



# **JBoss 企业级应用程序平台 6.3**

## **6.3.0 发行注记**

适用于 Red Hat JBoss 企业版应用程序平台 6.3



# JBoss 企业级应用程序平台 6.3 6.3.0 发行注记

---

适用于 Red Hat JBoss 企业版应用程序平台 6.3

Red Hat 客户内容服务 JBoss EAP 团队

## 法律通告

Copyright © 2014 Red Hat, Inc..

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 摘要

这些注记包含和 Red Hat 企业级应用程序平台 6.3 相关的重要信息。在安装产品前请仔细阅读这些发行注记。

---

## 目录

1. 概述 .....	2
2. 常见问题 .....	2
3. 新的功能 .....	4
4. 仅作为技术预览提供的功能 .....	6
5. 不受支持的功能 .....	7
使用 STSClientPoolFactory .....	8
6. 组件 .....	9
7. 这个版本中所做的修改 .....	9
7.1. 程序错误修复 .....	9
7.2. 增强 .....	43
7.3. 已知的问题 .....	46
8. PICKETLINK 2.1 和 2.5 版本间的 API 修改 .....	65
9. JBOSSWS-SPI 和 JBOSSWS-CXF 里的修改 .....	66
A. 修订 .....	68

## 1. 概述

Red Hat JBoss 企业级应用程序平台 6 (JBoss EAP 6) 是 Red Hat 对目前机构开发和部署企业级应用程序的方式的重大变化的回应。由于机构寻求更低的运营成本并缩短新应用程序的上市时间，我们重构了 JBoss EAP 6，使其拥有创新的模块化、对云架构的支持、强大的管理和自动化和世界级的开发人员生产力。

JBoss EAP 6 是 Java EE 6 认证的功能强大的平台，它可以灵活地管理，也拥有改进的性能和扩充性、以及提高开发人员生产力的许多新的功能。有着 Red Hat 的市场领先的认证和支持的信誉，它可以确保您的管理和发展的需要并继续面向未来和超越。

[提交 bug 报告](#)

## 2. 常见问题

问：这个版本有哪些改动？

答：JBoss EAP 6.3.0 包含了许多改进和修复。具体细节请参考《*本版本里的修改*》章节。

---

问：完整的文档套件在哪里？

答：完整的 JBoss EAP 6.3.0 文档套件位于 [https://access.redhat.com/documentation/en-US/JBoss\\_Enterprise\\_Application\\_Platform/](https://access.redhat.com/documentation/en-US/JBoss_Enterprise_Application_Platform/)。

---

问：升级说明在哪里？

答：升级说明可在《*安装指南*》里找到。

---

问：这个产品支持哪些操作系统、Java 虚拟机和数据库服务器？

答：关于已经针对 JBoss EAP 6.3.0 进行测试和验证的操作系统、Java 虚拟机、数据库服务器和 JDBC 驱动组合的完整列表，请访问：<https://access.redhat.com/site/articles/111663>。

---

问：产品环境支持附带的 H2 数据库吗？

答：否。附带的 H2 数据库仅可用于评估、测试和演示目的。产品环境不支持它。更多信息请参考 <https://access.redhat.com/site/solutions/148633>。

---

问：JBoss EAP 6.3.0 支持哪些工业标准？

答：关于被支持的规格和标准的完整列表，请参考 <https://access.redhat.com/site/articles/113373>。

---

问：移植到这个版本我可能遇到哪些问题？

答：请参考《*本版本里的修改*》章节来了解本版本和之前版本的区别以及移植时可能导致的问题。

---

问：本版本包含了哪些组件，它们的版本是什么？

答：关于包含的组件的完整列表，请参考 <https://access.redhat.com/site/articles/112673>。

---

问：本版本里包含了哪些技术预览？

答：JBoss EAP 6.3.0 包含了大量的技术预览功能。这些功能是不享有支持服务的，且可能功能不完备，不适合用在产品环境里。这些功能为客户提供了对产品创新的早期介入机会，让他们可以测试功能并在开发阶段提供反馈。

关于本版本里的技术预览的完整列表，请参考 《*仅作为技术预览提供的功能*》章节。

---

问：在哪可以找到关于我的支持合约的更多细节？

答：支持策略的细节位于下列 URL：

#### 支持过程

[https://access.redhat.com/site/support/policy/support\\_process](https://access.redhat.com/site/support/policy/support_process)

#### 产品支持的覆盖范围

<https://access.redhat.com/site/support/offerings/production/soc>

#### 产品支持服务级别协议

<https://access.redhat.com/site/support/offerings/production/sla>

#### 开发人员支持的覆盖范围

<https://access.redhat.com/site/support/offerings/developer/soc/>

#### 开发人员支持服务级别协议

<https://access.redhat.com/site/support/offerings/developer/sla/>

#### 产品更新和支持策略

[https://access.redhat.com/site/support/policy/updates/jboss\\_notes/](https://access.redhat.com/site/support/policy/updates/jboss_notes/)

#### JBoss 最终用户许可证协议

[http://www.redhat.com/licenses/jboss\\_eula.html](http://www.redhat.com/licenses/jboss_eula.html)

---

问：我在文档里发现了错误，我怎样报告？

答：要为此文档提供反馈，请在 <https://bugzilla.redhat.com> 报告程序错误并指定产品：**JBoss Enterprise Application Platform 6**，版本：**6.3.0** 和组件：**Documentation**。

下列 URL 会自动填充产品、版本和组件字段：[https://bugzilla.redhat.com/enter\\_bug.cgi?component=Documentation&product=JBoss%20Enterprise%20Application%20Platform%206&version=6.3.0](https://bugzilla.redhat.com/enter_bug.cgi?component=Documentation&product=JBoss%20Enterprise%20Application%20Platform%206&version=6.3.0)

---

---

---

[提交 bug 报告](#)

## 3. 新的功能

JBoss EAP 6.3 添加了下列新的功能。

### 对 PicketLink 的增强

JBoss EAP 的 PicketLink 组件的主要修改是：

- 用于 IDP 和 Federation 的管理子系统
- 安全配置的 CDI 注入
- IDP 基于证书的登录
- IDP 基于 Kerberos 的授权
- IDP 的专有 SSO 功能
- 服务提供者的动态帐号选择器
- 自定义 `picketlink.xml` 的路径。

