过把被移除的角色的角色改成 “none” 并发送适当的出席信息给这个被移除的成员来实现。无论是否被移除的那个用户在或不在一个仅限会员的房间里，服务必须 MUST 随后拒绝这个用户的进入。

对所有的房间类型来说，服务必须 MUST 以这个用户的名义发送更新的出席信息给所有的房客，这个出席信息拥有一个满足 ['http://jabber.org/protocol/muc#user'](http://jabber.org/protocol/muc#user) 名字空间的 <x/> 元素并包含一个 <item/> 子元素，这个子元素的 'affiliation' 属性值设为 “none”，以指明这个岗位的变更。

例子 123. 服务发送失去成员资格的通知给所有房客

```
<presence
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='cronel@shakespeare.lit/desktop' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
    <item affiliation='none'
      jid='hag66@shakespeare.lit/pda'
      role='participant' />
  </x>
</presence>

[ ... ]
```

另外，服务必须 SHOULD 发送一个邀请给任何已加入到仅限会员的房间里的成员名单中的用户，如果该用户目前在房间还没有岗位，例如作为一个管理员或所有者（这类用户在定义时不在房间里；同时要注意这个例子里使用了一个密码 password 而不是原因 reason -- 这两个子元素都是可选的 OPTIONAL）：

例子 124. 房间发送邀请给新成员

```
<message
  from='darkcave@chat.shakespeare.lit'
  to='hecate@shakespeare.lit' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
    <invite from='bard@shakespeare.lit' />
    <password>cauldronburn</password>
  </x>
</message>
```

因为只有管理员们和所有者们应该 SHOULD 被允许修改成员列表，一个实现可以 MAY 提供一个配置选项，在仅限会员的房间里开放邀请权限给任何成员。这种情况下，任何被发送的邀请都应该 SHOULD 自动触发被邀请者加入成员列表。无论如何，如果邀请权限被限于管理员们，而普通成员尝试发送邀请，服务必须 MUST 拒绝这个邀请的请求并返回一个 <forbidden/> 错误给发送者：

例子 125. 服务在普通成员尝试邀请其他人加入仅限会员的房间时返回错误

```
<message
  from='darkcave@chat.shakespeare.lit'
  to='hag66@shakespeare.lit/pda'
  type='error' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
    <invite to='hecate@shakespeare.lit' >
      <reason>
        Hey Hecate, this is the place for all good witches!
      </reason>
    </invite>
  </x>
  <error type='auth' >
    <forbidden xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</message>
```

从开放房间发送的邀请不能 MUST NOT 触发被邀请者加入成员列表。

如果一个用户被加入一个开放房间的成员列表并且该用户在该房间内，服务必须 MUST 以该用户的名义发送更新的出席信息给所有房客，这个出席信息拥有一个满足 '<http://jabber.org/protocol/muc#user>' 名字空间的<x/> 元素并包含一个<item/>子元素，这个子元素的'affiliation' 属性值设为"member"，以指明这个岗位的变更。

例子 126. 服务发送成员资格通知给所有房客

```
<presence
  from='darkcave@chat.shakespeare.lit/hecate'
  to='cronel@shakespeare.lit/desktop' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
    <item affiliation='member'
      jid='hecate@shakespeare.lit/broom'
      role='participant' />
  </x>
</presence>

[ ... ]
```


授予主持人权限

管理员可能想授予主持人权限给一个与会者或游客；通过把用户的角色改为“moderator”：

例子 127. 管理员授予主持人权限

```
<iq from='cronel@shakespeare.lit/desktop'
  id='mod1'
  to='darkcave@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item nick='thirdwitch'
      role='moderator' />
  </query>
</iq>
```

<reason/> 元素是可选的 OPTIONAL.

例子 128. 管理员授予主持人权限(包含一个原因 Reason)

```
<iq from='cronel@shakespeare.lit/desktop'
  id='mod1'
  to='darkcave@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item nick='thirdwitch'
      role='moderator'>
      <reason>A worthy witch indeed!</reason>
    </item>
  </query>
</iq>
```

服务必须 MUST 添加这个用户到主持人列表然后通知管理员成功了：

例子 129. 服务通知管理员成功了

```
<iq from='darkcave@chat.shakespeare.lit'
  id='mod1'
  to='cronel@shakespeare.lit/desktop'
  type='result' />
```

服务必须 MUST 接着以该用户的名义发送更新的出席信息给所有的房客，这个出席信息拥有一个满足 '<http://jabber.org/protocol/muc#user>' 名字空间的

<x/> 元素并包含一个<item/>子元素, 这个子元素的'role' 属性值设为 "moderator", 以指明添加了主持人权限.

例子 130. 服务发送主持人权限通知给所有房客

```
<presence
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='cronel@shakespeare.lit/desktop' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
    <item affiliation='member'
      jid='hag66@shakespeare.lit/pda'
      role='moderator' />
  </x>
</presence>

[ ... ]
```

撤销主持人权限

管理员可能想撤销用户的主持人权限. 一个管理员只可以 MAY 撤销岗位为 "member" 或 "none" (也就是, 非管理员和所有者) 的用户的主持人权限. 权限的撤销是通过把用户的角色改为 "participant"实现的:

例子 131. 管理员撤销主持人权限

```
<iq from='cronel@shakespeare.lit/desktop'
  id='mod2'
  to='darkcave@chat.shakespeare.lit'
  type='set' >
  <query xmlns='http://jabber.org/protocol/muc#admin' >
    <item nick='thirdwitch'
      role='participant' />
  </query>
</iq>
```

<reason/> 元素是可选的 OPTIONAL.

例子 132. 管理员撤销主持人权限(包含一个原因 Reason)

```
<iq from='cronel@shakespeare.lit/desktop'
  id='mod2'
  to='darkcave@chat.shakespeare.lit'
  type='set' >
  <query xmlns='http://jabber.org/protocol/muc#admin' >
```

```

    <item nick='thirdwitch'
          role='participant'>
      <reason>Not so worthy after all!</reason>
    </item>
  </query>
</iq>

```

服务必须 MUST 从主持人列表移除这个用户然后通知管理员成功了：

例子 133. 服务通知管理员成功了

```

<iq from='darkcave@chat.shakespeare.lit'
    id='mod2'
    to='cronel@shakespeare.lit/desktop'
    type='result' />

```

服务必须 MUST 接着以该用户的名义发送更新的出席信息给所有的房客，这个出席信息拥有一个满足 '<http://jabber.org/protocol/muc#user>' 名字空间的 <x/> 元素并包含一个<item/>子元素，这个子元素的'role'属性值设为"participant"，以指明移除了主持人权限。

例子 134. 服务通知失去了主持人权限

```

<presence
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='cronel@shakespeare.lit/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='member'
          jid='hag66@shakespeare.lit/pda'
          role='participant' />
  </x>
</presence>

```

[...]

大家知道，管理员不 MUST NOT 被允许从岗位为 "owner" 或 "admin" 的用户撤销主持人权限。如果一个管理员尝试撤销这类用户的权限，服务必须 MUST 拒绝这个请求并返回一个 <not-allowed/> 错误给发送者：

例子 135. 服务在用户尝试撤销管理员或所有者的主持人权限时返回错误

```

<iq from='darkcave@chat.shakespeare.lit'
    id='modtest'
    to='cronel@shakespeare.lit/desktop'
    type='error'>

```

```

<query xmlns='http://jabber.org/protocol/muc#admin'>
  <item nick='secondwitch' role='participant' />
</query>
<error type='cancel'>
  <not-allowed xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
</error>
</iq>

```

修改主持人列表

管理员可能希望修改主持人列表。为此，管理员首先通过请求房间内所有角色为‘moderator’的用户来请求主持人列表。

例子 136. 管理员请求主持人列表

```

<iq from='cronel@shakespeare.lit/desktop'
  id='mod3'
  to='darkcave@chat.shakespeare.lit'
  type='get'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item role='moderator' />
  </query>
</iq>

```

服务必须 MUST 接着返回主持人列表给管理员；每个条目必须 MUST 包含 ‘jid’，‘nick’，‘role’ 属性并应该 SHOULD 包含 ‘affiliation’ 属性：

例子 137. 服务发送主持人列表给管理员

```

<iq from='darkcave@chat.shakespeare.lit'
  id='mod3'
  to='cronel@shakespeare.lit/desktop'
  type='result'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='member'
      jid='hag66@shakespeare.lit/pda'
      nick='thirdwitch'
      role='moderator' />
  </query>
</iq>

```

管理员可以 MAY 接着修改主持人列表。为此，管理员必须 MUST 发送修改的条目(即，仅“delta”)给服务；每个条目必须 MUST 包含 ‘jid’ 属性和‘role’ 属性(通常值设为 “member” 或 “participant”) 但不应该 SHOULD NOT 包含

'nick' 属性并且不能 MUST NOT 包含 'affiliation' 属性(它被用于管理类似管理员这样的岗位而不是主持人这样的角色):

例子 138. 管理员发送修改了的主持人列表给服务

```
<iq from='cronel@shakespeare.lit/desktop'
  id='mod4'
  to='darkcave@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item jid='hag66@shakespeare.lit/pda'
      role='participant' />
    <item jid='hecate@shakespeare.lit/broom'
      role='moderator' />
  </query>
</iq>
```

服务必须 MUST 修改主持人列表并通知管理员成功了:

例子 139. 服务通知管理员成功了

```
<iq from='darkcave@chat.shakespeare.lit'
  id='mod4'
  to='cronel@shakespeare.lit/desktop'
  type='result' />
```

服务必须 MUST 随后以所有受影响的用户发送更新的出席信息给所有的房客, 支出主持人权限的变更, 通过发送前面用例所述的适当的扩展出席信息.

显然, 房间所有者或房间管理员的主持人权限不能被撤销. 如果一个房间管理员尝试通过修改主持人列表来撤销这类用户的主持人权限, 服务必须 MUST 拒绝请求并返回一个 <not-allowed/> 错误给发送者:

例子 140. 服务在用户尝试撤销管理员或所有者的主持人权限时返回错误

```
<iq from='darkcave@chat.shakespeare.lit'
  id='modtest'
  to='hag66@shakespeare.lit/pda'
  type='error'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item jid='hecate@shakespeare.lit/broom'
      nick='Hecate'
      role='participant' />
  </query>
  <error type='cancel'>
```

```
<not-allowed xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
</error>
</iq>
```

批准注册申请

如果一个服务不自动接受注册到房间的请求，它可以 MAY 为房间管理员提供一个方法来批准或拒绝来自 Jabber 的注册请求(替代方案是，它可能提供一个 web 接口或一些其他管理工具)。对这个服务来说，最简单的办法就是，当接收到注册请求时发送一个 <message/> 节给房间管理员(们)，这里的<message/> 节包含一个数据表格 Data Form asking 用来询问管理员是否批准用户的注册申请。推荐 RECOMMENDED 以下数据表格 Data Form，但是实现可以 MAY 使用完全不同的表格，或 or 在下面的表格基础上补充字段。

例子 141. 注册申请批准表格

```
<message from='darkcave@chat.shakespeare.lit'
  id='approve'
  to='cronel@shakespeare.lit/pda'>
  <x xmlns='jabber:x:data' type='form'>
    <title>Registration request</title>
    <instructions>
      To approve this registration request, select the
      &quot;Allow this person to register with the room?&quot;;
      checkbox and click OK. To skip this request, click the
      cancel button.
    </instructions>
    <field var='FORM_TYPE' type='hidden'>
      <value>http://jabber.org/protocol/muc#register</value>
    </field>
    <field var='muc#register_first'
      type='text-single'
      label='Given Name'>
      <value>Brunhilde</value>
    </field>
    <field var='muc#register_last'
      type='text-single'
      label='Family Name'>
      <value>Entwhistle-Throckmorton</value>
    </field>
    <field var='muc#register_roomnick'
      type='text-single'
      label='Desired Nickname'>
      <value>thirdwitch</value>
```

```

</field>
<field var='muc#register_url'
        type="text-single"
        label="User URL">
    <value>http://witchesonline/~hag66/</value>
</field>
<field var='muc#register_email'
        type="text-single"
        label="Email Address">
    <value>hag66@witchesonline</value>
</field>
<field var='muc#register_faentry'
        type="text-single"
        label="FAQ Entry">
    <value>Just another witch.</value>
</field>
<field var='muc#register_allow'
        type='boolean'
        label='Allow this person to register with the room?'>
    <value>0</value>
</field>
</x>
</message>

```

如果管理员批准了注册申请，服务将把该用户注册到房间。

更多高级的批准机制(例如，使用特定命令 [Ad-Hoc Commands 19](#) 来接收注册申请列表, 就像 [Publish-Subscribe 20](#) 里所做的一样) 超出了本文的范围。

所有者用例

每个房间必须 MUST 至少有一个所有者，而所有者(或一个成功者)在一个房间的生命周期里是这个房间长期存在的属性(例如，所有者在退出一个持久性的房间时不会失去所有权)。本文假定(初始的)房间所有者是那个新建了该房间的用户并且有一个房间所有者有权修改房间配置选项的定义，例如房间类型。理想情况下，房间所有者不仅能指定房间类型(密码保护的，仅限会员的，等等)而且包括如本文的 [XEP-0045#需求|需求] 章节所述的房间特定属性。另外，如果所有者能指定其他所有者们的 JID 也是不错的，但那将取决于具体实现。

为了让配置选项更加广泛提供必要的伸缩性，房间配置将使用 Data Forms (XEP-0004)，通过使用由 <http://jabber.org/protocol/muc> 名字空间触发。也就是，如果一个实体在它请求加入房间的 join/request 里不包含 MUC 名字空间，那么服务将立刻新建房间，在新建房间之前不等待通过数据表格进行配置(这保证了和旧的“groupchat 1.0”协议的向后兼容)；无论如何，如果房间的

join/create 请求包含了 MUC 扩展，那么服务在新建和解锁该房间之前将通过数据表格请求配置。

注意：以下展示的配置选项列出了本文的需求章节的所有特性和房间类型；无论如何，实际的配置选项和表格布局将取决于实现和具体的布署。而且，这些只是例子，不代表这些就是房间可以拥有的所有允许或需要的配置选项。一个特定的实现或布署可以 MAY 选择提供额外的配置选项(敏感词过滤，设置房间的缺省语言，消息记录，等等)，这就是为什么在这里使用 'jabber:x:data' 协议是很有价值的。

新建房间

一般注意事项

新建房间的权限可以 MAY 限制在特定的用户群或可以 MAY 保留给一个服务级别的管理员。如果访问被拒绝而一个用户试图新建一个房间，服务必须 MUST 返回一个 <not-allowed/> 错误：

