

Hyperic HQ 用户手册（节选）

（版本：Hyperic HQ4.4）
北京铸锐数码科技有限公司

2010 年 10 月

本用户手册描述了 Hyperic HQ 提供的每一个功能模块，如果您还没有开始使用 Hyperic，请考虑先阅读产品的概览，标记为*的功能只在 HQ 企业版才提供。

版权说明：

版权所有：北京铸锐数码科技有限公司。<http://www.innovatedigital.com>

目录

- 1 HQ 资源和资源库管理**
 - 1.1 资源库模型
 - 1.2 自动发现并添加资源到资源库
 - 1.3 使用应用管理基础设施
- 2 管理 HQ 监控设置**
 - 2.1 监控总览
 - 2.2 查看更改默认收集度量
 - 2.3 管理度量数据
 - 2.4 日志跟踪
 - 2.5 配置跟踪
 - 2.6 基线*
 - 2.7 NAGIOS 数据
- 3 使用 HQ 管理选项**
 - 3.1 HQ 用户和角色
 - 3.2 HQ 服务器配置
- 4 警报指南**
 - 4.1 HQ 警报的关键因素
 - 4.2 控制警报和通知量
 - 4.3 建立配置通知的方式
 - 4.4 创建管理警报定义
 - 4.5 启用和禁用警报定义
 - 4.6 查看警报及更新警报状态
- 5 监控、分析和报表**
 - 5.1 为度量数据生成图表
 - 5.2 警报中心
 - 5.3 事件中心
 - 5.4 报表中心
- 6 使用资源的控制操作**
 - 6.1 控制总览

1 HQ 资源和资源库管理

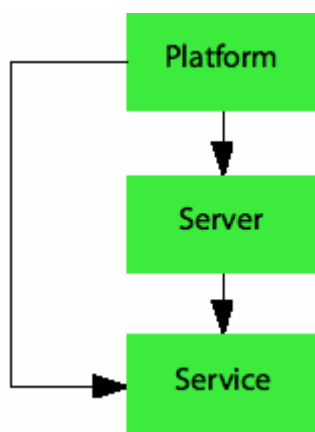
1.1 资源库模型

HQ 通常用于大的环境：它的核心价值是能够自动并且轻松的管理和控制数百台机器的上千软件资源。HQ 管理的资源库资源包括：操作系统、应用服务器、应用组件以及其他软件组件。HQ 管理复杂性的能力是基于其自身资源库和访问模式。该模型将软件资源装化为基本类型，并允许对资源分类进行访问控制和监控。

接下来将介绍资源库的描述和访问模式，如何使用 HQ 软件的资源库以及管理用户的访问和控制功能。

- I 平台、服务器和服务
 - I 组和应用
 - 应用
 - 兼容组
 - 自动化组
 - 混合组
 - I 访问控制类型：用户和角色
 - 一个角色具有特定类型的权限
 - 一个角色访问特定组别的项目
 - 一个角色必须指定到一个组
 - 用户需分配到角色

1.1.1 平台、服务器、服务



n 平台

一个平台是指运行在一台物理机器上的操作系统。HQ 使用 HQ 系统度量插件发现平台，支持的平台类型有：Linux 和 Win32 等。

n 服务器

服务器是一个运行在系统平台上的软件产品。服务器提供一个通信接口，并执行特定的任务响应请求。HQ 使用资源插件发现服务器。HQ 提供了很多服务器类型的资源插件。例如：“Apache httpd”、“Tomcat5.5”、“Exchange 2003”。

n 服务

服务是一个专注于特殊任务的单独的软件模块，运行在服务器或平台。资源插件在发现一个平台或者服务器的时候也会发现在其运行上的服务。运行在一个服务器上的服务可以是服务器的内部组件（例如：“Weblogic Admin 9.2 Entity EJB service”）或者其它项目（“CustomerEntityEJB”）。服务也可以运行在平台上。如：“DNS”或“POP3”。