### 对域恢复的改进

现在 JBoss EAP 6.3 的主机控制器可以配置备份 IP 地址以用于 JBoss EAP 域控制器。这个功能让管理员可以配置主机控制器自动失效切换到备份域控制器，从而确保管理域的可用性。然而，在失效切换发生之前，管理员将需要手动促使备份主机控制器充当新的域控制器。

### 对 PKCS11 密钥库的支持

除了以前版本里的 Java 密钥库以外，我们增强了 JBoss EAP 安全域和 JMS 子系统 (HornetQ) 来支持 PKCS11 密钥库的授权。

### 补丁功能在 Web 管理控制台里可用

JBoss EAP 6.2 里引入并通过管理 API 开放的补丁功能，在 JBoss EAP 6.3 里的 Web 管理控制台里也可以使用了。现在可以安装和回滚补丁、查看补丁状态和历史。

### Web 管理控制台里新的主页

JBoss EAP 的 Web 管理控制台的主页现在提供到最常用的管理功能的链接，使得通过控制台导航更加容易和迅捷。

### 在 Web 管理控制台里测试数据源

Web 管理控制台现在开放了测试数据源的能力，给予管理员正确的访问权限以确保在创建后或部署应用程序前数据源处于连接状态。

### 顶层命名的改动

我们统一了独立模式和域模式下的顶层导航标签。

新的标签是：

#### Configuration

这个标签页显示了和配置集相关的持久化配置。

#### Runtime



这个标签页显示了服务器的 runtime 信息。

### Administration

这个标签页显示了访问控制设置。

### Domain

这个标签页显示了域配置（仅用于域模式）。

## 分析数据的收集

Web 管理控制台现在有能力将使用情况报告给 Red Hat。在将来的 JBoss EAP 版本里，Red Hat 将使用这些数据来增强可用性。这个功能默认是关闭的，但可以在控制台设置里启用。Red Hat 鼓励您启用分析数据的收集。

## 部署覆盖的增强

部署覆盖（Deployment Overlay）提供了替换已部署应用程序里的部署描述符的能力，我们增强这个功能来覆盖部署的二进制文件。这个增强扩展了系统管理员修改应用程序的行为而无需修改应用程序归档的能力。

## 对 Microsoft Windows Server 2012 R2 的支持

我们已经测试了 Microsoft Windows Server 2012 R2，并将其添加到了受支持的配置里。

## 对 Red Hat Enterprise Linux 7 的支持

我们已对 Red Hat Enterprise Linux 7 进行测试并添加到了受支持的配置里。

## 对装有 Red Hat Enterprise Linux 的 PowerPC 架构的支持

JBoss EAP 6.3.0 已在 PPC64 架构上的 Red Hat Enterprise Linux 6 和 7 里进行了测试，目前两种配置都是受支持的。

## WSI-Basic Security 1.1

我们测试了 JBoss Web Services 以确保与 WSI-Basic Security 1.1 配置集文件的兼容。

## Hibernate

改进和扩展了字节码。

## WS-Atomic Transactions/XA Bridge

WS-AT/XA 事务桥功能已从技术预览转成完全支持状态。这个桥为开发人员提供了 WebService 和 Java EE 事务域间的双向链接。

## 恢复 ServiceMBean Helper 类

我们引入了 `org.jboss.as.system-jmx` 模块来恢复之前版本里协助用户创建 MBean 的 Helper 类。

## EJB 和 MDB 属性替换

JBoss EAP 现在允许用 `@ActivationConfigProperty` 和 `@Resource` 注解在 EJB 和 MDB 里启用属性替换（property substitution）。

## 垃圾收集日志

在 **standalone** 模式里默认是启用垃圾收集日志（Garbage Collection Logging）的。我们改进了协助诊断过程，有需要时才登记日志。而不是先启用垃圾收集日志，然后才等待问题的再次发生。

### Web 部署里的注解

已注解的 Servlet 组件可以在共享模块里定义，然后由 Servlet 容器选择并应用到对该模块有依赖关系的所有 Web 部署上。这降低了在 **web.xml** 文件里定义所有 Servlet 依赖关系的需要。例如，您可以添加注解的 Web 过滤器或 Listener 为共享模块且由所有 Web 应用程序自动选择。

### 日志

现在可以配置 JBoss EAP，使日志模块默认不会添加到部署里。

### 事务管理者的增强

除了对 Last Resource Commit Optimization (LRCO) 的支持，现在 Commit Markable Resource (CMR) 事务优化也是可用的了。

### 为 CORBA 延迟配置 IOR 设置

现在可以为 CORBA 延迟配置 IOR 设置了。

### Java EE

Java EE 子系统现在支持在全局模块 XML 配置里使用 **annotation** 标志。将其设置为 **true** 时，全局模块将导出它的注解索引到服务器上所有的 Java EE 部署里。

### 配置 mod\_cluster sessionDrainingStrategy

本版本的 JBoss EAP 可以配置 session draining strategy (session-draining-strategy attribute) 了。虽然 mod\_cluster 自身支持这个配置，现在也不能配置了。它已开放给用户。

### [提交 bug 报告](#)

## 4. 仅作为技术预览提供的功能

下面的配置和功能只是作为技术预览提供，产品环境是不支持的。

### Stateless Identity Bean

JBoss EAP 6.3 在 PicketLink 里包含了一个新的标识符管理功能，它提供使用 stateless 以及 session-scoped identity bean 的能力。

### WS-Trust/STS 和 JBoss Web Service

JBoss Web Service 现在开放了底层的 CXF 实现的 WS-Trust/STS 功能。

### 用 JBoss CLI 添加和删除模块

CLI 提供了添加和删除模块的新命令。

### 用 Hibernate Validator 进行 RestEasy 检验

RestEasy 现在包含了一个检验提供者来支持 JBoss EAP 6 附带的 Hibernate Validator。

### Multi-JSF

这个功能让用户可以用自己的 JSF 实现替换 JBoss EAP 6 提供的 JSF 实现。

## mod\_jk 和 IPv6

mod\_jk 版本已经从 1.2.36 升级到 1.2.40。这个新的版本包含对 IPv6 的支持，但是这个功能还未经过完备的测试。

## WebSockets

WebSocket 协议提供 Web 客户端和服务端间的两种通讯方式。客户端和服务端间的通讯是基于事件的，和轮询方法相比，它有更快的处理速度和更小的带宽。

[提交 bug 报告](#)