例子 142. 服务通知用户不能新建房间

```
<presence
  from='darkcave@chat.shakespeare.lit/thirdwitch'
  to='hag66@shakespeare.lit/pda'
  type='error'>
  <x xmlns='http://jabber.org/protocol/muc' />
  <error type='cancel'>
    <not-allowed xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</presence>
```

如果访问不限制，服务必须 MUST 允许用户按以下步骤新建房间。

从房间创建的视角来看，本质上有两种房间：

- 即时房间“Instant rooms” -- 可以立刻访问并基于某些缺省配置自动创建。
- 保留房间“Reserved rooms” -- 在任何人访问之前由房间配置者手动创建。

新建和配置这些房间的流程如下：

1. 用户必须 MUST 发送出席信息到 <room@service/nick> 并声明他或她对 MUC 协议的支持，通过包含一个扩展的出席信息，并包含在一个空的满足

- '<http://jabber.org/protocol/muc>' 名字空间的 <x/> 子元素里(注意这里不包含 '#owner' 或 '#user' 后缀).
2. 如果用户被允许新建房间并且房间还不存在, 服务必须 MUST 根据一些缺省配置新建此房间, 指定请求的用户成为初始的房间拥有者, 并增加这个拥有者到该房间但不允许任何别的用户进入该房间(有效地锁定"locking"该房间). 从房间发送由所有者收到的初始的出席信息节必须 MUST 包含扩展的出席信息以指出该用户的状态为一个所有者并承认房间已经被创建了(通过状态码 201) 并等待配置.
 3. 如果初始的房间所有者想新建和配置一个保留房间, 房间所有者必须 MUST 接着请求一个配置, 通过发送类型为"get"的 IQ 节并包含一个空的满足'<http://jabber.org/protocol/muc#owner>' 名字空间的<query/>元素给该房间, 然后完成第4和第5步. 如果房间所有者喜欢新建一个即时房间, 该房间所有者必须 MUST 发送一个遵循'<http://jabber.org/protocol/muc#owner>' 名字空间的 query 元素并包含一个遵循 'jabber:x:data' 名字空间的空的类型为 "submit" 的 <x/> 元素, 然后跳到第6步.
 4. 如果房间所有者请求了一个配置表格, 服务必须 MUST 发送一个包含配置表格并遵循 'jabber:x:data' 名字空间的 IQ 给房间拥有者. 如果没有配置选项可用, 房间必须 MUST 返回一个空的 query 元素给房间所有者.
 5. 初始的房间所有者应该 SHOULD 为该房间提供一个开始的配置(或接受缺省配置), 通过发送"set"类型并包含完整的配置表格的 IQ. 另外, 房间所有者可以 MAY 取消配置过程. (实现可以 MAY 设置一个初始化配置的超时, 这样如果房间所有者再给定的超时时间内不配置房间, 房间所有者就被假定已经接受了缺省得配置或取消了配置过程.)
 6. 一旦服务从初始房间所有者接收了完整的配置表格(或接收到了一个即时房间的请求), 服务必须 MUST 解锁 "unlock" 这个房间 (即, 允许其他用户进入此房间) 并发送"result"类型的 IQ 给房间所有者. 如果服务接收到了取消 (指令), 它必须 MUST 销毁这个房间.

以下例子展示了这个协议流程.

首先, Jabber 用户必须 MUST 发送出席信息给房间, 包含空的 <x/> 元素, 遵循 '<http://jabber.org/protocol/muc>' 名字空间(当他寻求进入一个房间时也发送和这同样的节).

例子 143. Jabber 用户新建一个房间并声明对多用户聊天的支持

```
<presence
  from='cronel@shakespeare.lit/desktop'
  to='darkcave@chat.shakespeare.lit/firstwitch' >
  <x xmlns='http://jabber.org/protocol/muc' />
</presence>
```

如果该房间不存在，服务应该 SHOULD 新建这个房间(取决于关于新建房间的本地策略)，指定发出请求的用户的纯 JID 成为所有者，添加这个所有者到房间，并通过发送以下格式的出席信息节承认房间新建成功：

例子 144. 服务承认房间新建成功

```
<presence
  from='darkcave@chat.shakespeare.lit/firstwitch'
  to='cronel@shakespeare.lit/desktop' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
    <item affiliation='owner'
      role='moderator' />
    <status code='110' />
    <status code='201' />
  </x>
</presence>
```

接收到该房间已经创建的通知之后，房间所有者需要决定是否接受缺省的房间配置(即，新建一个即时房间“instant room”)还是做一些不同于缺省房间设置的配置(即，新建一个保留房间“reserved room”)。完成这两个用例的协议流程展示如下。

注意：如果如上的发送到一个不存在的房间里的出席信息节没有包含一个遵循 <http://jabber.org/protocol/muc> 名字空间的 `<x/>` 元素，服务应该 SHOULD 立刻新建一个缺省的房间(即，它必须 MUST 假定客户端支持“groupchat 1.0”而不是 Multi-User Chat，所以在等待房间创建者决定是创建即时房间还是保留房间的时候，它不能 MUST NOT 锁定这个房间)。

新建即时房间

如果初始的房间所有者想接受缺省的房间配置(即，新建一个即时房间“instant room”)，房间所有者必须 MUST 拒绝初始配置表格，通过发送一个 IQ set 给 `<room@service>` 本身，包含一个遵循 <http://jabber.org/protocol/muc#owner> 名字空间的 `<query/>` 元素，这里 `<query/>` 的唯一子元素是一个空的遵循 `jabber:x:data` 名字空间的 `<x/>` 元素并且拥有一个 `'type'` 属性值为“submit”：

例子 145. 所有者请求即时房间

```
<iq from='cronel@shakespeare.lit/desktop'
  id='createl'
  to='darkcave@chat.shakespeare.lit'
  type='set' >
  <query xmlns='http://jabber.org/protocol/muc#owner' >
    <x xmlns='jabber:x:data' type='submit' />
```

```
</query>
</iq>
```

服务必须 MUST 接着解锁该房间并允许其他实体加入它.

新建保留房间

如果初始的房间所有者想新建并配置一个保留房间, 这个房间所有者必须 MUST 请求初始配置表格, 通过发送一个 IQ get 给 <room@service> 本身, 包含一个空的遵循 '<http://jabber.org/protocol/muc#owner>' 名字空间的 <query/> 元素 :

例子 146. 所有者请求配置表单

```
<iq from='cronel@shakespeare.lit/desktop'
  id='createl'
  to='darkcave@chat.shakespeare.lit'
  type='get'>
  <query xmlns='http://jabber.org/protocol/muc#owner' />
</iq>
```

如果房间还不存在, 服务必须 MUST 返回一个初始的房间配置表单给该用户. (注意: 以下例子展示一个配置选项的典型例子. 已登记用于房间创建和配置的所有 x:data 字段列表由 XMPP Registrar 维护; 参见本文的 [XMPP 注册项事项](#) 章节.)

例子 147. 服务发送配置表单

```
<iq from='darkcave@chat.shakespeare.lit'
  id='createl'
  to='cronel@shakespeare.lit/desktop'
  type='result'>
  <query xmlns='http://jabber.org/protocol/muc#owner'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Configuration for "darkcave" Room</title>
      <instructions>
        Your room darkcave@macbeth has been created!
        The default configuration is as follows:
        - No logging
        - No moderation
        - Up to 20 occupants
        - No password required
        - No invitation required
        - Room is not persistent
        - Only admins may change the subject
```

- Presence broadcasted for all users

To accept the default configuration, click OK. To select a different configuration, please complete this form.

</instructions>

<field

type='hidden'

var='FORM_TYPE'>

<value>http://jabber.org/protocol/muc#roomconfig</value>

</field>

<field

label='Natural-Language Room Name'

type='text-single'

var='muc#roomconfig_roomname' />

<field

label='Short Description of Room'

type='text-single'

var='muc#roomconfig_roomdesc' />

<field

label='Natural Language for Room Discussions'

type='text-single'

var='muc#roomconfig_lang' />

<field

label='Enable Public Logging?'

type='boolean'

var='muc#roomconfig_enablelogging'>

<value>0</value>

</field>

<field

label='Allow Occupants to Change Subject?'

type='boolean'

var='muc#roomconfig_changesubject'>

<value>0</value>

</field>

<field

label='Allow Occupants to Invite Others?'

type='boolean'

var='muc#roomconfig_allowinvites'>

<value>0</value>

</field>

<field

label='Maximum Number of Occupants'

type='list-single'

var='muc#roomconfig_maxusers'>

```

    <value>20</value>
    <option label=' 10' ><value>10</value></option>
    <option label=' 20' ><value>20</value></option>
    <option label=' 30' ><value>30</value></option>
    <option label=' 50' ><value>50</value></option>
    <option label=' 100' ><value>100</value></option>
    <option label=' None' ><value>none</value></option>
</field>
<field
    label='Roles for which Presence is Broadcast'
    type=' list-multi'
    var=' muc#roomconfig_presencebroadcast' >
    <value>moderator</value>
    <value>participant</value>
    <value>visitor</value>
    <option label=' Moderator' ><value>moderator</value></option>
    <option
label=' Participant' ><value>participant</value></option>
    <option label=' Visitor' ><value>visitor</value></option>
</field>
<field
    label='Roles and Affiliations that May Retrieve Member List'
    type=' list-multi'
    var=' muc#roomconfig_getmemberlist' >
    <value>moderator</value>
    <value>participant</value>
    <value>visitor</value>
    <option label=' Moderator' ><value>moderator</value></option>
    <option
label=' Participant' ><value>participant</value></option>
    <option label=' Visitor' ><value>visitor</value></option>
</field>
<field
    label=' Make Room Publicly Searchable?'
    type=' boolean'
    var=' muc#roomconfig_publicroom' >
    <value>1</value>
</field>
<field
    label=' Make Room Persistent?'
    type=' boolean'
    var=' muc#roomconfig_persistentroom' >
    <value>0</value>
</field>

```

```

<field
  label='Make Room Moderated?'
  type='boolean'
  var='muc#roomconfig_moderatedroom' >
  <value>0</value>
</field>
<field
  label='Make Room Members-Only?'
  type='boolean'
  var='muc#roomconfig_memberonly' >
  <value>0</value>
</field>
<field
  label='Password Required to Enter?'
  type='boolean'
  var='muc#roomconfig_passwordprotectedroom' >
  <value>0</value>
</field>
<field type='fixed' >
  <value>
    If a password is required to enter this room,
    you must specify the password below.
  </value>
</field>
<field
  label='Password'
  type='text-private'
  var='muc#roomconfig_roomsecret' />
<field
  label='Who May Discover Real JIDs?'
  type='list-single'
  var='muc#roomconfig_whois' >
  <option label='Moderators Only' >
    <value>moderators</value>
  </option>
  <option label='Anyone' >
    <value>anyone</value>
  </option>
</field>
<field type='fixed' >
  <value>
    You may specify additional people who have
    administrative privileges in the room. Please
    provide one Jabber ID per line.
  </value>

```

```

        </value>
    </field>
    <field
        label='Room Admins'
        type='jid-multi'
        var='muc#roomconfig_roomadmins' />
    <field type='fixed'>
        <value>
            You may specify additional owners for this
            room. Please provide one Jabber ID per line.
        </value>
    </field>
    <field
        label='Room Owners'
        type='jid-multi'
        var='muc#roomconfig_roomowners' />
</x>
</query>
</iq>

```

注意：_whois 配置选项指定该房间是非匿名的(值为 "anyone")，半匿名的(值为 "moderators")，还是全匿名的(值为 "none"，不显示在这)。

如果没有配置选项可用，服务必须 MUST 返回空的 query 元素给房间所有者：

例子 148. 服务通知所有者没有配置可用

```

<iq from='darkcave@chat.shakespeare.lit'
    id='create1'
    to='cronel@shakespeare.lit/desktop'
    type='result'>
    <query xmlns='http://jabber.org/protocol/muc#owner' />
</iq>

```

房间所有者应该 SHOULD 接着填好表单并提交给服务。

例子 149. 所有者提交配置表单

```

<iq from='cronel@shakespeare.lit/desktop'
    id='create2'
    to='darkcave@chat.shakespeare.lit'
    type='set'>
    <query xmlns='http://jabber.org/protocol/muc#owner'>
        <x xmlns='jabber:x:data' type='submit'>
            <field var='FORM_TYPE'>

```

```
<value>http://jabber.org/protocol/muc#roomconfig</value>
</field>
<field var='muc#roomconfig_roomname'>
  <value>A Dark Cave</value>
</field>
<field var='muc#roomconfig_roomdesc'>
  <value>The place for all good witches!</value>
</field>
<field var='muc#roomconfig_enablelogging'>
  <value>0</value>
</field>
<field var='muc#roomconfig_changesubject'>
  <value>1</value>
</field>
<field var='muc#roomconfig_allowinvites'>
  <value>0</value>
</field>
<field var='muc#roomconfig_maxusers'>
  <value>10</value>
</field>
<field var='muc#roomconfig_publicroom'>
  <value>0</value>
</field>
<field var='muc#roomconfig_persistentroom'>
  <value>0</value>
</field>
<field var='muc#roomconfig_moderatedroom'>
  <value>0</value>
</field>
<field var='muc#roomconfig_memberonly'>
  <value>0</value>
</field>
<field var='muc#roomconfig_passwordprotectedroom'>
  <value>1</value>
</field>
<field var='muc#roomconfig_roomsecret'>
  <value>cauldronburn</value>
</field>
<field var='muc#roomconfig_whois'>
  <value>moderators</value>
</field>
<field var='muc#roomconfig_roomadmins'>
  <value>wiccarocks@shakespeare.lit</value>
  <value>hecate@shakespeare.lit</value>
```



```
    </field>
  </x>
</query>
</iq>
```

如果房间创建成功，服务必须 MUST 通知新的房间所有者成功了：

例子 150. 服务通知新房间所有者成功

```
<iq from='darkcave@chat.shakespeare.lit'
  id='create2'
  to='cronel@shakespeare.lit/desktop'
  type='result' />
```

如果因为指定的房间配置违反了一个或多个服务策略而导致房间创建失败（例如，因为密码保护房间的密码为空），服务必须 MUST 返回一个 `<not-acceptable/>` 错误。

例子 151. 服务通知所有者请求的配置选项不被接受

```
<iq from='darkcave@chat.shakespeare.lit'
  id='create2'
  to='cronel@shakespeare.lit/desktop'
  type='error' >
  <error type='modify' >
    <not-acceptable xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>
```

另一方面，房间所有者可以 MAY 取消配置过程：

例子 152. 所有者取消初始的配置

```
<iq from='cronel@shakespeare.lit/desktop'
  id='create2'
  to='darkcave@chat.shakespeare.lit'
  type='set' >
  <query xmlns='http://jabber.org/protocol/muc#owner' >
    <x xmlns='jabber:x:data' type='cancel' />
  </query>
</iq>
```

如果房间所有者取消了初始的房间配置，服务应该 SHOULD 销毁房间，确保发送不可用出席信息给房间所有者（详见“销毁房间”用例）。

如果房间所有者在提交表单之前因为任何原因下线了(例如, 失去连接), 服务将接收到一个类型为“unavailable”的出席信息节, 从所有者到所有者的 <room@service/nick> 或到 <room@service> (或两者). 服务必须 MUST 接着销毁这个房间, 发送一个“unavailable”类型的出席信息节, 从房间到所有者, 包含一个 <destroy/> 元素以及 reason (子元素)(如果提供了), 参见本文的[销毁房间](#)章节.