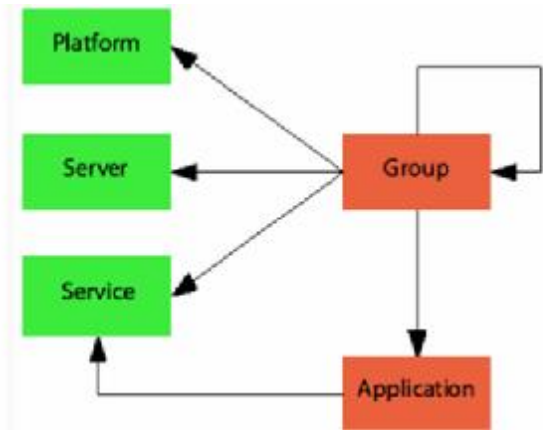
注意：内部和部署服务的区别与监控、警报策略、访问控制策略有关。详情请参考：[内部与部署服务章节](#)。

下面列出了 HQ 自动发现和监控的平台、服务器、服务列表。

Platforms	Servers	Services
<ul style="list-style-type: none">• AIX• Linux• Solaris• Win32• Cisco IOS• Xen Host• VMware V13 Host• ...	<ul style="list-style-type: none">• Apache httpd• Tomcat 5.5• Exchange 2007• MQ Series 5.x• .NET 2.0• WebLogic Admin 10• NTP 4.x• ...	<ul style="list-style-type: none">• TCP Socket• JBoss 4.0 JCA Connection Pool• IIS 6.x Vhost• Cisco IOS Interface• Zen VM NIC• Tomcat 5.5 Webapp• Sun JVM 1.5 Memory Pool•

1.1.2组和应用

HQ 模型包括两种资源库类型，分别是：应用和组。一个组可以是同种性质或不同性质的资源库资源。组包含平台、服务器、服务、应用或其他组。



HQ 自动配置了组。管理员也可以自定义组。

建立资源库项目组应用有以下好处：

- 管理复杂的环境：管理具有上千资源的复杂环境，在一个组级别查看可用性和性能

数据，可降低用户界面的混乱。

- 自动化：你可以用一个命令去自动执行组的管理操作。
- 控制访问：在组级别设置访问权限，简化了实施，访问控制的管理。
- 提供有意义的监控视图：组和应用可根据特定的用户或角色提供视图，其目的是为了满足不同业务的需求。

n 应用

在 HQ 中，一个应用是服务于一个特定功能的一组服务的集合。一个应用通常由它所提供的功能识别，例如一个购物订单入口的应用或一个在线银行的应用。一个应用可以跨多个物理资源的例子是，一个 Apache 虚拟机的后端是一个使用 Oracle 数据库的 Weblogic 应用服务器。

n 兼容组

兼容组是指具有相同平台类型、服务器类型、服务类型的资源库集合。例如：一个兼容组可能只包含“Tomcat5.5”的服务器。

兼容组的每一个成员具有相同的测量和控制文件：HQ 收集相同的度量和提供相同的控制选项。

兼容组的优点：

- 对具有相同平台类型、服务器类型或服务类型的资源库项目进行单独或整体监控。
- 定义应用：如果你定义一个应用作为一个组，这样你就可以无需修改应用的定义去添加或删除资源。
- 可对兼容组执行相同的管理操作，不必再单独的去管理操作。

n 自动化组

在资源库中，HQ 会把相关特定主机上相同的资源类型自动创建为兼容组。例如：如果在监控的平台上有五个“WebLogic9.2”服务器，HQ 会创建一个包含这五个服务器的组。

自动化组具有与平台类型、服务器类型或服务类型相同的名称。因此自动化组的名称是唯一的。在 HQ 中，自动化组只能出现在资源库的层次范围之内。例如，在监控的特定平台上，HQ 创建了自动化组。如果你浏览所有组，自动化组不会在组列表中显示，只有明确设定才会出现。

n 混合组

混合组包含不同类型的资源库资源。HQ 不会自动创建混合组，必须明确定义混合组。

混合组在执行访问控制策略时很有用。例如：一组来自同一的供应商或特定的客户托管。混合组没有共同的测量和控制文件。混合组可监控不同类型资源的度量变化。因此，混合组不支持整个组的管理操作