## 5. 不受支持的功能

JBoss EAP 目前不支持下列功能。

### RHEL 7 上 Apache 的 mod\_jk 和 mod\_cluster

RHEL 7 上可通过 RHN 频道安装的 Apache HTTP 服务器带来了和 **mod\_cluster** 及 **mod\_jk** 不兼容的问题，且没有支持服务。而 JBoss EAP 附带的 Apache HTTP 服务器却享有所有的支持服务。

### mod\_rt 和 mod\_snmp

JBoss EAP 6 的 Apache HTTP 服务器附带的 mod\_rt (**mod\_rt.so**) 和 mod\_snmp (**snmpmonagt.so**) 模块不享有支持服务。

您可以在 Enterprise Web Server 2.1 的文档里找到关于对这些模块的支持的更多信息。

### STOMP 协议和 HornetQ

HornetQ 对 STOMP 协议提供社区级别的支持。这个协议没有接受 Red Hat 的测试，所以不享有 JBoss EAP 的支持服务。

### REST 协议和 HornetQ

HornetQ 对 REST 协议提供社区级别的支持。这个协议没有接受 Red Hat 的测试，所以不享有 JBoss EAP 的支持服务。

### Infinispan API

JBoss EAP 6 里不支持直接使用 Infinispan API。Infinispan 被用作 JBoss EAP 6 内部不同群集技术的实现细节。直接使用 Infinispan API 需要订阅 JBoss Data Grid。

### JDK 6 的 IPv6 限制

下面的 IPv6 限制是 JDK 6 造成的，并非 JBoss EAP 6 的问题。

- 在 Microsoft Windows 服务器上，JDK 6 只有一部分的 IPv6 实现。这个实现不足以运行 JBoss EAP 6。Microsoft Windows 服务器上对 IPv6 的完整支持要求 JDK 7。
- 在 RHEL 上，Oracle JDK 6 里的一个程序错误导致在包含 zone-id 的客户（建立连接的网络端点）上指定的任何地址都会失败。要使用 zone-id，要么升级到 JDK 7，要么使用 RHEL 里的 IcedTea/OpenJDK 6，且不要展示这个错误。关于这个错误的更多信息，请参考 [http://bugs.sun.com/bugdatabase/view\\_bug.do?bug\\_id=6800096](http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6800096) 和 <https://issues.jboss.org/browse/JBPAPP-8833>。

### JTA 事务外部的 JPA 2.0 内容填充

扩展持久性上下文（Extended Persistence Contexts, XPC）的填充没有考虑事务的存在，所以 XPC 总是被填充。这种行为不符合 JPA 2.0 规格。我们修改了 XPC 的处理，当没有活动事务时，XPC 的填充将被忽略且被调用的 Bean 将有自己的持久性内容而不是 XPC。

如果您的应用程序希望在 JTA 事务外填充 XPC，您需要考虑是否修改应用程序。关于如何修改应用程序的说明，请参考《JBoss EAP 6 移植指南》。

JBoss EAP 5 提供了一个系统属性（**JBPAPP-923.alwaysPropagate**）来启用这个行为。这个系统属性在 JBoss EAP 6 里是不可用的。

关于这个决定的更多信息，请参考 <https://issues.jboss.org/browse/AS7-1663>。

## PicketLink 子系统

JBoss EAP 6.3 不支持 PicketLink 子系统。

## STS 客户池

PicketLink 在服务器上提供一个 STS 客户池。这消除了创建 STS 客户的瓶颈。

需要 STS 客户来获取 SAML 票据的登录模块可以利用客户池。

可以利用 STS 客户池的登录模块：

- `org.picketlink.identity.federation.core.wstrust.auth.STSIssuingLoginModule`
- `org.picketlink.identity.federation.core.wstrust.auth.STSValidatingLoginModule`
- `org.picketlink.trust.jbossws.jaas.JBWSTokenIssuingLoginModule`

通过 **initialNumberOfClients** 选项可以配置客户池里每个登录模块的默认客户数量。

STSCientPoolFactory 类

**org.picketlink.identity.federation.bindings.stspool.STSCientPoolFactory** 为应用程序提供了客户池功能。

## 使用 STSCientPoolFactory

STS 客户会插入子池且以它们的 **configuration** 为键。获取 STSCientPool 实例并根据配置初始化子池，可以指定初始的 STS 客户或使用默认的数量。

```
final STSCientPool pool = STSCientPoolFactory.getPoolInstance();
pool.createPool(20, stsClientConfig);
final STSCient client = pool.getClient(stsClientConfig);
```

当您使用完客户后，您可以将其退回池里：

```
pool.returnClient();
```

根据给定的配置检查子池是否已经存在：

```
if (! pool.configExists(stsClientConfig) {
    pool.createPool(stsClientConfig);
}
```

启用 PicketLink Federation 子系统时，为某个部署创建的所有客户池都会在卸载过程中自动销毁。如要手动销毁池，可用：

```
pool.destroyPool(stsClientConfig);
```

### Tanuki Service Wrapper

JBoss EAP 6 不支持 Tanuki Service Wrapper。更多信息请参考 <https://issues.jboss.org/browse/JBPAPP-8651>。

### Microsoft SQL Server 2008 上的 XA 恢复

Microsoft SQL Server 2008 R1 里无法使用 XA 恢复。这个功能可以用于 Microsoft SQL Server 2008 R2 SP2。相关详情请参考：

- <https://issues.jboss.org/browse/JBPAPP-8983>
- [https://bugzilla.redhat.com/show\\_bug.cgi?id=901254](https://bugzilla.redhat.com/show_bug.cgi?id=901254)
- <https://community.jboss.org/thread/145358>

### MySQL 上的 XA 恢复

MySQL JDBC 驱动没有正确实现 XA 恢复，当和数据库的连接停止时它会返回不正确的错误代码。

更多信息请参考 <https://issues.jboss.org/browse/JBPAPP-2576> 和 <http://bugs.mysql.com/bug.php?id=72890>。

### JBoss OSGi

JBoss OSGi 是一个 OSGi 规格的实现，它已从技术预览降为 Unsupported。其他信息请参考 <https://access.redhat.com/site/solutions/362814>。

### Quickstarts

因为 OSGi 从未被支持过且已从产品里删除，所以我们也从 JBoss EAP 6.3 附带的例程里删除了 **helloworld-osgi** quickstart。