申请唯一房间名

在一些场合(例如, 当[把一对一聊天转为会议](#)), 房间创建者可能想在尝试新建房间之前请求一个唯一的房间名(例如, 避免可能的房间冲突). 为此, 服务可以 MAY 如本章所述支持这个特性. (如果服务支持这个特性, 它必须 MUST 在对服务发现信息请求应答时返回一个[“http://jabber.org/protocol/muc#unique”](#)特性.)

房间创建者通过发送一个 IQ-get 给服务本身来请求唯一房间名, 这个 IQ 节中包含一个空的 <unique/> 元素, 遵循[“http://jabber.org/protocol/muc#unique”](#) 名字空间:

例子 153. 实体请求唯一房间名

```
<iq from='cronel@shakespeare.lit/desktop'
  id='unique1'
  to='chat.shakespeare.lit'
  type='get'>
  <unique xmlns='http://jabber.org/protocol/muc#unique' />
</iq>
```

如果服务支持这个特性, 它应该 SHOULD 以 XML 字符数据的方式返回一个唯一房间名, 包含一个 <unique/> 元素 (但不创建该房间):

例子 154. 服务返回唯一房间名

```
<iq from='chat.shakespeare.lit'
  id='unique1'
  to='cronel@shakespeare.lit/desktop'
  type='result'>
  <unique xmlns='http://jabber.org/protocol/muc#unique'>
    6d9423a55f499b29ad20bf7b2bdea4f4b885ead1
  </unique>
</iq>
```

服务可以 MAY 拒绝返回一个唯一房间名给一个没有资格创建房间的实体, 或那些发送请求唯一房间名过多次数的实体, 等等.

服务可以 MAY 使用算法保证房间名的创建在服务上下文中是唯一的（例如，对发出请求的 JID, datetime, 和 random salt 的 SHA-1 哈希运算）。

房间创建者将接着使用 XML 字符数据 <unique/> 元素作为它请求的房间 JID 的节点标识符(ID)：

例子 155. 所有者以唯一名创建房间

```
<presence
  from='cronel@shakespeare.lit/desktop'

to='6d9423a55f499b29ad20bf7b2bdea4f4b885ead1@chat.shakespeare.lit/fir
stwitch'>
  <x xmlns='http://jabber.org/protocol/muc' />
</presence>
```

随后的房间配置

在指定房间的初始配置之后的任何时间，房间所有者可能想修改房间配置。为此，房间所有者必须 MUST 向房间发出一个新的配置表单请求，通过发送一个 IQ 到 <room@service>，包含一个空的 <query/> 元素，遵循 ['http://jabber.org/protocol/muc#owner'](http://jabber.org/protocol/muc#owner) 名字空间。

例子 156. 所有者请求配置表单

```
<iq from='cronel@shakespeare.lit/desktop'
  id='config1'
  to='darkcave@chat.shakespeare.lit'
  type='get'>
  <query xmlns='http://jabber.org/protocol/muc#owner' />
</iq>
```

如果 'from' 地址的 <user@host> 部分和房间所有者的纯 JID 不符，服务必须 MUST 返回一个 <forbidden/> 错误给发送者：

例子 157. 服务禁止非所有者的访问配置

```
<iq from='darkcave@chat.shakespeare.lit'
  id='configures'
  to='wiccarocks@shakespeare.lit/laptop'
  type='error'>
  <query xmlns='http://jabber.org/protocol/muc#owner' />
  <error type='auth'>
    <forbidden xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
```

```
</error>
</iq>
```

另外，服务必须 MUST 以当前使用的选项组作为缺省值发送一个配置表单给房间所有者：

例子 158. 服务发送配置表单给所有者

```
<iq from='darkcave@chat.shakespeare.lit'
  id='config1'
  to='cronel@shakespeare.lit/desktop'
  type='result'>
  <query xmlns='http://jabber.org/protocol/muc#owner'>
    <x xmlns='jabber:x:data' type='form'>
      <title>Configuration for "darkcave" Room</title>
      <instructions>
        Complete this form to make changes to
        the configuration of your room.
      </instructions>
      <field
        type='hidden'
        var='FORM_TYPE'>
        <value>http://jabber.org/protocol/muc#roomconfig</value>
      </field>
      <field
        label='Natural-Language Room Name'
        type='text-single'
        var='muc#roomconfig_roomname'>
        <value>A Dark Cave</value>
      </field>
      <field
        label='Short Description of Room'
        type='text-single'
        var='muc#roomconfig_roomdesc'>
        <value>The place for all good witches!</value>
      </field>
      <field
        label='Enable Public Logging?'
        type='boolean'
        var='muc#roomconfig_enablelogging'>
        <value>0</value>
      </field>
      <field
        label='Allow Occupants to Change Subject?'
        type='boolean'
```

```

        var='muc#roomconfig_changesubject' >
        <value>0</value>
    </field>
    <field
        label='Allow Occupants to Invite Others?'
        type='boolean'
        var='muc#roomconfig_allowinvites' >
        <value>0</value>
    </field>
    <field
        label='Maximum Number of Occupants'
        type='list-single'
        var='muc#roomconfig_maxusers' >
        <value>10</value>
        <option label='10' ><value>10</value></option>
        <option label='20' ><value>20</value></option>
        <option label='30' ><value>30</value></option>
        <option label='50' ><value>50</value></option>
        <option label='100' ><value>100</value></option>
        <option label='None' ><value>none</value></option>
    </field>
    <field
        label='Roles for which Presence is Broadcast'
        type='list-multi'
        var='muc#roomconfig_presencebroadcast' >
        <value>moderator</value>
        <value>participant</value>
        <value>visitor</value>
        <option label='Moderator' ><value>moderator</value></option>
        <option
label='Participant' ><value>participant</value></option>
        <option label='Visitor' ><value>visitor</value></option>
    </field>
    <field
        label='Roles and Affiliations that May Retrieve Member List'
        type='list-multi'
        var='muc#roomconfig_getmemberlist' >
        <value>moderator</value>
        <value>participant</value>
        <value>visitor</value>
        <option label='Moderator' ><value>moderator</value></option>
        <option
label='Participant' ><value>participant</value></option>
        <option label='Visitor' ><value>visitor</value></option>

```

```
</field>
<field
  label='Make Room Publicly Searchable?'
  type='boolean'
  var='muc#roomconfig_publicroom' >
  <value>0</value>
</field>
<field
  label='Make Room Persistent?'
  type='boolean'
  var='muc#roomconfig_persistentroom' >
  <value>0</value>
</field>
<field
  label='Make Room Moderated?'
  type='boolean'
  var='muc#roomconfig_moderatedroom' >
  <value>0</value>
</field>
<field
  label='Make Room Members Only?'
  type='boolean'
  var='muc#roomconfig_memberonly' >
  <value>0</value>
</field>
<field
  label='Password Required for Entry?'
  type='boolean'
  var='muc#roomconfig_passwordprotectedroom' >
  <value>1</value>
</field>
<field type='fixed' >
  <value>
    If a password is required to enter this room,
    you must specify the password below.
  </value>
</field>
<field
  label='Password'
  type='text-private'
  var='muc#roomconfig_roomsecret' >
  <value>cauldronburn</value>
</field>
<field
```

```

        label='Who May Discover Real JIDs?'
        type='list-single'
        var='muc#roomconfig_whois' >
<value>moderators</value>
<option label='Moderators Only' >
    <value>moderators</value>
</option>
<option label='Anyone' >
    <value>anyone</value>
</option>
</field>
<field type='fixed' >
    <value>
        You may specify additional people who have
        administrative privileges in the room. Please
        provide one Jabber ID per line.
    </value>
</field>
<field
    label='Room Admins'
    type='jid-multi'
    var='muc#roomconfig_roomadmins' >
    <value>wiccarocks@shakespeare.lit</value>
    <value>hecate@shakespeare.lit</value>
</field>
<field type='fixed' >
    <value>
        You may specify additional owners for this
        room. Please provide one Jabber ID per line.
    </value>
</field>
<field
    label='Room Owners'
    type='jid-multi'
    var='muc#roomconfig_roomowners' />
</x>
</query>
</iq>

```

如果没有配置选项可用，服务必须 MUST 返回一个空的 query 元素给房间所有者，如前面的用例所示。

该房间所有者应该 SHOULD 接着以更新的配置信息提交表单。

另外，房间所有者可以 MAY 取消这次配置过程：

例子 159. 所有者取消随后的配置

```
<iq from='cronel@shakespeare.lit/desktop'
  id='config2'
  to='darkcave@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#owner'>
    <x xmlns='jabber:x:data' type='cancel' />
  </query>
</iq>
```

如果房间所有者取消随后的配置，服务必须 MUST 让该房间的配置保持和房间所有者请求这次配置之前一样。

如果一次房间配置变更导致一个房间管理员失去管理权限，而这个管理员正在房间里，该房间必须 MUST 为那个管理员发送更新的出席信息给所有房客，表明状态的变更，通过包含一个遵循 '<http://jabber.org/protocol/muc#user>' 名字空间的 `<x/>` 元素以及一个 `<item/>` 子元素，该 `<item/>` 子元素的 'affiliation' 属性值为 "member" 或 'role' 属性值为 "participant" 或 "visitor"，以适当地表达岗位级别和房间类型：

例子 160. 服务通知失去管理员岗位

```
<presence
  from='darkcave@chat.shakespeare.lit/secondwitch'
  to='cronel@shakespeare.lit/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='member'
      jid='wiccarocks@shakespeare.lit/laptop'
      role='participant' />
  </x>
</presence>

[ ... ]
```

如果一次房间配置变更导致一个用户获得管理员权限，而这个用户正在房间里，房间必须 MUST 为那个用户发送更新的出席信息给所有房客，表明状态的变更，通过包含一个遵循 '<http://jabber.org/protocol/muc#user>' 名字空间的 `<x/>` 元素以及一个 `<item/>` 子元素，该 `<item/>` 子元素的 'affiliation' 属性值为 "admin" 或 'role' 属性值为 "admin"：

例子 161. 服务通知所有用户有人获得管理员岗位

```
<presence
  from='darkcave@chat.shakespeare.lit/secondwitch'
```



```

    to='cronel@shakespeare.lit/desktop'>
<x xmlns='http://jabber.org/protocol/muc#user'>
  <item affiliation='admin'
        jid='wiccarocks@shakespeare.lit/laptop'
        role='moderator' />
</x>
</presence>

[ ... ]

```

如果一次房间配置变更导致一个房间所有者失去所有者权限，而这个所有者正在房间里，该房间必须 MUST 为那个所有者发送更新的出席信息给所有房客，表明状态的变更，通过包含一个遵循 '<http://jabber.org/protocol/muc#user>' 名字空间的 `<x/>` 元素以及一个 `<item/>` 子元素，该 `<item/>` 子元素的 'affiliation' 属性值为 "admin" 或 'role' 属性值为根据岗位和房间类型确定的适当的值(推荐为"moderator")。

例子 162. 服务通知失去所有者岗位

```

<presence
  from='darkcave@chat.shakespeare.lit/secondwitch'
  to='cronel@shakespeare.lit/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='admin'
          jid='wiccarocks@shakespeare.lit/laptop'
          role='moderator' />
  </x>
</presence>

[ ... ]

```

如果没有其他所有者，服务不能 MUST NOT 允许一个所有者撤销他或她自己的所有权；如果一个所有者企图这么干，服务必须 MUST 返回一个 `<conflict/>` 错误给这个所有者。然而，如果没有其他所有者，服务应该 SHOULD 允许一个所有者撤销自己的所有者权限。

如果一次房间配置变更导致一个用户获得所有者权限，而这个用户正在房间里，房间必须 MUST 为那个用户发送更新的出席信息给所有房客，表明状态的变更，通过包含一个遵循 '<http://jabber.org/protocol/muc#user>' 名字空间的 `<x/>` 元素以及一个 `<item/>` 子元素，该 `<item/>` 子元素的 'affiliation' 属性值为 "owner" 或 'role' 属性值为根据岗位和房间类型确定的适当的值(推荐为"moderator")。

例子 163. 服务通知所有用户有人获得所有者权限

```

<presence
  from='darkcave@chat.shakespeare.lit/secondwitch'
  to='cronel@shakespeare.lit/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='owner'
      jid='wiccarocks@shakespeare.lit/laptop'
      role='moderator' />
  </x>
</presence>

[ ... ]

```

如果一次房间配置变更导致房间类型变成仅限会员，但还有非成员在房间里，服务必须 MUST 从房间移除任何非成员，并在发送给那些剩余的房客的 '不可用' 出席信息节里包含状态码 322.

配置变更通知

当一个房间的配置变更会对房间的隐私和安全策略产生影响时，该房间必须 MUST 发送通知给所有房客。这个通知将包括一个 <message/> 节，包含一个遵循 <http://jabber.org/protocol/muc#user> 名字空间的 <x/> 元素，<x/> 元素则只有一个 <status/> 子元素，其 'code' 属性为一个适当的值。这是例子：

例子 164. 配置状态码

```

<message from='darkcave@chat.shakespeare.lit'
  to='cronel@shakespeare.lit/desktop'
  type='groupchat'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <status code='170' />
  </x>
</message>

```

房间配置中和隐私相关的策略变更导致生成这些状态码，如下：

- 如果房间日志功能可用了，状态码 170.
- 如果房间日志现在禁止了，状态码 171.
- 如果房间现在是非匿名的了，状态码 172.
- 如果房间现在是半匿名的了，状态码 173.
- 如果房间现在是全匿名的了，状态码 174.

对更多其他配置变更，房间应该 SHOULD 发送状态码 104 这样感兴趣的房客如果想要的话可以接受到更新的房间配置.

授予所有者权限

如果实现允许，一个所有者可以 MAY 授予所有权给其他用户；只要把用户的岗位改成“owner”：

例子 165. 所有者授予所有权

```
<iq from='cronel@shakespeare.lit/desktop'
  id='owner1'
  to='darkcave@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='owner'
      jid='hecate@shakespeare.lit' />
  </query>
</iq>
```

<reason/> 元素是可选的 OPTIONAL.