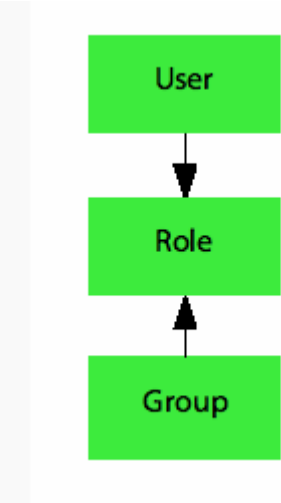
混合组有三个基本的子类型，他们都具有不同组成员的混合组。当你浏览 HQ 资源界面时，这些混合组的子类别将显示在“组”类别的模块。

- 混合组-平台、服务器、服务类型：如果您的客户有这样的业务需求，你可以使用这种混合组配置，包含客户 A 托管的所有资源，并以客户 A 命名为该混合组的名称。客户 A 混合组可以包含 Linux 平台，Tomcat 服务器以及 EJB 和 servlet 服务等。
- 混合组-组类型：这类的混合组是由其它组组成的“超级组”。例如：区域经理可能会使用这类混合组，它包含许多客户特定的组（比如上面提到的客户 A 混合组），从一个区域视角去监控可用性和度量。

- 混合组-应用类型：这类混合组是由多个应用组成。例如：业务经理可以通过混合组去监控和评估产品线的水平。

n 访问控制类型：用户和角色

在 HQ 中，你可以控制用户访问资源库，执行相关操作。访问类型和模式通过用户和角色实现。



n 角色对特定的类型有特定的权限

角色定义了一组权限和这些权限被授予的对象类型。例如“SysAdmin”角色，这个角色能查看，创建，修改和删除服务器和服务。(如图)

Resource Type	View	Create	Modify	Delete	Alerting	Control
Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Roles	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Groups	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Escalations		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Platforms	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Servers	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Services	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applications	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

定义一个角色的权限应注意以下几点：

- 资源拥有者的权限：如果你创建一个资源，你就是拥有者。你拥有该资源的所有权限。
- 权限不能够继承：一个资源和该资源下的分支资源的权限是不相同的。例如：Sarah 修改了平台 A 的权限，并不意味着她也改变了该平台下服务器 B 的权限。在平台 A 上添加一个服务器，Sarah 必须修改平台 A 具有添加服务器的权限。同样，如果 Sarah 已经删除了平台的权限，她不能删除带有子关系的平台，除非她具有删除服务器的

权限。

- 不能创建自动发现平台的权限，你也不能创建一个 HQ 可以自动发现平台。这些平台类型包括 Linux, Windows, Solaris, HP-UX, Mac OS X, FreeBSD, 和 AIX.

n 角色在特定组的访问条目

在实际资源库中，一个角色仅限于访问分配给该角色的组。例如：在下表中每一行定义了特定角色访问的组。上面的例子我们知道 SysAdmin 在 CommerceGroup 和 BackendGroup 具有查看、创建、修改和删除服务器和服务的权限。

	CommerceGroup	BackendGroup	PartnerGroup
SysAdmin	x	x	
HQAdmin	x	x	
VPNAdmin			x
OpsOwner	x	x	x
RetailOwner	x		
RelManagers			x

由上图可知：CommerceGroup 和 BackendGroup 可能包含平台，SysAdmin 角色能访问 CommerceGroup 和 BackendGroup，但只是对服务器和服务运行访问，因此该用户角色不具有访问组中平台的权限。

n 角色必须指定给一个组

在由多个组组成的混合组中，指定给父类组的角色，不能指定给子组。一个角色必须明确指定到具体的组。

n 指定用户角色

用户的访问和权限取决于他被指定的角色。下图中：Tom 和 Jane 被指定到 SysAdmin 角色。因此，Tom 和 Jane 具有在 CommerceGroup 和 BackendGroup 中查看、创建、修改和删除服务器和服务的权限。

	Tom	Jane	Uma	George	Mel
SysAdmin	x	x			
HQAdmin	x				
Compliance		x	x		
Owner				x	
SecondLine					x

未完

其余章节可联系 010-62139280 铸锐数码 <http://www.innovatedigital.com>

相关参考: <http://www.innovatedigital.com/node/539>