**wicket-war** 和 **wicket-ear** quickstarts 依赖 **seam-conversation-weld.jar**。因为 Seam 仍不享有支持服务，所以这些 Quickstart 例程也从 JBoss EAP 6.3 附带的例程里删除了。

[提交 bug 报告](#)

## 6. 组件

在这里您可以找到 JBoss EAP 6 组件矩阵：<https://access.redhat.com/knowledge/articles/112673>。

[提交 bug 报告](#)

## 7. 这个版本中所做的修改

### 7.1. 程序错误修复

#### CDI/Weld

**1051375** - 如果 EAR 里部署的 WAR 中存在多个 beans.xml 文件，所有的 CDI bean 都会注册两次。

在以前的 EAP 6 版本里，当 EAR 部署包含一个 WAR 子部署以及多个 beans.xml 文件时，如 WEB-INF/beans.xml 和 WEB-INF/classes/META-INF/beans.xml，所有的 Bean 都会注册两次且部署会失败。我们通过修改部署处理器解决了这个问题，这样的部署现在不会再失败了。

### 1029099 - 当复制 EAP 6.1.0 群集上的 CDI bean 时抛出 NPE

早期版本的 JBoss EAP 的群集实例里创建的 Session ID 包含

新引入的代码所包含了相对路径而不是绝对路径。这意味着 bean 不再会重新创建且不会再出现 NPE 了。

### 1050963 - 永久性地修复了: org.jboss.weld.exceptions.DeploymentException: WELD-001414 Bean name is ambiguous

具有多个子部署的部署里没有隔离 Bean 名称的模糊性检验，再某些场景里这会导致 **DeploymentException**。本版本的 JBoss EAP 6 通过升级 Weld 组件解决了这个问题。

### 1070069 - 过期的会话意外地出现在下一个请求里

在以前的 JBoss EAP 6 版本里，我们发现会话可能会意外地被激活，且及时在会话过期后仍会和请求关联，从而导致 **NonExistentConversationException**。

这是因为在 JSF 应用程序里，在请求的开始 Weld 没有正确检查会话状态。

这个版本包含了升级的会话上下文激活和失效过程以更详尽地检查会话状态。因此，过期的会话不再错误地和请求关联了。

## CLI

### 988283 - CLI GUI - 字符串值的 write attribute 对话框应该用双引号将生成的命令值括起来

试图用 **jboss-cli** 工具设置某个包含属性的值时只会保存字符 **\$** 而不是属性，除非整个值用双引号括起。这是因为 CLI 命令行错误地解析了任何 **#{X}** 表达式为 **\$**，除非用双引号括起来。

本版本里修复了命令行解析器，所以带有属性的值将被正确解析，即使它们没有用双引号括起来。

### 1007831 - CLI GUI - webservices client-config 和 endpoint-config 的 add 操作失败

在以前的 JBoss EAP 6 版本里，没有任何参数的 **add** 操作会被视作不要求 CLI GUI 进行任何输入的操作。

这导致用户无需提供所需的参数 **name** 就可以执行操作，且会失败。

在这个版本里，不带参数的 **add** 操作将被 GUI 界面逻辑按要求 **name** 参数的操作来处理。

现在，在允许用户执行在管理模型描述里不要求任何参数的 **add** 操作之前，会提示用户提供所需的 **name** 参数。

### 1019232 - 在数据源节点上使用 tab complete 时 jboss-cli.sh 抛出 NullPointerException

在以前的 JBoss EAP 6 版本里，在某些情况下当用户试图使用 tab complete 功能时，**jboss-cli** 工具将抛出异常并退出。

这是因为命令解析器里对这种情况没有足够的异常处理，本版本已纠正了这个问题。

**1031173 - 当 EAP 安装到目录名带有空格的目录里时, jboss-cli.bat 无法运行。**

在 Windows 上, 如果 JBoss EAP 6 安装到目录名带有空格或其他特殊字符 (如括号) 的目录里, `jboss-cli` 工具将无法成功启动。例如, 如果 JBoss EAP 6 安装在 `C:\JBoss EAP\jboss-eap-6.2` 里, 下列错误信息将会显示:

```
Error: Could not find or load main class EAP\jboss-eap-6.2\bin\jboss-
cli-logging.properties
```

通过修改 `jboss-cli` 工具在 Windows 上声明和使用文件路径的方式, 我们修复了这个问题。现在, 对于带有空格或其他特殊字符 (如括号) 的目录, 这个工具无需任何变通办法就可以正常使用了。

**1026418 - 接受 SSL 证书时作为参数传入的命令没有被执行**

当用作为参数传入的命令启动 JBoss EAP 6 CLI 时, 如果服务器提示用户接受服务器证书, 它会将此作为错误登记日志。这导致作为参数传入的任何命令都被忽略, 因为这些命令只有在没发生错误时才会被执行。

我们将接受证书提示作为普通输出而不是错误处理, 从而解决了这个问题。因此, 在用户已经接受服务器的证书后, 用作为参数的命令可以成功启动 CLI。

**CLI, Remoting****1037574 - 运行多个 CLI 操作时遇到的 OOM 可能是内存清理不正确引起的**

以前的 JBoss EAP 6 的用户在通过命令行接口执行多个操作时可能遇到 `OutOfMemory` 错误。

这个问题可以追溯到使用 CLI 时内存清理错误引起的内存溢出。

这个版本已经解决了这个问题。

**类加载****1060997 - testConnection 应考虑部署的类加载器**

以前的 JBoss EAP 6 版本在测试数据源时抛出异常是由于一个程序错误引起的。当在数据源 `"connection-url"` 标签里使用 LDAP 协议时, 这个异常会被抛出。产品将无法实例化 `InitialContextFactory`, 因为 `"org.jboss.as.connector"` 模块不能访问某些 JDK 提供的类 (如 `com.sun.jndi.ldap.LdapCtxFactory`)。这导致数据源 `"connection-url"` 使用 `"ldap://"` 协议时连接测试失败。

我们在 `org.jboss.as.connector` 模块里添加了对 `sun.jdk` 的依赖关系, 从而解决了这个问题。这使得连接器模块可以访问所需的 JDK 类, 使用 `"test-connection-in-pool"` CLI 操作进行的数据源测试可以成功完成了。