例子 166. 所有者授予所有权(包含一个原因 Reason)

```
<iq from='cronel@shakespeare.lit/desktop'
  id='owner1'
  to='darkcave@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='owner'
      jid='hecate@shakespeare.lit'>
      <reason>A worthy witch indeed!</reason>
    </item>
  </query>
</iq>
```

服务必须 MUST 把用户添加到所有者列表并通知所有者成功了：

例子 167. 服务通知所有者成功了

```
<iq from='darkcave@chat.shakespeare.lit'
  id='owner1'
  to='cronel@shakespeare.lit/desktop'
  type='result' />
```

如果该用户在房间里，服务必须 MUST 接着以这个用户的名义发送更新的出席信息给所有房客，表明已授予所有权，通过包含一个遵循

'<http://jabber.org/protocol/muc#user>' 名字空间的 `<x/>` 元素以及一个 `<item/>` 子元素, 该 `<item/>` 子元素的 'affiliation' 属性值为 "owner" 并且 'role' 属性值为根据岗位和房间类型确定的适当的值 (推荐为 "moderator").

例子 168. 服务发送所有权通知给所有房客

```
<presence
  from='darkcave@chat.shakespeare.lit/hecate'
  to='cronel@shakespeare.lit/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='owner'
      jid='hecate@shakespeare.lit'
      role='moderator' />
  </x>
</presence>

[ ... ]
```

如果该用户不在房间, 服务可以 MAY 从房间本身发送一个消息给房间的房客们, 表明已授予所有权, 通过包含一个遵循

'<http://jabber.org/protocol/muc#user>' 名字空间的 `<x/>` 元素以及一个 `<item/>` 子元素, 该 `<item/>` 子元素的 'affiliation' 属性值为 "owner".

例子 169. 服务发送所有权通知给所有房客

```
<message
  from='chat.shakespeare.lit'
  to='cronel@shakespeare.lit/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='member'
      jid='hecate@shakespeare.lit'
      role='none' />
  </x>
</message>

[ ... ]
```

撤销所有者权限

实现可以 MAY 允许一个所有者撤销其他用户的所有权; 只要把用户的岗位改成非 "owner":

例子 170. 所有者撤销所有权

```
<iq from='cronel@shakespeare.lit/desktop'
  id='owner2'
  to='darkcave@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='admin'
      jid='hecate@shakespeare.lit' />
  </query>
</iq>
```

<reason/>元素是可选的 OPTIONAL.

例子 171. 所有者撤销所有权(包含一个原因)

```
<iq from='cronel@shakespeare.lit/desktop'
  id='owner2'
  to='darkcave@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='admin'
      jid='hecate@shakespeare.lit'>
      <reason>Not so worthy after all!</reason>
    </item>
  </query>
</iq>
```

如果没有其它所有者, 服务不能 MUST NOT 允许一个所有者撤销他或她自己的所有权; 如果一个所有者尝试这么干, 服务必须 MUST 返回一个 <conflict/> 错误给该所有者. 然而, 如果有其他所有者, 服务应该 SHOULD 允许一个所有者撤销自己的所有权.

如果一个实现不允许所有者撤销另一个用户的所有权, 实现必须 MUST 返回一个 <not-authorized/> 错误给做出这个请求的所有者.

注意: 允许一个所有者移除其它用户的所有权能给房间管理一个折衷的控制模式; 所以这个特性是可选的 OPTIONAL, 并且鼓励实现支持通过一个只有拥有服务范围管理权限的用户使用的接口来移除所有者.

其它情况下, 服务必须 MUST 把用户从所有者列表移除并通知所有者成功了:

例子 172. 服务通知所有者成功了

```
<iq from='darkcave@chat.shakespeare.lit'
  id='owner2'
  to='cronel@shakespeare.lit/desktop'>
```

```
type='result' />
```

如果该用户在房间里，服务必须 MUST 接着以这个用户的名义发送更新的出席信息给所有房客，表明已失去所有权，通过包含一个遵循 ['http://jabber.org/protocol/muc#user'](http://jabber.org/protocol/muc#user) 名字空间的 `<x/>` 元素以及一个 `<item/>` 子元素，该 `<item/>` 子元素的 `'affiliation'` 属性值为非 `"owner"` 并且 `'role'` 属性值为根据岗位和房间类型确定的适当的值：

例子 173. 服务通知失去所有权

```
<presence
  from='darkcave@chat.shakespeare.lit/secondwitch'
  to='cronel@shakespeare.lit/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='admin'
      jid='hecate@shakespeare.lit'
      role='moderator' />
  </x>
</presence>

[ ... ]
```

如果该用户不在房间，服务可以 MAY 从房间本身发送一个消息给房间的房客们，表明已失去所有权，通过包含一个遵循 ['http://jabber.org/protocol/muc#user'](http://jabber.org/protocol/muc#user) 名字空间的 `<x/>` 元素以及一个 `<item/>` 子元素，该 `<item/>` 子元素的 `'affiliation'` 属性值为非 `"owner"`。

例子 174. 服务发送失去所有权通知给所有房客

```
<message
  from='chat.shakespeare.lit'
  to='cronel@shakespeare.lit/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='admin'
      jid='hecate@shakespeare.lit'
      role='none' />
  </x>
</message>

[ ... ]
```

修改所有者列表

如果实现允许，一个房间所有者可能想修改所有者列表。为此，所有者首先请求所有者列表，通过向房间请求所有岗位为 'owner' 的用户。

例子 175. 所有者请求所有者列表

```
<iq from='bard@shakespeare.lit/globe'
  id='owner3'
  to='darkcave@chat.shakespeare.lit'
  type='get'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='owner' />
  </query>
</iq>
```

如果 'from' 地址的 <user@host> 部分和房间所有者的纯 JID 不符，服务必须 MUST 返回一个 <forbidden/> 错误给发送者。

否则，服务必须 MUST 接着返回所有者列表给所有者；每个条目必须 MUST 包含 'affiliation' 和 'jid' 属性，对任何正是一名房客（也就是正在房间里）的所有者可以 MAY 包含 'nick' 和 'role' 属性：

例子 176. 服务发送所有者列表给所有者

```
<iq from='darkcave@chat.shakespeare.lit'
  id='owner3'
  to='bard@shakespeare.lit/globe'
  type='result'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='owner'
      jid='cronel@shakespeare.lit' />
  </query>
</iq>
```

所有者可以 MAY 接着修改所有者列表。为此，所有者必须 MUST 发送修改的条目（即，仅 "delta"）给服务；[21](#) 每个条目必须 MUST 包含 'affiliation' 和 'jid' 属性，但不应该 SHOULD NOT 包含 'nick' 属性，不能 MUST NOT 包含 'role' 属性（它用于管理参与者之类的角色，而不是所有者之类的岗位）：

例子 177. 所有者发送修改过的所有者列表给服务

```
<iq from='bard@shakespeare.lit/globe'
  id='owner4'
  to='darkcave@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
```

```

    <item affiliation='owner'
        jid='hecate@shakespeare.lit' />
</query>
</iq>

```

只有所有者被允许修改所有者列表。如果一个非所有者试图察看或修改所有者列表，服务必须 MUST 拒绝这个请求并返回一个 <forbidden/> 错误给发送者：

例子 178. 服务对于非所有者试图修改所有者列表返回错误

```

<iq from='darkcave@chat.shakespeare.lit'
    id='ownertest'
    to='hag66@shakespeare.lit/pda'
    type='error' >
  <query xmlns='http://jabber.org/protocol/muc#admin' >
    <item affiliation='owner'
        jid='hecate@shakespeare.lit' />
  </query>
  <error type='auth' >
    <forbidden xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>

```

如果没有别的所有者，服务不能 MUST NOT 允许一个所有者撤销自己的所有权；如果一个所有者尝试这么干，服务必须 MUST 返回一个 <conflict/> 错误给该所有者。然而，如果有其他所有者，服务应该 SHOULD 允许一个所有者撤销自己的所有权。

其它情况下，服务必须 MUST 修改所有者列表并通知所有者成功了：

例子 179. 服务通知所有者成功了

```

<iq from='darkcave@chat.shakespeare.lit'
    id='owner4'
    to='cronel@shakespeare.lit/desktop'
    type='result' />

```

服务必须 MUST 也为任何因前述所有者列表修改而导致的岗位变化而发送出席信息通知。

授予管理员权限

一个所有者可以授予管理员权限给一个成员或无岗位的用户；只要把用户的岗位改成“admin”：

例子 180. 所有者授予管理员权限

```
<iq from='cronel@shakespeare.lit/desktop'
  id='admin1'
  to='darkcave@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='admin'
      jid='wiccarocks@shakespeare.lit' />
  </query>
</iq>
```

<reason/> 元素是可选的 OPTIONAL.

例子 181. 所有者授予管理员权限(包含一个原因 Reason)

```
<iq from='cronel@shakespeare.lit/desktop'
  id='admin1'
  to='darkcave@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='admin'
      jid='wiccarocks@shakespeare.lit'>
      <reason>A worthy witch indeed!</reason>
    </item>
  </query>
</iq>
```

服务必须 MUST 把用户添加到管理员列表并通知所有者成功了:

例子 182. 服务通知所有者成功了

```
<iq from='darkcave@chat.shakespeare.lit'
  id='admin1'
  to='cronel@shakespeare.lit/desktop'
  type='result' />
```

如果该用户在房间里, 服务必须 MUST 接着以这个用户的名义发送更新的出席信息给所有房客, 表明已授予管理员权限, 通过包含一个遵循 ['http://jabber.org/protocol/muc#user'](http://jabber.org/protocol/muc#user) 名字空间的 <x/> 元素以及一个 <item/> 子元素, 该 <item/> 子元素的 'affiliation' 属性值为 "admin" 并且 'role' 属性值为根据岗位和房间类型确定的适当的值.

例子 183. 服务发送管理员权限通知给所有房客

```

<presence
  from='darkcave@chat.shakespeare.lit/secondwitch'
  to='cronel@shakespeare.lit/desktop' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
    <item affiliation='admin'
      jid='wiccarocks@shakespeare.lit'
      role='moderator' />
  </x>
</presence>

[ ... ]

```

如果该用户不在房间，服务可以 MAY 从房间本身发送一个消息给房间的房客们，表明已授予管理员权限，通过包含一个遵循 ['http://jabber.org/protocol/muc#user'](http://jabber.org/protocol/muc#user) 名字空间的 `<x/>` 元素以及一个 `<item/>` 子元素，该 `<item/>` 子元素的 `'affiliation'` 属性值为 `"admin"`。

例子 184. 服务发送管理员权限通知给所有房客

```

<message
  from='chat.shakespeare.lit'
  to='cronel@shakespeare.lit/desktop' >
  <x xmlns='http://jabber.org/protocol/muc#user' >
    <item affiliation='admin'
      jid='wiccarocks@shakespeare.lit'
      role='none' />
  </x>
</message>

[ ... ]

```

撤销管理员权限

一个所有者可能想撤销一个用户的管理员权限；只要把用户的岗位改成非 `"admin"` 和非 `"owner"`：

例子 185. 所有者撤销管理员权限

```

<iq from='cronel@shakespeare.lit/desktop'
  id='admin2'
  to='darkcave@chat.shakespeare.lit'
  type='set' >
  <query xmlns='http://jabber.org/protocol/muc#admin' >
    <item affiliation='member'

```

```

        jid='wicarocks@shakespeare.lit' />
    </query>
</iq>

```

<reason/>元素是可选的 OPTIONAL.

例子 186. 所有者撤销管理员权限(包含一个原因)

```

<iq from='cronel@shakespeare.lit/desktop'
    id='admin2'
    to='darkcave@chat.shakespeare.lit'
    type='set'>
    <query xmlns='http://jabber.org/protocol/muc#admin'>
        <item affiliation='member'
            jid='wicarocks@shakespeare.lit'>
            <reason>Not so worthy after all!</reason>
        </item>
    </query>
</iq>

```

服务必须 MUST 把该用户从管理员列表移除并接着通知所有者成功了:

例子 187. 服务通知所有者成功了

```

<iq from='darkcave@chat.shakespeare.lit'
    id='admin2'
    to='cronel@shakespeare.lit/desktop'
    type='result' />

```

如果该用户在房间里, 服务必须 MUST 接着以这个用户的名义发送更新的出席信息给所有房客, 表明已失去管理员权限, 通过包含一个遵循 ['http://jabber.org/protocol/muc#user'](http://jabber.org/protocol/muc#user) 名字空间的 <x/> 元素以及一个 <item/> 子元素, 该 <item/> 子元素的 'affiliation' 属性值为非"admin"非"owner", 并且 'role' 属性值为根据岗位和房间类型确定的适当的值:

例子 188. 服务通知失去管理员权限

```

<presence
    from='darkcave@chat.shakespeare.lit/secondwitch'
    to='cronel@shakespeare.lit/desktop'>
    <x xmlns='http://jabber.org/protocol/muc#user'>
        <item affiliation='member'
            jid='wicarocks@shakespeare.lit'
            role='participant' />
    </x>

```

```
</presence>
```

```
[ ... ]
```

如果该用户不在房间，服务可以 MAY 从房间本身发送一个消息给房间的房客们，表明已失去所有权，通过包含一个遵循 ['http://jabber.org/protocol/muc#user'](http://jabber.org/protocol/muc#user) 名字空间的 `<x/>` 元素以及一个 `<item/>` 子元素，该 `<item/>` 子元素的 `'affiliation'` 属性值为非 `"admin"`。

例子 189. 服务通知失去管理员权限

```
<message
  from='chat.shakespeare.lit'
  to='cronel@shakespeare.lit/desktop'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='member'
      jid='wicarocks@shakespeare.lit'
      role='none' />
  </x>
</message>

[ ... ]
```

修改管理员列表

一个房间所有者可能想修改管理员列表。为此，所有者首先请求管理员列表，通过向房间请求所有岗位为 `'admin'` 的用户。

例子 190. 所有者请求管理员列表

```
<iq from='bard@shakespeare.lit/desktopaffiliation
  id='admin3'
  to='darkcave@chat.shakespeare.lit'
  type='get'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='admin' />
  </query>
</iq>
```

如果 `'from'` 地址的 `<user@host>` 部分和房间所有者的纯 JID 不符，服务必须 MUST 返回一个 `<forbidden/>` 错误给发送者。

否则，服务必须 MUST 接着返回管理员列表给所有者；每个条目必须 MUST 包含 'affiliation' 和 'jid' 属性，对任何正是一名房客（也就是正在房间里）的管理员可以 MAY 包含 'nick' 和 'role' 属性：

例子 191. 服务发送管理员列表给所有者

```
<iq from='darkcave@chat.shakespeare.lit'
  id='admin3'
  to='bard@shakespeare.lit/globe'
  type='result'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='admin'
      jid='wiccarocks@shakespeare.lit'
      nick='secondwitch' />
    <item affiliation='admin'
      jid='hag66@shakespeare.lit' />
  </query>
</iq>
```

所有者可以 MAY 接着修改管理员列表。为此，所有者必须 MUST 发送修改的条目（即，仅 "delta"）给服务；[22](#) 每个条目必须 MUST 包含 'affiliation' 属性（通常值为 "admin" 或 "none"）和 'jid' 属性，但不应该 SHOULD NOT 包含 'nick' 属性，不能 MUST NOT 包含 'role' 属性（它用于管理参与者之类的角色，而不是管理员之类的岗位）：

例子 192. 所有者发送修改的管理员列表给服务

```
<iq from='bard@shakespeare.lit/globe'
  id='admin4'
  to='darkcave@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='none'
      jid='hag66@shakespeare.lit'>
    </item>
    <item affiliation='admin'
      jid='hecate@shakespeare.lit'>
    </item>
  </query>
</iq>
```

只有所有者被允许修改管理员列表。如果一个非所有者试图察看或修改所有者列表，服务必须 MUST 拒绝这个请求并返回一个 <forbidden/> 错误给发送者：

例子 193. 服务对于非所有者试图修改管理员列表返回错误

```
<iq from='darkcave@chat.shakespeare.lit'
  id='admintest'
  to='hag66@shakespeare.lit/pda'
  type='error'>
  <query xmlns='http://jabber.org/protocol/muc#admin'>
    <item affiliation='admin'
      jid='hecate@shakespeare.lit' />
  </query>
  <error type='auth'>
    <forbidden xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>
```

否则，服务必须 MUST 修改管理员列表并通知所有者成功了：

例子 194. 服务通知所有者成功了

```
<iq from='darkcave@chat.shakespeare.lit'
  id='admin4'
  to='cronel@shakespeare.lit/desktop'
  type='result' />
```

服务必须 MUST 也为任何因前述管理员列表修改而导致的岗位变化而发送出席信息通知。

销毁房间

房间所有者必须 MUST 能够销毁一个房间，特别是如果这个房间不是持久房间的时候。流程如下：

1. 房间所有者请求销毁房间，如果必要的话指出一个原因 reason 和一个备用场地。
2. 该房间移除所有房客(包含适当的关于备用场地和被移除的原因的信息)并销毁房间，即使它被定义为持久房间。

不像前述的，本文不指定一个 MUC 服务实现收到一个销毁房间请求之后将会如何做。例如，如果房间定义为持久地，一个实现可以 MAY 选择锁定房间 I，这样它不能被重用，把加入该房间的请求重定向到替代场地，或邀请当前的参与者到新的房间；无论如何，这些行为是可选的 OPTIONAL。

为了销毁一个房间，房间所有者必须 MUST 发送一个 IQ set 指令到要销毁的房间的地址。这个 <iq/> 节将包含一个遵循 ['http://jabber.org/protocol/muc#owner'](http://jabber.org/protocol/muc#owner) 名字空间的 <query/> 元素，它将包含一个 <destroy/> 元素。替代场地的地址可以 MAY 用这个 <destroy/> 元

素的 'jid' 属性来提供. 一个密码保护的替代场地可以 MAY 通过 <destroy/> 元素的 <password/> 子元素的 XML 字符数据来提供. 摧毁房间的原因可以 MAY 通过 <destroy/> 元素的 <reason/> 子元素的 XML 字符数据来提供.

以下例子展示了协议发送和接收的元素:

例子 195. 所有者提交房间摧毁请求

```
<iq from='cronel@shakespeare.lit/desktop'
  id='begone'
  to='heath@chat.shakespeare.lit'
  type='set'>
  <query xmlns='http://jabber.org/protocol/muc#owner'>
    <destroy jid='darkcave@chat.shakespeare.lit'>
      <reason>Macbeth doth come.</reason>
    </destroy>
  </query>
</iq>
```

服务负责移除所有房客. 它不应该 SHOULD NOT 广播类型为“unavailable”的出席信息节给所有房客, 只需要发送一个“unavailable”类型的出席信息节给每个房客, 这样该用户知道他或她已经从房间移除了. 如果所有者的扩展出席信息指定了一个替代场地的 JID 以及房间销毁的原因, 这个出席信息节必须 MUST 包含那些信息.

例子 196. 服务移除每个房客

```
<presence
  from='heath@chat.shakespeare.lit/firstwitch'
  to='cronel@shakespeare.lit/desktop'
  type='unavailable'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='none' role='none' />
    <destroy jid='darkcave@chat.shakespeare.lit'>
      <reason>Macbeth doth come.</reason>
    </destroy>
  </x>
</presence>

<presence
  from='heath@chat.shakespeare.lit/secondwitch'
  to='wiccarocks@shakespeare.lit/laptop'
  type='unavailable'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='none' role='none' />
```

```

    <destroy jid='darkcave@chat.shakespeare.lit'>
      <reason>Macbeth doth come.</reason>
    </destroy>
  </x>
</presence>

<presence
  from='heath@chat.shakespeare.lit/thirdwitch'
  to='hag66@shakespeare.lit/pda'
  type='unavailable'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <item affiliation='none' role='none' />
    <destroy jid='darkcave@chat.shakespeare.lit'>
      <reason>Macbeth doth come.</reason>
    </destroy>
  </x>
</presence>

```

例子 197. 服务通知所有者成功销毁房间

```

<iq from='heath@chat.shakespeare.lit'
  id='begone'
  to='cronel@shakespeare.lit/desktop'
  type='result' />

```

如果在一个销毁请求中接收到的 'from' 地址的 <user@host> 和一个房间所有者的纯 JID 不符，服务必须 MUST 返回一个 <forbidden/> 错误给发送者：

例子 198. 服务拒绝由非所有者提交的销毁请求

```

<iq from='heath@chat.shakespeare.lit'
  id='destroytest'
  to='wiccarocks@shakespeare.lit/laptop'
  type='error'>
  <query xmlns='http://jabber.org/protocol/muc#owner'>
    <destroy jid='darkcave@chat.shakespeare.lit'>
      <reason>Macbeth doth come.</reason>
    </destroy>
  </query>
  <error type='auth'>
    <forbidden xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>

```


错误和状态码

错误码

和'<http://jabber.org/protocol/muc#user>' 名字空间相关的错误码相当简单，总结于下表之中。关于传统的错误码到 XMPP 格式的错误的映射的详细信息，参见 [错误条件映射 23](#)；实现应该 SHOULD 支持传统和 XMPP 错误处理两者。

表 9: <http://jabber.org/protocol/muc#user> 名字空间的错误码

码	类型	元素	上下文	目的
401	Error	Presence	进入一个房间	通知用户需要密码
403	Error	Presence	进入一个房间	通知用户他或她被房间禁止了
404	Error	Presence	进入一个房间	通知用户房间不存在
405	Error	Presence	进入一个房间	通知用户限制创建房间
406	Error	Presence	进入一个房间	通知用户必须使用保留的房间昵称
407	Error	Presence	进入一个房间	通知用户他或她不在成员列表中
409	Error	Presence	进入一个房间	通知用户他或她的房间昵称正在使用或被别的用户注册了
503	Error	Presence	进入一个房间	通知用户已经达到最大用户数

本文不规定和上述错误条件相关的文本字符串(即，XMPP `<text/>` 元素值)。

状态码

多用户聊天的使用一个 `<status/>` 元素(特指，`<status/>` 元素的 `'code'` 属性) 来传达关于用户在一个房间里的状态的信息。随着时间的推移，状态码的数量已经增加了很多，而新的状态码继续被作者申请。所以，这些状态码现在记录在一个由 XMPP 登记处维护的注册表里。细节可参考本文的 [状态码注册表](#)。

注意：通常，MUC 状态码倾向于沿用 [RFC 2616 24](#) 和 [RFC 1893 25](#) (1xx 码表示信息，2xx 码说明情况良好可继续，3xx 码指定重定向被踢或被禁止的用户，

x3x 码指系统状态, x7x 码指安全或策略事务, 等等) 里面的状态码的“抽象”含义.

注意: 如果今天来定义 MUC 协议, 它将指定一个更有弹性的, XML-友好的 途径而不是硬编码的状态数字; 然而, 现在修改状态汇报系统带来的痛苦将远大于好处, 这是为什么状态码数字保持使用至今. 本文的未来版本可能定义一个更 类 XMPP 的途径来表示状态条件, 保留状态码数字但是给它们补充更多的描述性的子元素, 就像 [RFC 3920](#) 里那样.

国际化事项

如 [RFC 3920](#) 中所定义的, XMPP 实体 (包括 MUC 房间和 MUC 服务) 应该 SHOULD 遵守任何给定的节提供的 'xml:lang' 属性. 然而, 群聊消息的同声翻译超出了本文的范围.

这里定义的状态和错误码允许一个客户端实现展示一个本地化的界面; 然而, 任何给定语言社区的本地化文本字符串的定义超出了本文范围.

尽管这里的很多数据表单字段的标签显示为英文, MUC 客户端应该 SHOULD 把这些字段展示为本地化的文本而不是英文文本.

安全事项

用户验证和授权

本文没有定义或要求比明文密码更安全的房间准入验证或授权方法. 然而, 这些潜在的风险可能使用 [RFC 3920](#) 描述的通过使用 TLS 和 SASL 加密通道来减轻.

端到端加密

这里没有定义没有端到端消息或会话加密方法. 用户不应该 SHOULD NOT 相信一个服务能保持通过房间发送的任何文本的安全.

隐私

取决于房间配置, 一个房间可以公开地记录房间里发生的所有讨论. 服务必须 MUST 警告用户该房间是公开记录的, 通过在该用户的初始出席信息中返回一个状态码“170”, 并且如果房间讨论被记录 (用户的客户端也应该 SHOULD 在允许用户进入之前查询房间的配置, 以“预先发现”房间是否被记录), 该用户的客

户端也必须 MUST 警告用户。如果房间的配置随后修改成允许房间记录(当房间发送状态码 170 时客户端将发现)，客户端也必须 MUST 警告用户。注意：房间内的历史和公开房间记录是不同的，并且很自然的一个房间不能有效地阻止房客独立维护的自有的房间记录，它可能被公开；用户应该 SHOULD 谨慎操作并认识到任何房间讨论可能被有效地公开。

匿名

取决于房间配置，一个房间可以 MAY 暴光每个房客的真实 JID 给其他房客(如果该房间是非匿名的)并且将几乎肯定地暴光每个房客的真实 JID 给该房间的所有者和管理员(如果该房间不是全匿名的)。服务必须 MUST 警告用户真实 JIDs 在房间被暴光，通过在该用户的初始出席信息中包含状态码 “100”，并且用户的客户端必须 MUST 警告该用户(一个用户的客户端应该 SHOULD 也在允许用户进入房间之前查询房间配置以“预先发现”是否真实 JIDs 会在房间中暴光)。如果房间配置随后从半匿名或全匿名修改成非匿名(当房间发送状态码 172 时客户端将发现)，客户端必须 MUST 也警告用户，如果房间的配置随后从全匿名改成半匿名时(当房间发送状态码 173 时客户端将发现)，客户端也应该 SHOULD 警告用户。

拒绝服务

公开的 MUC 房间能承受一定数量的攻击，大部分能减少拒绝服务攻击。这些攻击包括但不限于：

1. 向房间里塞进大量的非法房客从而阻止合法用户加入房间。
2. 发送侮辱性的消息接着在被踢或被禁止之前离开房间；这些侮辱性的消息包含但不限于，大量消息以阻止参与者正常跟踪会话线索或房间历史，对参与者的人身攻击（特别是房间管理员和主持人），攻击性的文字，以及垃圾网站链接。
3. 高频率的制造出席信息变更。
4. 使用过长的昵称导致无法看到完整的发言。
5. 辱骂房间管理员或其他房间房客。
6. 在一个服务里注册很多昵称然后禁止这些昵称的使用。
7. 模仿别的房客的昵称(例如，通过在尾部增加一个空格或看起来相似的字符串)，然后以那个房间昵称发送消息用于欺骗房客。

这些攻击可能被减轻不能完全被阻止，通过灵活地使用管理员操作。例如禁止用户，有管理员权限的自动的房间机器人出席信息，智能内容过滤的实现，检查连接的用户的 IP 地址(在分布式的系统里不一定能实现)，应用发言规则到出席信息以及消息，使用比 Resourceprep profile of stringprep 更严格的规则匹配房间昵称，等等。然而，经验表明无法完全阻止这类攻击。

其它事项

关于延迟递送符号的列入和流程的更多安全事项参见 XEP-0203.

IANA 事项

本文档与[互联网编号分配授权机构 26](#) 无关。

XMPP 登记事项

[XMPP 登记处 27](#) 在它的登记处包含了以下信息.

协议名字空间

XMPP 登记处在它的协议名字空间注册表里包含了以下 MUC 相关的名字空间:

- <http://jabber.org/protocol/muc>
- <http://jabber.org/protocol/muc#admin>
- <http://jabber.org/protocol/muc#owner>
- <http://jabber.org/protocol/muc#user>

服务发现种类/类型

一个多用户聊天服务或房间在服务发现里是用 “conference” 种类 category 和 “text” 类型 type 来标识的.

服务发现特性

有很多和 MUC 相关的服务或房间相关的特性可以被“服务发现”来发现. 这里面最基本的是 '<http://jabber.org/protocol/muc>' 名字空间. 另外, 一个 MUC 房间应该 SHOULD 提供关于它实现的特定房间特性的信息, 例如密码保护和房间主持.