**1054972 - MBean 的初始化使用了错误的 TCCL**

在以前的 JBoss EAP 版本里, 我们发现当初始化 SAR 文件里找到的 MBean 时, 应用程序的 TCCL (线程上下文类加载器) 没有合适地设置。这意味着这个部署里的资源在初始化时对于 MBean 不可用。在这个版本里, TCCL 根据对 MBean 初始化的调用合适地进行了设置, MBean 在初始化时现在可以访问部署资源了。

**971076 - "org.jboss.log4j.logmanager" 模块需要对 "javax.mail.api" 模块的依赖关系**

在以前的 JBoss EAP 6 版本里，`org.jboss.log4j.logmanager` 没有在其 `module.xml` 里指定对 `javax.mail.api` 的依赖关系。

这个版本里添加了对 `logmanager` 模块的依赖关系。

## 群集

### 990567 - 保存 HTTP 会话到 PostgreSQL 时出现 `ClassCastException`

我们发现了一个影响 HTTP 会话存储到 Postgresql 的问题。使用下列 `jdbc-store` 配置时会抛出 `ClassCastException`，就像这里显示的错误信息示例。HTTP 会话的值将被持久化但应用程序重部署会失败且显示相同的错误信息。

```
<binary-keyed-jdbc-store datasource="java:jboss/datasources/testDS"
preload="true" passivation="false" purge="false">
  <binary-keyed-table prefix="b">
    <id-column name="id" type="VARCHAR(255)"/>
    <data-column name="datum" type="BYTEA"/>
    <timestamp-column name="ver" type="BIGINT"/>
  </binary-keyed-table>
</binary-keyed-jdbc-store>
```

```
14:24:21,262 ERROR
[org.infinispan.interceptors.InvocationContextInterceptor] (http-
/127.0.0.1:8080-1) ISPN000136: Execution error:
java.lang.ClassCastException: java.lang.Class cannot be cast to
org.infinispan.loaders.bucket.Bucket...
```

通过升级组件，我们解决了这个问题，现在 HTTP 会话数据可以在 Postgresql 数据库里成功存储了。

### 917010 - `CacheException: Failure while unregistering mbeans at server shutdown`

以前的 JBoss EAP 版本里的 Infinispan 包含了一个程序错误，它在服务器关闭时导致下列异常：

```
WARN [org.infinispan.jmx.CacheJmxRegistration] (MSC service thread 1-1)
ISPN000032: Problems un-registering MBeans:
org.infinispan.CacheException: Failure while unregistering mbeans
```

这个错误不会阻止服务器的关闭，它是因为多个 `stop` 请求（`CacheService` 和 `EmbeddedCacheManagerService` 里的请求）被发送到单个缓存实例而引起的。

通过升级 Infinispan 组件，这个问题已得到解决。

### 963448 - `CoreGroupCommunicationService#handle` 里的异常处理不正确

在以前的 JBoss EAP 6 版本里，我们发现 `CoreGroupCommunicationService#handle` 没有正确处理异常，影响了类的调用者。

我们发现这个类隐藏了隐藏并返回 `null`。这个值可能在调用者里产生意外的行为。

在这个版本里，`null` 响应将由分布式锁管理者处理，然而以后版本里的 `CommandDispatcher` 将永久性地解决这个问题。



## 1039585 - 群集会话内存泄露

以前的 JBoss EAP 6 版本里的一个程序错误会导致分布式 Web 会话里的 `OutOfMemoryException`。如果 Web 会话过期且无需释放或销毁带有会话管理者创建的锁对象，那这个异常将会抛出。当 Web 会话继续过期，残余的锁对象将在内存里积累。最终这会导致 `OutOfMemoryException`。唯一的解决办法是重新部署 Web 应用程序。

这个版本正确地释放了锁对象，从而不会再遇到 `OutOfMemory` 异常了。

## 956904 - Infinispan remote-store 需要 remote-servers，但在管理模型里并没有标记为必需的。

在处理 Infinispan 的 `remote-store` 配置时我们发现了一个问题。`remote-store` 选项要求设置 `remote-server` 选项的值，但这并未强制执行，导致 `remote store` 出现故障。我们通过标记 `remote-server` 参数为强制选项，阻止无效的 `remote-store` 配置，从而解决了这个问题。

在这个例子里，管理 CLI 命令没有完成并显示了警告信息。

```
[standalone@localhost:9990 /] /subsystem=infinispan/cache-
container=web/distributed-cache=dist/remote-
store=REMOTE_STORE:add(remote-servers=[])    {
  "outcome" => "failed",
  "failure-description" => "JBAS014706: [0] is an invalid size for
parameter remote-servers. A minimum length of [1] is required",
  "rolled-back" => true,
  "response-headers" => {"process-state" => "reload-required"}
}
```

在这个例子里，管理 CLI 命令正常完成。

```
[standalone@localhost:9990 /] /subsystem=infinispan/cache-
container=web/distributed-cache=dist/remote-
store=REMOTE_STORE:add(remote-servers=[{"outbound-socket-binding" =>
"fred"}])
{
  "outcome" => "success",
  "response-headers" => {"process-state" => "reload-required"}
}
```

## 域管理

### 1015303 - LDAP 安全区需要可配置的超时选项

本版本的 JBoss EAP 6 进行了改进，允许对转出 LDAP 连接使用自定义属性。

在以前的版本里，转出 LDAP 连接是用有限属性集创建的，剩余部分使用默认的行为。因此不可能定义自定义属性来控制有些行为，如连接和读超时。

现在这个版本可以为转出 LDAP 连接定义自定义属性，代码类似于：

```
<ldap name="LocalLdap" url="ldap://localhost:10389" search-
dn="uid=wildfly,dc=simple,dc=wildfly,dc=org" search-
credential="password1!">
  <properties>
    <property name="one" value="two"/>
  </properties>
</ldap>
```

```
<property name="three" value="four"/>
</properties>
</ldap>
```

### 1074999 - 应用程序从 EAP 控制台的 Manage Deployments 部分消失

在以前的 JBoss EAP 6 版本里，使用控制台时在文件系统上没有更新部署的状态。

因此，使用文件系统扫描器和控制台来管理部署的状态会导致扫描器以为部署已被卸载。

这个版本器用了控制台和文件系统扫描器来共享部署的状态，用户现在也可以使用管理工具来管理部署了。

### 1035232 - EAP 域模式无法和安全管理者一起正常使用

在以前的 JBoss EAP 6 版本里，如果使用了 Java 安全管理者而没有指定其类名，受管域里的服务器将无法成功启动。

例如，我们通常是在 'domain.conf' 里指定 `-Djava.security.manager` 或作为命令行参数来使用默认的安全管理者的。

这是因为未赋值的系统属性会导致主机控制器将 `true` 传递给受管服务器，所以服务器会错误地用 `true` 的类名尝试使用 Java Security Manager。