注册提交

```
<var>
  <name>http://jabber.org/protocol/muc#register</name>
  <desc>Support for the muc#register FORM_TYPE</desc>
  <doc>XEP-0045</doc>
</var>
```

```
<var>
  <name>http://jabber.org/protocol/muc#roomconfig</name>
  <desc>Support for the muc#roomconfig FORM_TYPE</desc>
  <doc>XEP-0045</doc>
</var>
<var>
  <name>http://jabber.org/protocol/muc#roominfo</name>
  <desc>Support for the muc#roominfo FORM_TYPE</desc>
  <doc>XEP-0045</doc>
</var>
<var>
  <name>muc_hidden</name>
  <desc>Hidden room in Multi-User Chat (MUC)</desc>
  <doc>XEP-0045</doc>
</var>
<var>
  <name>muc_memberonly</name>
  <desc>Members-only room in Multi-User Chat (MUC)</desc>
  <doc>XEP-0045</doc>
</var>
<var>
  <name>muc_moderated</name>
  <desc>Moderated room in Multi-User Chat (MUC)</desc>
  <doc>XEP-0045</doc>
</var>
<var>
  <name>muc_nonanonymous</name>
  <desc>Non-anonymous room in Multi-User Chat (MUC)</desc>
  <doc>XEP-0045</doc>
</var>
<var>
  <name>muc_open</name>
  <desc>Open room in Multi-User Chat (MUC)</desc>
  <doc>XEP-0045</doc>
</var>
<var>
  <name>muc_passwordprotected</name>
  <desc>Password-protected room in Multi-User Chat (MUC)</desc>
  <doc>XEP-0045</doc>
</var>
<var>
  <name>muc_persistent</name>
  <desc>Persistent room in Multi-User Chat (MUC)</desc>
  <doc>XEP-0045</doc>
```

```

</var>
<var>
  <name>muc_public</name>
  <desc>Public room in Multi-User Chat (MUC)</desc>
  <doc>XEP-0045</doc>
</var>
<var>
  <name>muc_rooms</name>
  <desc>List of MUC rooms (each as a separate item)</desc>
  <doc>XEP-0045</doc>
</var>
<var>
  <name>muc_semianonymous</name>
  <desc>Semi-anonymous room in Multi-User Chat (MUC)</desc>
  <doc>XEP-0045</doc>
</var>
<var>
  <name>muc_temporary</name>
  <desc>Temporary room in Multi-User Chat (MUC)</desc>
  <doc>XEP-0045</doc>
</var>
<var>
  <name>muc_unmoderated</name>
  <desc>Unmoderated room in Multi-User Chat (MUC)</desc>
  <doc>XEP-0045</doc>
</var>
<var>
  <name>muc_unsecured</name>
  <desc>Unsecured room in Multi-User Chat (MUC)</desc>
  <doc>XEP-0045</doc>
</var>

```

知名服务发现节点

知名服务发现节点 '<http://jabber.org/protocol/muc#rooms>' 允许发现一个用户是哪个房间的房客。

知名服务发现节点 'x-roomuser-item' 允许一个用户从房间外发现自己的已注册房间昵称。

知名服务发现节点 '<http://jabber.org/protocol/muc#traffic>' 允许发现通过一个房间能发送哪些名字空间的通讯(参见本文允许的 [允许的通讯](#) 章节)。

字段标准化

[数据表单的字段标准化 28](#) 定义了用于遵循特定名字空间的数据表单的字段标准化的过程. 在 MUC 里面, 使用了四种这类表单: 房间注册 ("muc#register" FORM_TYPE), 请求发言权和批准请求 ("muc#request"), 房间配置 ("muc#roomconfig"), 以及用于房间信息的服务发现扩展 ("muc#roominfo"). 这些保留的字段定义如下.

muc#register FORM_TYPE

注册处提交

```
<form_type>
  <name>http://jabber.org/protocol/muc#register</name>
  <doc>XEP-0045</doc>
  <desc>
    Forms enabling user registration with a
    Multi-User Chat (MUC) room or admin approval
    of user registration requests.
  </desc>
  <field
    var='muc#register_allow'
    type='boolean'
    label='Allow this person to register with the room?' />
  <field
    var='muc#register_email'
    type='text-single'
    label='Email Address' />
  <field
    var='muc#register_faentry'
    type='text-multi'
    label='FAQ Entry' />
  <field
    var='muc#register_first'
    type='text-single'
    label='Given Name' />
  <field
    var='muc#register_last'
    type='text-single'
    label='Family Name' />
  <field
    var='muc#register_roomnick'
    type='text-single'
    label='Desired Nickname' />
```

```
<field
  var='muc#register_url'
  type='text-single'
  label='Your URL' />
</form_type>
```

muc#request FORM_TYPE

注册处提交

```
<form_type>
  <name>http://jabber.org/protocol/muc#request</name>
  <doc>XEP-0045</doc>
  <desc>
    Forms enabling voice requests in a
    Multi-User Chat (MUC) room or admin
    approval of such requests.
  </desc>
  <field var='muc#role'
    type='text-single'
    label='Requested role' />
  <field var='muc#jid'
    type='text-single'
    label='User ID' />
  <field var='muc#roomnick'
    type='text-single'
    label='Room Nickname' />
  <field var='muc#request_allow'
    type='boolean'
    label='Whether to grant voice' />
</form_type>
```

muc#roomconfig FORM_TYPE

注册处提交

```
<form_type>
  <name>http://jabber.org/protocol/muc#roomconfig</name>
  <doc>XEP-0045</doc>
  <desc>
    Forms enabling creation and configuration of
    a Multi-User Chat (MUC) room.
  </desc>
  <field
    var='muc#roomconfig_allowinvites'
```



```
    type='boolean'
    label='Whether to Allow Occupants to Invite Others' />
<field
  var='muc#roomconfig_changesubject'
  type='boolean'
  label='Whether to Allow Occupants to Change Subject' />
<field
  var='muc#roomconfig_enablelogging'
  type='boolean'
  label='Whether to Enable Public Logging of Room Conversations' />
<field
  var='muc#roomconfig_getmemberlist'
  type='list-multi'
  label='Roles and Affiliations that May Retrieve Member List' />
<field
  var='muc#roomconfig_lang'
  type='text-single'
  label='Natural Language for Room Discussions' />
<field
  var='muc#roomconfig_pubsub'
  type='text-single'
  label='XMPP URI of Associated Publish-Subscribe Node' />
<field
  var='muc#roomconfig_maxusers'
  type='list-single'
  label='Maximum Number of Room Occupants' />
<field
  var='muc#roomconfig_membersonly'
  type='boolean'
  label='Whether an Make Room Members-Only' />
<field
  var='muc#roomconfig_moderatedroom'
  type='boolean'
  label='Whether to Make Room Moderated' />
<field
  var='muc#roomconfig_passwordprotectedroom'
  type='boolean'
  label='Whether a Password is Required to Enter' />
<field
  var='muc#roomconfig_persistentroom'
  type='boolean'
  label='Whether to Make Room Persistent' />
<field
  var='muc#roomconfig_presencebroadcast'
```

```

        type='list-multi'
        label='Roles for which Presence is Broadcast' />
    <field
        var='muc#roomconfig_publicroom'
        type='boolean'
        label='Whether to Allow Public Searching for Room' />
    <field
        var='muc#roomconfig_roomadmins'
        type='jid-multi'
        label='Full List of Room Admins' />
    <field
        var='muc#roomconfig_roomdesc'
        type='text-single'
        label='Short Description of Room' />
    <field
        var='muc#roomconfig_roomname'
        type='text-single'
        label='Natural-Language Room Name' />
    <field
        var='muc#roomconfig_roomowners'
        type='jid-multi'
        label='Full List of Room Owners' />
    <field
        var='muc#roomconfig_roomsecret'
        type='text-private'
        label='The Room Password' />
    <field
        var='muc#roomconfig_whois'
        type='list-single'
        label='Affiliations that May Discover Real JIDs of Occupants' />
</form_type>

```

muc#roominfo FORM_TYPE

注册处提交

```

<form_type>
  <name>http://jabber.org/protocol/muc#roominfo</name>
  <doc>XEP-0045</doc>
  <desc>
    Forms enabling the communication of extended service discovery
    information about a Multi-User Chat (MUC) room.
  </desc>
  <field
    var='muc#roominfo_contactjid'

```

```

        type='jid-multi'
        label='Contact Addresses (normally, room owner or owners)' />
<field
    var='muc#roominfo_description'
    type='text-single'
    label='Short Description of Room' />
<field
    var='muc#roominfo_lang'
    type='text-single'
    label='Natural Language for Room Discussions' />
<field
    var='muc#roominfo_ldapgroup'
    type='text-single'
    label='An associated LDAP group that defines room membership;
           this should be an LDAP Distinguished Name according to an
           implementation-specific or deployment-specific definition
           of a group.' />
<field
    var='muc#roominfo_logs'
    type='text-single'
    label='URL for Archived Discussion Logs' />
<field
    var='muc#roominfo_occupants'
    type='text-single'
    label='Current Number of Occupants in Room' />
<field
    var='muc#roominfo_subject'
    type='text-single'
    label='Current Subject or Discussion Topic in Room' />
<field
    var='muc#roominfo_subjectmod'
    type='boolean'
    label='The room subject can be modified by participants' />
</form_type>

```

状态码登记处

过程

XMPP 注册员为遵循 '<http://jabber.org/protocol/muc#user>' 名字空间的 `<status/>` 元素的 'code' 属性维护注册表中的值。

为了提交新值给这个注册表，注册人将按以下格式定义一个 XML 段并把它包含在相关的 XMPP 扩展协议中，或者发到 `<registrar@xmpp.org>`：

```

<statuscode>
  <number>the three-digit code number</number>
  <stanza>the stanza type of which it is a child (message or
presence)</stanza>
  <context>the use case or situation in which the status is used</context>
  <purpose>a natural-language description of the meaning</purpose>
  <child>the descriptive child element (reserved for future use)</child>
</statuscode>

```

注册人可一次注册多个状态码，每个状态码包含在独立的<statuscode/>元素中。

初始提交

作为本文的一部分，以下状态码已被注册了：

```

<statuscode>
  <number>100</number>
  <stanza>message or presence</stanza>
  <context>Entering a room</context>
  <purpose>Inform user that any occupant is allowed to see the user's full
JID</purpose>
</statuscode>
<statuscode>
  <number>101</number>
  <stanza>message (out of band)</stanza>
  <context>Affiliation change</context>
  <purpose>Inform user that his or her affiliation changed while not in
the room</purpose>
</statuscode>
<statuscode>
  <number>102</number>
  <stanza>message</stanza>
  <context>Configuration change</context>
  <purpose>Inform occupants that room now shows unavailable
members</purpose>
</statuscode>
<statuscode>
  <number>103</number>
  <stanza>message</stanza>
  <context>Configuration change</context>
  <purpose>Inform occupants that room now does not show unavailable
members</purpose>
</statuscode>
<statuscode>
  <number>104</number>

```

```
<stanza>message</stanza>
<context>Configuration change</context>
<purpose>
  Inform occupants that a non-privacy-related room configuration
  change has occurred
</purpose>
</statuscode>
<statuscode>
  <number>110</number>
  <stanza>presence</stanza>
  <context>Any room presence</context>
  <purpose>Inform user that presence refers to one of its own room
  occupants</purpose>
</statuscode>
<statuscode>
  <number>170</number>
  <stanza>message or initial presence</stanza>
  <context>Configuration change</context>
  <purpose>Inform occupants that room logging is now enabled</purpose>
</statuscode>
<statuscode>
  <number>171</number>
  <stanza>message</stanza>
  <context>Configuration change</context>
  <purpose>Inform occupants that room logging is now disabled</purpose>
</statuscode>
<statuscode>
  <number>172</number>
  <stanza>message</stanza>
  <context>Configuration change</context>
  <purpose>Inform occupants that the room is now non-anonymous</purpose>
</statuscode>
<statuscode>
  <number>173</number>
  <stanza>message</stanza>
  <context>Configuration change</context>
  <purpose>Inform occupants that the room is now
  semi-anonymous</purpose>
</statuscode>
<statuscode>
  <number>174</number>
  <stanza>message</stanza>
  <context>Configuration change</context>
```

```
<purpose>Inform occupants that the room is now
fully-anonymous</purpose>
</statuscode>
<statuscode>
  <number>201</number>
  <stanza>presence</stanza>
  <context>Entering a room</context>
  <purpose>Inform user that a new room has been created</purpose>
</statuscode>
<statuscode>
  <number>210</number>
  <stanza>presence</stanza>
  <context>Entering a room</context>
  <purpose>Inform user that service has assigned or modified occupant's
roomnick</purpose>
</statuscode>
<statuscode>
  <number>301</number>
  <stanza>presence</stanza>
  <context>Removal from room</context>
  <purpose>Inform user that he or she has been banned from the
room</purpose>
</statuscode>
<statuscode>
  <number>303</number>
  <stanza>presence</stanza>
  <context>Exiting a room</context>
  <purpose>Inform all occupants of new room nickname</purpose>
</statuscode>
<statuscode>
  <number>307</number>
  <stanza>presence</stanza>
  <context>Removal from room</context>
  <purpose>Inform user that he or she has been kicked from the
room</purpose>
</statuscode>
<statuscode>
  <number>321</number>
  <stanza>presence</stanza>
  <context>Removal from room</context>
  <purpose>Inform user that he or she is being removed from the room
because of an affiliation change</purpose>
</statuscode>
<statuscode>
```

```

<number>322</number>
<stanza>presence</stanza>
<context>Removal from room</context>
<purpose>Inform user that he or she is being removed from the room
    because the room has been changed to members-only and the user
    is not a member</purpose>
</statuscode>
<statuscode>
    <number>332</number>
    <stanza>presence</stanza>
    <context>Removal from room</context>
    <purpose>Inform user that he or she is being removed from the room
        because of a system shutdown</purpose>
</statuscode>

```

URI 查询类型

作为由 [XMPP URI Query Components](http://www.xmpp.org/registrars/querytypes.html) [29](#) 授权的机构，XMPP 注册员维护着一个用于 XMPP URIs 的查询和键-值对的注册表（见 <http://www.xmpp.org/registrars/querytypes.html>）。

join

“join” 查询类型被注册为一个 MUC 相关的动作，伴随一个可选的键 “password”。

例子 199. Join 动作: IRI/URI

```
xmpp:darkcave@chat.shakespeare.lit?join
```

应用必须 MUST 要么展示一个界面允许用户提供一个房间昵称，要么基于配置好的选项或昵称发现来获取这个房间昵称。

例子 200. Join 动作: 结果节

```

<presence to='darkcave@chat.shakespeare.lit/thirdwitch'>
  <x xmlns='http://jabber.org/protocol/muc' />
</presence>

```

join 动作可以 MAY 为这房间包含一个密码。自然的，对一个包含了房间密码的 URI 的访问必须 MUST 得到适当的控制。

例子 201. Join 动作包含密码: IRI/URI

```
xmpp:darkcave@chat.shakespeare.lit?join;password=cauldronburn
```

例子 202. Join 动作包含密码: 结果节

```
<presence to='darkcave@chat.shakespeare.lit/thirdwitch'>
  <x xmlns='http://jabber.org/protocol/muc'>
    <password>cauldronburn</password>
  </x>
</presence>
```

以下提交注册 “join” 查询类型.

```
<querytype>
  <name>join</name>
  <proto>http://jabber.org/protocol/muc</proto>
  <desc>enables joining a multi-user chat room</desc>
  <doc>XEP-0045</doc>
  <keys>
    <key>
      <name>password</name>
      <desc>the password required to enter a multi-user chat room</desc>
    </key>
  </keys>
</querytype>
```

invite

“invite” 查询类型被注册为一个 MUC 相关的动作, 伴随一个可选的键 “jid”.

例子 203. Invite 动作: IRI/URI

```
xmpp:darkcave@chat.shakespeare.lit?invite;jid=hecate@shakespeare.lit
```

如果加入中的用户还未在房间里, 应用必须 MUST 发送两个节: 第一个加入房间, 第二个邀请另一个人. 如果加入中的用户已经在房间里, 应用将只发送邀请节.

例子 204. Invite 动作: 结果节(s)

```
<presence to='darkcave@chat.shakespeare.lit/thirdwitch'>
  <x xmlns='http://jabber.org/protocol/muc' />
</presence>

<message to='darkcave@chat.shakespeare.lit'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
```



```
<invite to='hecate@shakespeare.lit' />
</x>
</message>
```

URI 可以包含多个邀请:

例子 205. Invite 动作包含多个邀请: IRI/URI

```
xmpp:darkcave@chat.shakespeare.lit?invite;jid=hecate@shakespeare.lit;
jid=bard@shakespeare.lit
```

例子 206. Invite 动作包含多个邀请: 结果节

```
<message to='darkcave@chat.shakespeare.lit'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <invite to='hecate@shakespeare.lit' />
    <invite to='bard@shakespeare.lit' />
  </x>
</message>
```

URI 也可以包含一个密码:

例子 207. Invite 动作包含密码: IRI/URI

```
xmpp:darkcave@chat.shakespeare.lit?invite;jid=hecate@shakespeare.lit;
password=cauldronburn
```

如果加入中的用户还未在房间里, 应用必须 MUST 发送两个节: 第一个加入房间, 第二个邀请另一个人. 如果加入中的用户已经在房间里, 应用将只发送邀请节.

例子 208. Invite 动作包含密码: 结果节(s)

```
<presence to='darkcave@chat.shakespeare.lit/thirdwitch'>
  <x xmlns='http://jabber.org/protocol/muc' />
</presence>

<message to='darkcave@chat.shakespeare.lit'>
  <x xmlns='http://jabber.org/protocol/muc#user'>
    <invite to='hecate@shakespeare.lit' />
    <password>cauldronburn</password>
  </x>
</message>
```

以下提交注册 “invite” 查询类型.

```

<querytype>
  <name>invite</name>
  <proto>http://jabber.org/protocol/muc</proto>
  <desc>enables simultaneously joining a groupchat room and inviting
others</desc>
  <doc>XEP-0045</doc>
  <keys>
    <key>
      <name>jid</name>
      <desc>the Jabber ID of the invitee</desc>
    </key>
    <key>
      <name>password</name>
      <desc>the password required to enter a multi-user chat room</desc>
    </key>
  </keys>
</querytype>

```

商业规则

Addresses

为了提供关于从房间 JIDs 抓获的地址的一致性，Room IDs 必须 MUST 遵循 Stringprep 的 Nodeprep 规范并且 Room Nicknames 必须 MUST 符合 Stringprep（这些都定义在 [RFC 3920](#)）的 Resourceprep 规范。尽管在 [RFC 3920](#) 中没有显式的说明，一个 Room JID 的 Room ID (node) 和 Room Nickname (resource) 部分都必须 MUST 长度不为零。另外，一个 MUC 服务不能 MUST NOT 允许空的或不可见的房间昵称 Room Nicknames（即，房间昵称 Room Nicknames 只包含一个或多个空格）。

取决于服务实现，是否更多地限制房间昵称（例如，通过应用情景例程，stringprep 的 Nodeprep 规范，或其他限制）。

Message

1. 如果一个房客想发送一个消息给所有其他房客，MUC 客户端必须 MUST 把 'type' 属性值设为 "groupchat"。服务可以 MAY 忽略不正确的消息类型，或用 <bad-request/> 错误弹回。
2. 如果一个 MUC 服务从一个角色为 "none" 的 Jabber 用户收到一个发送给该房间的消息或给某个房客的消息，服务不能 MUST NOT 递送这个消息并应该 SHOULD 返回给这个消息给发送者并伴随一个 <forbidden/> 错误。

3. 如果一个 MUC 服务 接收到一个发送给不存在的或尚未解锁的房的消息, 服务应该 SHOULD 返回这个消息给发送者并伴随一个 `<item-not-found/>` 错误.
4. 一个 MUC 服务应该 SHOULD 不做修改地传递扩展的消息 (例如, 一个消息主体的 XHTML 版本) 给房客; 然而, 一个 MUC 服务可以 MAY 不允许消息的特定扩展 (参见本文的[允许的通讯](#)章节).
5. 一个 MUC 客户端可以 MAY 生成扩展以满足 [消息事件 30](#) 或 [聊天状态通知 31](#) 规范; 然而, 一个 MUC 服务可以 MAY 不允许这些扩展 (参见本文的[允许的通讯](#)章节).

Presence

1. 一个房间必须 MUST 安静地忽略从一个角色为“none”的用户发来的不可用出席信息.
2. 只有 MUC 服务自身应该 SHOULD 生成关于角色, 岗位, 全 JIDs 或遵循 '<http://jabber.org/protocol/muc#user>' 名字空间的状态码的扩展的出席信息 (基于服务所知道的关于房客的信息, 例如, 角色, 或由一个主持人或房间管理员的动作所产生的结果). 一个客户端不应该 SHOULD NOT 推定生成这类信息. 如果一个 MUC 服务从一个房客接收到这类扩展的出席信息, 它不能 MUST NOT 反射它给其他房客们. (一个客户端可以 MAY 为了提供一个密码而生成遵循 '<http://jabber.org/protocol/muc#user>' 名字空间的扩展的出席信息, 但自然的这是不反射给其他房客的.)
3. 一个 MUC 服务应该 SHOULD 允许所有其他出席信息通过, 尽管它可以 MAY 选择阻塞扩展的出席信息; 参见本文的 [允许的通讯](#) 章节.
4. 为了适当地通知房客角色和岗位, 并使之更易于 Jabber 客户端跟踪房间里所有用户的当前状态, MUC 服务实现必须 MUST 在所有出席信息节里提供关于角色和岗位的扩展的出席信息, 包括一个用户因为任何原因退出该房间时被发送的类型为“unavailable”的出席信息节.
5. 如果一个权限被撤销, 服务必须 MUST 通知这件事, 通过发送一个遵循 '<http://jabber.org/protocol/muc#user>' 名字空间的 `<x/>` 元素并包含一个 `<item/>` 子元素, 该`<item/>` 子元素的 'role' 和/或 'affiliation' 属性值的设定指明是去了有关的权限. 所有将来的为这个房客发出的出席信息节必须 MUST 包含这个更新的角色和岗位, 直到除非它们再次改变.
6. 一个 MUC 服务必须 MUST 发送扩展的出席信息给一个客户端, 即使客户端在进入该房间时没有发送一个空的遵循 '<http://jabber.org/protocol/muc>' 名字空间的 `<x/>` 元素 ; 自然的, 一个客户端必须 MUST 忽略这类信息, 如果它不懂得它的话 (根据 [RFC 3920](#)).
7. 在 `muc#user` 名字空间中被发送的关于角色和岗位的扩展的出席信息必须 MUST 包含全 JID (不是纯 JID) 作为 'jid' 属性的值.

8. 如果想要，一个客户端可以 MAY 发送一个客户化的退出消息（就像 IRC 频道里经常出现的那样），通过在退出时被发送的“unavailable”类型的出席信息节里包含一个 <status/> 元素.

IQ

1. MUC 被设计用于分享消息和出席信息，而不是 IQs. 一个被发送到房间本身 JID 的 IQ 由房间本身来处理并且不反射给所有房客.
2. 如果一个房客想在一个非匿名房间发送一个 IQ 节给其他用户，发送者应该 SHOULD 直接发送请求给接收者的纯 JID 或全 JID，而不是试图通过房间发送请求（即，通过接收者的房间 JID）.
3. 如果一个房客想在一个半匿名房间发送一个 IQ 节给其他用户，发送者能直接发送这个节给接受者的房间 JID 并且服务可以 MAY 转发这个节给接收者的真实 JID. 然而，任何时候一个 MUC 服务不能 MUST NOT 泄露这个发送者的真实 JID 给接收者，也不能泄露接收者的真实 JID 给发送者.
4. 一个 MUC 客户端必须 MUST 在 IQ set 中的遵循
'<http://jabber.org/protocol/muc#admin>' <item/> 子元素中只发送 'affiliation' 属性或 'role' 属性；如果一个主持人，管理员，或所有者试图在相同的 IQ set 中修改相同条目的岗位或角色，服务必须 MUST 返回一个 <bad-request/> 错误给发送者. 无论如何，一个 MUC 服务可以 MAY 基于一个岗位的变更来修改一个角色，从而可以 MAY 发送出席信息更新，同时包含一个修改的角色和一个修改的岗位.
5. 在关于角色的 IQ sets 中，一个 MUC 客户端必须 MUST 只包含 'nick' 属性；在关于角色的 IQ results 中，一个 MUC 服务必须 MUST 包含 'nick', 'role', 'affiliation', 和 'jid' 属性（值为后来设置的用户的全 JID）.
6. 在关于岗位的 IQ sets 中，一个 MUC 客户端必须 MUST 只包含 'jid' 属性（值为纯 JID）；在关于岗位的 IQ results 中，一个 MUC 服务不能 MUST NOT 包含 'role' 属性，必须 MUST 包含 'affiliation' 属性和 'jid' 属性（值为纯 JID），并且应该 SHOULD 包含 'nick' 属性（除非岗位为“outcast”，以为被排斥者不应该 SHOULD NOT 有保留的房间昵称）.

实现注意事项

以下方针有助于客户端和组件开发者建立 MUC 实现.

服务端

1. 在处理一个被主持的房间里游客发送的消息时，一个 MUC 服务可以 MAY 通过一个主持人让每个消息排队等待批准并且可以 MAY 通知发送者消息正在等待批准；然而，这一行为是可选的 OPTIONAL，并且一个消息批准

协议的定义（例如，使用 XEP-0004 定义的数据表单）超出了本文的范围。

2. 对于一个 MUC 服务来说，在特定时间发生时提供房间内的消息是很常见的，例如当标题变更时，当一个房客加入或退出时，或当一个房间被销毁时。这类消息完全是可选的 OPTIONAL 并且留给实现或部署来决定，但如果使用了，则必须 MUST 是从房间 JID 本身(<room@service>) 而不是从一个特定的

房客(<room@service/nick>)发送的类型为“groupchat”类型的消息。无论如何，通常接收的客户端倾向于基于房间的事件以及 MUC 提供的特定状态码来生成类似的消息（例如，用户加入或退出）；这将帮助确保这类消息的正确的本地化。

1. 出于礼貌，一个 MUC 服务可以 MAY 发送一个房间外的 <message/> 给一个被踢的或被禁止的房客，并且可以 MAY 广播一个房间内的 <message/> 给所有剩余的房客通知他们该房客已被该房间踢出或禁止。无论如何，这类消息是可选的 OPTIONAL，并且事实上是多余的，因为接收的客户端生成这类消息所必需的信息已经通过 MUC 服务发送的出席信息节（特别是状态码）得到了。
2. 出于礼貌，如果一个用户的岗位变更了而该用户不在房间里，一个 MUC 服务可以 MAY 发送一个房间外的 <message/>；这消息应该 SHOULD 被从房间发送给该用户的纯 JID，可以 MAY 包含一个 <body/> 元素描述岗位变更，并且必须 MUST 包含一个状态码 101。
3. 没有需求要一个 MUC 服务将为旧的“groupchat 1.0”用户提供特别的治疗，例如包含等价于扩展的遵循 <http://jabber.org/protocol/muc#user> 名字空间的出席信息的信息。
4. 房间类型可以 MAY 被配置成任何组合。一个 MUC 服务可以 MAY 支持或允许任何想要的房间类型或它们的组合。
5. 一个 MUC 服务可以 MAY 限制在初始配置完成之后配置选项展示给一个所有者的次数，例如因为特定的选项除非重启服务无法生效。
6. 一个 MUC 服务可以 MAY 提供一个接口给房间创建和配置（例如，以一个特定的 Jabber 表单或一个网页），这样表面上房间所有者是一个应用而不是一个自然人用户。
7. 一个 MUC 服务可以 MAY 选择让一个特定的房间内资源提供接口给管理功能（例如，一个“user”名的机器人“ChatBot”），房客们可以和它直接互动，从而允许管理员在一个私有消息里键入命令参数‘/command parameter’给那个机器人“user”。显然这种服务要求服务在房间创建时添加一个‘ChatBot’用户到房间，并且阻止任何房客在该房间使用房间昵称‘ChatBot’。这可能在某些实现或部署中比较难以保证。任何情况下，任何这类接口是可选的 OPTIONAL。
8. 如果服务接收到它之前发送给该用户的节相关的递送类错误，一个 MUC 服务应该 SHOULD 移除一个用户；递送相关的错误即 <gone/>，<item-not-found/>，<recipient-unavailable/>，<redirect/>，<remote-server-not-found/>，和 <remote-server-timeout/>。

9. 一个 MUC 服务可以 MAY 选择在反射出席信息变更给一个房间的房客们之前，抛弃附加在<presence/> 节上的扩展的出席信息。也就是，一个实现可以 MAY 选择只反射该出席信息节的 <show/>, <status/>, 和 <priority/> 子元素, 如 'jabber:client' 名字空间描述的 XML 架构之中，结果导致那个在扩展的名字空间中的出席信息变更 “changes” (例如, gabber:x:music:info) 不被传递给房客。如果一个服务禁止特定的扩展名字空间，它应该 SHOULD 在本文 [允许的通讯](#) 章节描述的知名的服务发现节点 '<http://jabber.org/protocol/muc#traffic>' 提供一个允许的通讯的描述。
10. 一个 MUC 服务可以 MAY 在反射消息给一个房间的房客之前选择抛弃附加在 <message/> 节的扩展信息。一个这类扩展信息的例子是轻量级文本标记，定义于 [XHTML-IM](#) [XEP-0045#附录 G:备注|32]。如果一个服务禁止特定的扩展名字空间，它应该 SHOULD 在本文 [允许的通讯](#) 章节描述的知名的服务发现节点 '<http://jabber.org/protocol/muc#traffic>' 提供一个允许的通讯的描述。
11. 一个 MUC 服务可以 MAY 选择锁定 “lock down” 房间昵称 (例如，硬编码房间昵称给该房客的纯 JID)。如果这么干，该服务必须 MUST 把被锁定的昵称看作一个保留的房间昵称并且必须 MUST 支持本文[发现保留的房间昵称](#) 章节定义的协议。