我们通过添加对主机控制器属性的额外检查，使系统属性可以正确传递给受管服务器，从而解决了这个问题。因此，通过指定 `-Djava.security.manager`，使用安全管理者的受管域可以正常运行了。

### 1047515 - 域模式无法在 IBM JDK 里启动

本版本已解决了频繁阻止在 IBM JDK 环境里成功启动 JBoss EAP 6 的问题。

这个问题可以追溯到由父进程写入 `java.lang.Process.getOutputStream()` 的二进制数据（也就是 `byte[]`）如何被子进程通过 `System.in` 接收。高阶位为 1 的字节都会被篡改，从而在控制台日志里产生下列输出：

```
[Host Controller] 16:44:06,419 ERROR
[org.jboss.as.controller.management-operation] (management-handler-
thread - 1) JBAS014612: Operation ("start") failed - address: ([
[Host Controller] ("host" => "master"),
[Host Controller] ("server-config" => "server-one")
[Host Controller] ]): java.lang.IllegalStateException: JBAS010986: Host-
Controller is already shutdown.
[Host Controller] at
org.jboss.as.host.controller.ServerInventoryImpl.startServer(ServerInven-
toryImpl.java:175)
```

JBoss EAP 管理的域进程控制器与服务器的通讯是通过这种机制进行的，意味着这个问题会影响到 EAP 6 的受管域。

在本版本里，所有在 `stdin` 上和受管进程的通讯都是 Base64 编码，这样就改善了这个问题，Windows 里受管域服务器现在可以用 IBM JDK 正常启动了。

### 1054776 - ClientConfigurationImpl 不应该将 IP 地址解析为主机名

在以前的 JBoss EAP 6 里，当创建控制器客户的实例并将传入 IP 地址进行连接时，IP 地址会被解析成机器的主机名。

然后，当客户试图建立连接时，它使用主机名而不是 IP 地址。

这不仅会因为调用 DNS 服务器产生不必要的负荷，在某些特殊的部署场景下也会导致客户试图连接相同主机的不同 IP 地址而不是 JBoss EAP 实例绑定的地址，从而导致连接失败。

本版本已经修复了这个问题。控制器客户代码不会将给定的 IP 地址解析成主机名，而只会使用这个 IP 地址（而不是对应的主机名）进行连接。

请注意这次修复不会影响主机名传入到控制器客户的 `factory` 方法。此时，主机名将被解析成要使用的 IP 地址。

#### 1072915 - 如果启用了 RBAC，从控制器无法重新连接到重启的主控制器。

在以前的 JBoss EAP 版本里，当重新连接主主机控制器时，配置模型会被添加到错误的位置。

这意味着重新连接到启用了 RBAC 的主主机控制器会失败。

在这个版本里，这个模型被添加到了正确的位置，确保了从主机控制器可以正常进行连接。

#### 1040621 - 在域里不能使用具有相同 runtime-name 的部署

在以前的 JBoss EAP 6 版本里，启动时对服务器组里部署的重复 runtime-name 属性的检查过于激进。如果在整个域而不是单个服务器组里有任何重复的 runtime-name，系统都会错误地抛出异常。

因此，域里包含具有相同 runtime-name 的多个部署会导致引导失败，即使这些部署并没有映射到相同的服务器组里。

例如：

```
JBAS010932: Caught exception during boot:
org.jboss.as.controller.persistence.ConfigurationPersistenceException:
JBAS014676: Failed to parse configuration
...
Caused by: javax.xml.stream.XMLStreamException: ParseError at [row,col]:
[1348,9]
...
Message: JBAS014664: An element of this type named 'example.war' has
already been declared
```

在这个版本里，我们将 runtime-name 唯一性的检查从 XML 解析器移到了操作执行逻辑里，这个检查逻辑将忽略和相同服务器组以外的重复条目。

域里现在可以存在多个具有相同 runtime-name 值的部署，只要它们没有映射到同一个服务器组。

#### 1076066 - 无法提升从主机控制器为主主机控制器和不通过删除 domain.cached-remote.xml 加载。

以前的 JBoss EAP 6 版本里的一个程序错误阻止了已重新配置为主主机控制器的从主机控制器在重启时加载缓存的域配置。这是因为新提升的主主机控制器从 `domain/configuration/domain.xml` 文件而不是 `domain/configuration/domain.cached-remote.xml` 文件里加载配置。

在这个版本里，如果用 `--backup` 选项启动的从主机控制器被重新配置为主主机控制器，并重载来体现这个变化，它作为备份服务器运行时维护的 `domain.cached-remote.xml` 文件将被自动检测并使用。

### 1038465 - 当资源被删除时 remote-destination-outbound-socket-binding 服务并没有停止

在以前的 JBoss EAP 6 版本里，当资源被删除时 remote-destination-outbound-socket-binding 服务并没有停止。

这意味着无法用相同的名称添加 remote-destination-outbound-socket-binding，因为已有服务用这个名称进行了注册，:add 操作会失败。

在这个版本里，当删除 remote-destination-outbound-socket-binding 资源时，对应的服务器将被停止。

由此，现在可以正常删除和添加 remote-destination-outbound-socket-binding 服务了。

### 1110065 - 以批模式在从控制器上创建服务器并设置系统属性会失败

在以前的 JBoss EAP 版本里，当 HostController 升级其配置时，它会创建操作以应用这些修改到它管理的所有运行的服务器上。

因为主机控制器也为 **non-running** 创建操作来添加 "env" 系统属性，这会导致 "No handler for operation composite at address" 错误。

本版本不会再创建操作来修改 non-running 服务器的环境属性，这个错误不会再出现了。

### 1093866 - 没有正确实施 "admin-only-policy" 从主机控制器配置选项

在以前的 JBoss EAP 6 里，以 admin-only 模式启动的从主机控制器无法连接到主主机控制器来获取 domain-wide 配置。

由此，如果从主机控制器用 **--admin-only** 启动且其 **host.xml** 里设置了 **admin-only-policy="fetch-from-master"**，从主机控制器将无法连接到主主机控制器来获取 domain-wide 配置。它会启动失败。