允许的通讯

大家知道，一个服务（更准确地说，一个正确配置的房间）可以 MAY 抛弃一些或所有的扩展的附加在从发送者通过房间反射给所有房客的 <message/> 和 <presence/> 节的名字空间。如果房间这么干，它应该 SHOULD 允许发送者通过发送 disco#info 查询知名的服务发现节点 '<http://jabber.org/protocol/muc#traffic>' 来发现允许的扩展的列表，在结果中返回支持的名字空间每个用一个 <feature/> 元素表示。如果该房间不允许任何扩展的名字空间，它必须 MUST 如 **XEP-0030** 所述返回一个空的 query。如果该房间不支持 “#traffic” 节点，它必须 MUST 返回一个 <feature-not-implemented/> 错误应答给查询发送到 '<http://jabber.org/protocol/muc#traffic>' 节点的查询。

以下例子展示一个只允许 '<http://jabber.org/protocol/xhtml-im>' 和 '<http://jabber.org/protocol/rosterx>' 名字空间的房间，而不包括其他的名字空间。

例子 209. 用户查询服务关于允许的名字空间

```
<iq from='wicarocks@shakespeare.lit/laptop'
  to='heath@chat.shakespeare.lit'
  id='allow1'
  type='get'>
  <query xmlns='http://jabber.org/protocol/disco#info'
    node='http://jabber.org/protocol/muc#traffic' />
```



```
</iq>
```

例子 210. 服务返回允许的名字空间

```
<iq from='heath@chat.shakespeare.lit'
  to='wiccarocks@shakespeare.lit/laptop'
  id='allow1'
  type='result'>
  <query xmlns='http://jabber.org/protocol/disco#info'
    node='http://jabber.org/protocol/muc#traffic'>
    <feature var='http://jabber.org/protocol/xhtml-im' />
    <feature var='http://jabber.org/protocol/rosterx' />
  </query>
</iq>
```

如果一个服务不抛弃任何名字空间或不实现这个特性，它必须 MUST 返回一个 `<service-unavailable/>` 错误：

例子 211. 服务返回服务不可用错误

```
<iq from='heath@chat.shakespeare.lit'
  to='wiccarocks@shakespeare.lit/laptop'
  id='allow1'
  type='error'>
  <query xmlns='http://jabber.org/protocol/disco#info'
    node='http://jabber.org/protocol/muc#traffic' />
  <error type='cancel'>
    <service-unavailable
xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  </error>
</iq>
```

客户端

1. Jabber 客户端可以 MAY 展示房间角色，通过在一个房间名册里为每个角色显示特定的群。这将使房客清楚图形化的知道哪个房客是主持人，参与者，和游客。无论如何，这样一个展示是可选的 OPTIONAL.
2. Jabber 客户端可以 MAY 实现多样化的界面以提供快捷方式“shortcuts”给功能，例如修改某人昵称，踢人或禁止用户，发现一个房客的全 JID，或修改主题。一个选项包含了类 IRC 的命令例如 `’/nick’`，`’/kick’`，`’/ban’`，和 `’/whois’`；另一个是使用户能用鼠标右击房间名册里的项目。所有这些界面形式是可选的 OPTIONAL。然而，为方便起见，下面提供了一个 IRC 命令到 MUC 协议的映射。

IRC 命令映射

IRC 客户端使用大量常用的快捷方式 “shortcut” 命令, 以一个斜杠开始, 例如 ‘/nick’ and ‘/ban’. 下表提供一个 类 IRC 命令到 MUC 协议的映射, 用于希望支持类似功能的 Jabber 客户端.

表 10: IRC 命令映射

命令	功能	MUC 协议
/ban <roomnick> [comment]	在房间里以房间昵称禁止用户(客户端翻译房间昵称为纯 JID)	<pre><iq id='someid' to='room@service' type='set'> <query xmlns='http://jabber.org/protocol/muc#admin'> <item affiliation='outcast' jid='bare-jid-of-user'> <reason>comment</reason> </item> </query> </iq></pre>
/invite <jid> [comment]	以 JID 邀请用户到此房间	<pre><message to='room@service'> <x xmlns='http://jabber.org/protocol/muc#user'> <invite to='jid'> <reason>comment</reason> </invite> </x> </message></pre>
/join <roomname> [pass]	在服务里加入房间(房间昵称同本房间内的昵称)	<pre><presence to='room@service/nick'> <x xmlns='http://jabber.org/protocol/muc#user'> <password>pass</password> </x> </presence></pre>
/kick <roomnick> [comment]	以房间昵称从房间里踢人	<pre><iq id='someid' to='room@service' type='set'> <query xmlns='http://jabber.org/protocol/muc#admin'> <item nick='roomnick' role='none'> <reason>comment</reason> </item> </query></pre>

		</iq>
/msg <roomnick> <foo>	发送私有消息"foo"给房间昵称	<message to='room@service/nick' type='chat'> <body>foo</body> </message>
/nick <newnick>	变更在此房间内的昵称为"newnick"	<presence to='room@service/newnick' />
/part [comment]	退出本房间 (一些 IRC 客户端也支持 /leave)	<presence to='room@service/nick' type='unavailable'> <status>comment</status> </presence>
/topic <foo>	变更此房间主题为"foo"	<message to='room@service' type='groupchat'> <subject>foo</subject> </message>

注意：因为 MUC 房间昵称遵循 stringprep 的 Resourceprep 脚本，它们被允许包含一个空格字符，而 IRC 昵称不允许。尽管一个给定的客户端可以 MAY 支持引用字符串用于这个目的（导致命令类似 '/ban "king lear" insanity is no defense'），最常见的引用字符（类似 " 和 '）也是被 Resourceprep 允许的，从而导致增加了复杂性和包含空格和引号的房间昵称中引号的潜在问题。所以不建议 NOT RECOMMENDED Jabber 客户端支持包含了空格符的房间昵称的类 IRC 的快捷方式命令。

注意：很多 Jabber 客户端也实现了 '/me' 命令，如 [The /me Command 33](#) 所述。这个命令不会导致任何 MUC 或 IRC 协议的动作所以不显式在上表中。

XML 架构

<http://jabber.org/protocol/muc>

```
<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='http://jabber.org/protocol/muc'
  xmlns='http://jabber.org/protocol/muc'
  elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
```

```

XEP-0045: http://www.xmpp.org/extensions/xep-0045.html
</xs:documentation>
</xs:annotation>

<xs:element name='x'>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref='history' minOccurs='0' />
      <xs:element name='password' type='xs:string' minOccurs='0' />
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name='history'>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base='empty'>
        <xs:attribute name='maxchars' type='xs:int' use='optional' />
        <xs:attribute name='maxstanzas' type='xs:int'
use='optional' />
        <xs:attribute name='seconds' type='xs:int' use='optional' />
        <xs:attribute name='since' type='xs:dateTime'
use='optional' />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

<xs:simpleType name='empty'>
  <xs:restriction base='xs:string'>
    <xs:enumeration value='' />
  </xs:restriction>
</xs:simpleType>

</xs:schema>

```

<http://jabber.org/protocol/muc#user>

```
<?xml version='1.0' encoding='UTF-8'?>
```

```

<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='http://jabber.org/protocol/muc#user'

```

```

xmlns='http://jabber.org/protocol/muc#user'
elementFormDefault='qualified' >

<xs:annotation>
  <xs:documentation>
    The protocol documented by this schema is defined in
    XEP-0045: http://www.xmpp.org/extensions/xep-0045.html
  </xs:documentation>
</xs:annotation>

<xs:element name='x' >
  <xs:complexType>
    <xs:choice minOccurs='0' maxOccurs='unbounded' >
      <xs:element ref='decline' minOccurs='0' />
      <xs:element ref='destroy' minOccurs='0' />
      <xs:element ref='invite' minOccurs='0' maxOccurs='unbounded' />
      <xs:element ref='item' minOccurs='0' />
      <xs:element name='password' type='xs:string' minOccurs='0' />
      <xs:element ref='status' minOccurs='0' maxOccurs='unbounded' />
    </xs:choice>
  </xs:complexType>
</xs:element>

<xs:element name='decline' >
  <xs:complexType>
    <xs:sequence>
      <xs:element ref='reason' minOccurs='0' />
    </xs:sequence>
    <xs:attribute name='from' type='xs:string' use='optional' />
    <xs:attribute name='to' type='xs:string' use='optional' />
  </xs:complexType>
</xs:element>

<xs:element name='destroy' >
  <xs:complexType>
    <xs:sequence>
      <xs:element ref='reason' minOccurs='0' />
    </xs:sequence>
    <xs:attribute name='jid' type='xs:string' use='optional' />
  </xs:complexType>
</xs:element>

<xs:element name='invite' >
  <xs:complexType>

```

```

    <xs:sequence>
      <xs:element ref='reason' minOccurs='0' />
    </xs:sequence>
    <xs:attribute name='from' type='xs:string' use='optional' />
    <xs:attribute name='to' type='xs:string' use='optional' />
  </xs:complexType>
</xs:element>

<xs:element name='item'>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref='actor' minOccurs='0' />
      <xs:element ref='reason' minOccurs='0' />
      <xs:element ref='continue' minOccurs='0' />
    </xs:sequence>
    <xs:attribute name='affiliation' use='optional'>
      <xs:simpleType>
        <xs:restriction base='xs:NCName'>
          <xs:enumeration value='admin' />
          <xs:enumeration value='member' />
          <xs:enumeration value='none' />
          <xs:enumeration value='outcast' />
          <xs:enumeration value='owner' />
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name='jid' type='xs:string' use='optional' />
    <xs:attribute name='nick' type='xs:string' use='optional' />
    <xs:attribute name='role' use='optional'>
      <xs:simpleType>
        <xs:restriction base='xs:NCName'>
          <xs:enumeration value='moderator' />
          <xs:enumeration value='none' />
          <xs:enumeration value='participant' />
          <xs:enumeration value='visitor' />
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>

<xs:element name='actor'>
  <xs:complexType>
    <xs:simpleContent>

```

```

        <xs:extension base='empty' >
            <xs:attribute name='jid' type='xs:string' use='required' />
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
</xs:element>

<xs:element name='continue' >
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base='empty' >
                <xs:attribute name='thread' type='xs:string'
use='optional' />
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>

<xs:element name='status' >
    <xs:complexType>
        <xs:attribute name='code' use='required' >
            <xs:simpleType>
                <xs:restriction base='xs:int' >
                    <xs:minInclusive value='100' />
                    <xs:maxInclusive value='999' />
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
</xs:element>

<xs:element name='reason' type='xs:string' />

<xs:simpleType name='empty' >
    <xs:restriction base='xs:string' >
        <xs:enumeration value='' />
    </xs:restriction>
</xs:simpleType>

</xs:schema>

```

<http://jabber.org/protocol/muc#admin>

```

<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='http://jabber.org/protocol/muc#admin'
  xmlns='http://jabber.org/protocol/muc#admin'
  elementFormDefault='qualified'>

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0045: http://www.xmpp.org/extensions/xep-0045.html
    </xs:documentation>
  </xs:annotation>

  <xs:element name='query'>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref='item' maxOccurs='unbounded' />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name='item'>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref='actor' minOccurs='0' />
        <xs:element ref='reason' minOccurs='0' />
      </xs:sequence>
      <xs:attribute name='affiliation' use='optional'>
        <xs:simpleType>
          <xs:restriction base='xs:NCName'>
            <xs:enumeration value='admin' />
            <xs:enumeration value='member' />
            <xs:enumeration value='none' />
            <xs:enumeration value='outcast' />
            <xs:enumeration value='owner' />
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name='jid' type='xs:string' use='optional' />
      <xs:attribute name='nick' type='xs:string' use='optional' />
      <xs:attribute name='role' use='optional'>
        <xs:simpleType>

```

```

        <xs:restriction base='xs:NCName'>
            <xs:enumeration value='moderator' />
            <xs:enumeration value='none' />
            <xs:enumeration value='participant' />
            <xs:enumeration value='visitor' />
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>

<xs:element name='actor'>
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base='empty'>
                <xs:attribute name='jid' type='xs:string' use='required' />
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>

<xs:element name='reason' type='xs:string' />

<xs:simpleType name='empty'>
    <xs:restriction base='xs:string'>
        <xs:enumeration value='' />
    </xs:restriction>
</xs:simpleType>

</xs:schema>

```

<http://jabber.org/protocol/muc#owner>

```

<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
    xmlns:xs='http://www.w3.org/2001/XMLSchema'
    targetNamespace='http://jabber.org/protocol/muc#owner'
    xmlns='http://jabber.org/protocol/muc#owner'
    elementFormDefault='qualified'>

    <xs:annotation>
        <xs:documentation>

```

```

    The protocol documented by this schema is defined in
    XEP-0045: http://www.xmpp.org/extensions/xep-0045.html
  </xs:documentation>
</xs:annotation>

<xs:import
  namespace='jabber:x:data'
  schemaLocation='http://www.xmpp.org/schemas/x-data.xsd' />

<xs:element name='query'>
  <xs:complexType>
    <xs:choice xmlns:xdata='jabber:x:data' minOccurs='0'>
      <xs:element ref='xdata:x' />
      <xs:element ref='destroy' />
    </xs:choice>
  </xs:complexType>
</xs:element>

<xs:element name='destroy'>
  <xs:complexType>
    <xs:sequence>
      <xs:element name='password' type='xs:string' minOccurs='0' />
      <xs:element name='reason' type='xs:string' minOccurs='0' />
    </xs:sequence>
    <xs:attribute name='jid' type='xs:string' use='optional' />
  </xs:complexType>
</xs:element>

<xs:simpleType name='empty'>
  <xs:restriction base='xs:string'>
    <xs:enumeration value='' />
  </xs:restriction>
</xs:simpleType>

</xs:schema>

```

<http://jabber.org/protocol/muc#unique>

```
<?xml version='1.0' encoding='UTF-8'?>
```

```

<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='http://jabber.org/protocol/muc#unique'

```



```
xmlns='http://jabber.org/protocol/muc#unique'
elementFormDefault='qualified'>

<xs:annotation>
  <xs:documentation>
    The protocol documented by this schema is defined in
    XEP-0045: http://www.xmpp.org/extensions/xep-0045.html
  </xs:documentation>
</xs:annotation>

<xs:element name='unique' type='xs:string' />

</xs:schema>
```

鸣谢

作者感谢以下个人, 为他们很多对于本协议草案的帮助性的评论: David Sutton, Peter Millard, Joe Hildebrand, Craig Kaes, Alexey Shchepin, David Waite, Jean-Louis Seguneau, Jacek Konieczny, Gaston Dombiak, 以及其他在 jdev@conference.jabber.org 会议室和在 Standards 邮件列表里的人.