这阻止了 **--admin-only** 从主机控制器使用 RBAC，除非 domain-wide 配置文件在本地保存为 **domain.cached-remote.xml** 且从主机控制器使用 **--cached-dc** 选项启动。

在这个版本里，我们纠正了从连接逻辑，从主机控制器可以连接到主主机控制器并获取 domain-wide 配置，允许它获得 domain-wide RBAC 配置策略。

### 1024109 - DelegatingServerInventory 里出现 NPE

在主机控制器启动后立即发送到主机控制器的管理请求可能失败并抛出 **NullPointerException**。这是因为主机控制器能够接收请求和服务器管理系统完全启动之间会有一段时间。这段时间通常小于 1 秒，但期间收到的任何请求都会失败。

这个版本确保主机控制器只有在服务器管理系统完全启动后才接收请求，从而修复了这个问题。因此，不会再因为服务器管理系统没有完全启动而触发 **NullPointerException** 了。

### 1085122 - ApplyRemoteMasterDomainModelHandler 应该应用根资源的值

在以前的 JBoss EAP 版本里，我们发现域控制器的根域资源没有应用在从主机控制器上。相反，从主机控制器将使用它们自己的根资源。

这意味着当直接连接到从主机控制器时，根域资源里的版本号码就是从域控制器的版本号码，所以域配置在域里的各个控制器里并不是相同的。

在这个版本里，我们更新了从主机控制器注册过程来应用根域资源，现在整个域里的版本号一致了。

### 1016995 - 部署覆盖功能无法替换应用程序库

在以前的 JBoss EAP 6 里，部署覆盖（Deployment Overlay）功能无法像文档里描述的那样工作。它没有按预期地覆盖应用程序 JAR 库。现在我们进行了修改，允许部署覆盖功能可以替换应用程序 JAR 库并正常工作。

### 1077838 - SensitiveVaultExpressionConstraint 类的 isSensitiveValue 方法对 java.lang.String.substring 使用了不正确的索引

在以前的 JBoss EAP 6 里，当 write 属性值里出现多个 {} 时 **SensitiveVaultExpressionConstraint** 类会使用不正确的字符串索引

因此，由于使用了不正确的索引而导致了 **StringIndexOutOfBoundsException**。

在这个版本里，我们修复了 **SensitiveVaultExpressionConstraint** 类的 **isSensitiveValue** 方法，它会使用正确的索引，所以 **StringIndexOutOfBoundsException** 不会再出现了。

### 1074560 - 当 LDAP 组里包含斜杠字符时 EAP 管理授权抛出异常

我们发现以前的 JBoss EAP 6 版本里有一个程序错误，它阻止了在用于 Windows Active Directory LDAP 服务器上的 LDAP 组时 *backslash* (\) 字符的正确解析。这个错误会导致 EAP 管理授权抛出 **NamingException**。本版本包含的更新可以确保这个字符得到正确解析，从而不会再抛出异常了。

### 1077536 - JMX 监控客户引起的高 CPU 占用率，看起来是由 RBAC 引起的。

我们确定了以前的 JBoss EAP 6 里一个影响 JMX Query 的问题，CPU 负载会比预想的要高得多。其原因是基于角色的控制（RBAC）会检查每个查询的地址，而不管这个地址是否和 JMX Query 相关。为了解决这个问题，我们修改了查询的处理，首先检查这个地址是否和 JMX Query 相关，如果不相关则避免了进一步的处理。这个改动的结果是 JMX 监控的 CPU 负载又恢复到了正常的水平。

### 1038397 - 启用 Java Security Manager (JSM) 时基于角色的访问控制 (RBAC) 无法工作

当启用 Java Security Manager 来运行 JBoss EAP 6 时，基于角色的访问控制（RBAC）实际上被禁用，因为此时所有验证的用户都会当作超级用户对待。唯一可以使用 RBAC 的办法是不要启用 Java Security Manager。

这个版本让所有对当前 **AccessControlContext** 的访问都发生在私有动作之外，从而解决了这个问题。现在当启用 Java Security Manager 时，RBAC 仍然有效。

### 901275 - 系统属性使用库表达式会抛出 java.lang.SecurityException

如果使用了在 value 属性里带有库表达式的系统属性，以前的 JBoss EAP 6 版本将无法引导。

这是因为 XML 配置里的相同属性资源会在安全库资源之前被处理。解析表达式时安全库还不是可用的源。使用 vault 属性的任何资源都会导致下列错误：

```
java.lang.SecurityException: JBAS013322: Vault is not initialized
```

在这个版本里，如果无法解析系统属性的值表达式，在引导过程中安装了库（如果配置配置）后会进行第二次尝试，从而解决了这个问题，使用库表达式的系统可以正常引导了。

### 1092220 - 审计日志没有正确记录引导操作

以前的 JBoss EAP 版本在服务器引导期间的管理操作的审计日志缺乏效率。

- 引导操作使用了允许并行执行的特殊功能。审计日志功能没有考虑到这一点而导致了无序和混乱的日志。
- 启用了审计日志的排队执行的操作的日志记录会在添加日志处理程序的操作执行前从队列里被冲刷。这些记录没有出现在日志里。

这些问题的后果包括：

- 在添加审计日志处理程序前发生的操作的日志没有被记录。
- Extension add 操作不会被记录。
- 内部执行细节将被记录，而这是不应该的，这让某些操作看起来被执行了两次。
- 并行引导操作没有按照来自 XML 解析器的逻辑操作流的顺序组织或反映，相反它们根据子系统区块进行日志记录，而这些区块的顺序是随机的。

在这个版本里，在有机会安装任何日志处理程序及使得要记录的跟踪事件更复杂之前，队列里的日志记录不会被冲刷，从而实现了在引导过程中并行操作的正确跟踪。

所有在引导过程中发生的操作现在都会被记录，且不会包括任何无关的内部执行细节。操作将清楚地分成两组以反映服务器是如何从根本上组织引导的。在引导过程中并行执行的操作将以 XML 解析器提供的原始顺序在日志里出现。

#### 1092213 - 当设置 domain-uuid 头部时 AccessAuditContext domainUUID 没有被设置

以前的 JBoss EAP 版本里的 Infinispan 包含了一个程序错误，当 `PrepareStepHandler` 创建一个 UUID 并在操作上设置 "operations-headers" => "domain-uuid" 时，DID 没有将该 UUID 传递给 `AccessAuditContext`。

对于域控制器上的受管域的操作，这意味着审计日志记录里的 domainUUID 字段没有值，然而显示操作的 operation 字段包含一个 domain-uuid 操作的头部信息。

这个版本已经解决了这个问题。

#### 1092206 - OperationContextImpl.readResourceForUpdate 假定所有的资源都代表持久性的配置

在以前的 JBoss EAP 6 版本里，在处理 `OperationStepHandler` 的 `readResourceForUpdate` 方法时，操作执行逻辑假定所有的管理资源都代表持久性配置，

因此，`subsystem=transaction/log-store=log-store` 资源的 `probe` 操作和 `subsystem=transaction/log-store=log-store/transactions=*` 资源的 `delete` 操作无法被 `Operator` 角色里 admin 进行调用。

这个版本已经解决了这个问题。

#### 1092203 - 如果 log-read-only 为 "false"，未授权的 write 操作不会登记审计日志。

以前的 JBoss EAP 版本里有一个错误，如果 management audit-logging 资源被设置为 `false`，它会阻止未授权用户调用的 "write" 操作的日志记录。

这是因为模型控制器将 "acquisition of controller lock" 用作确定操作是否应该作为 "write" 操作记录在日志里的条件。当启用基于角色的访问控制 (RBAC) 并执行未授权的操作时，错误将在获得控制器锁前发生。

因此，如果 "log-read-only" 被设置为 **false** 时，未授权的 write 操作没有记录在审计日志里。如果 "log-read-only" 被设置为 **true**，日志记录将错误地表明这个操作为 "read" 操作。

这个版本已经解决了这个问题。

#### 1049102 当 runtime-name 和部署名称不一样时 CLI 无法显示应用程序的状态。

在以前的 JBoss EAP 6 版本里，用来确认部署状态的管理操作处理程序使用部署名称而不是 runtime-name 来查找部署服务。

因此，如果部署具有和管理名称不一样的 runtime-name，试图读取其 "status" 属性会遇到 **No metrics available**。

例如：

```
[standalone@localhost:9999 /] deploy /home/ABC.ear --name=ABC.ear --
runtime-name=XYZ.ear
```

```
[standalone@localhost:9999 /] /deployment=ABC.ear:read-
attribute(name=status)
{ "outcome" => "success", "result" => "no metrics available" }
```

在这个版本里，在查找部署服务时管理处理程序使用 runtime-name，这确保可以返回正确的状态。上面的例子的响应现在是：

```
[standalone@localhost:9999 /] /deployment=ABC.ear:read-
attribute(name=status)
{ "outcome" => "success", "result" => "OK" }
```

#### 1034700 - 启用 security manager 时无法使用 whoami 操作

JBoss EAP 6 启用 Java Security Manager 运行时无法使用 **:whoami** 操作。试图运行这个操作会导致 **IllegalArgumentException**。这是因为无效的 **AccessControlContext** 里当前调用者的标识符不正确而引起的。

本版本通过访问私有动作外部的当前 **AccessControlContext** 解决了这个问题。因此，当 JBoss EAP 6 启用 Java Security Manager 运行时，**:whoami** 操作现在可以正常使用了。

### 域管理、脚本和命令

#### 1023444 - 在 Windows 32bit JDK 上域无法用默认的内存设置来启动

以前的 JBoss EAP 6 版本用默认的内存设置运行在 Windows 32-bit JDK 环境里时可能无法启动。

在本版本里，我们降低了默认的内存设置以确保 EAP 在更多的 JVM 上可以成功启动。

我们建议依赖于默认 JVM 设置的用户复查其配置并根据要求调整 JVM 参数。

### 域管理、Web Service

#### 987898 - 写入 WS 端点的 wsdl-url 属性出现 'Unknown attribute wsdl-url' 而不是 'Attribute wsdl-url is not writable'

已部署的 SOAP Web Service 端点的 5 个属性 (name、context、class、type 和 wsdl-url) 不能通过管理工具访问。这是因为它们没有通过 Web Service 子系统对管理模型开放。本版本已经修复了这个问题，现在这些属性可以用管理 CLI 在 `webservices` 子系统下进行配置了。

## EE

### 1056799 - JBMETA-371: DefaultPropertyReplacer + PropertyResolver 无法解析库表达式

在以前的 JBoss EAP 6 版本里，`DefaultPropertyReplacer` 和 `PropertyResolver` 类里的表达式解析逻辑会假定 "\${" 和 "}" 间的表达式内容是固定的格式，而 ":" 字符则代表系统属性名称和默认值的分隔符。

这意味着部署描述符里的安全库表达式无法被成功解析，因为这些表达式总是使用 ":"，而并不是将其作为默认值前的分隔符。库表达式将被错误地评估，第一个 ":" 后的内容将被当作已解析的值处理。

在这个版本里，当检测到表达式的结尾时，在返回第一个 ":" 后面的表达式内容之前，解析器将首先检查是否可以解析整个表达式。

现在，安全库表达式可以用在允许表达式的部署描述符文件里了。

## EJB

### 1093128 - 远程客户事务过期的值被硬编码值替代

以前的 JBoss EAP 6 版本里有一个问题，它导致跨越多个服务器的远程客户事务比预期的要早或要晚过期。

这个问题的出现是因为超时时间没有正确地在服务器间传播，让系统依赖硬编码的值（300秒）。

JBoss EAP 6.3.0 已经解决了这个问题。

### 1059911 - 计算下一次超时时 @Schedule EJB Timer 未使用时区

在以前的 JBoss EAP 6 版本里，如果 @Schedule EJB timer 使用了和服务器不同的时区，在初始调用后的任何 Timer 调用都不会正确触发。这个版本解决了这个问题，当服务器的时区和 @Schedule 不一样时所有 timer 定时器调用都可以正常进行。

### 1035216 - EJBTransactionRecoveryService 上的定期恢复过程中出现 ArrayIndexOutOfBoundsException

定期恢复和 EJBTransactionRecoveryService 间间歇性出现的一个问题会导致 ArrayIndexOutOfBoundsException。

```
[com.arjuna.ats.jta] (Periodic Recovery) ARJUNA016009: Caught::  
java.lang.ArrayIndexOutOfBoundsException: 0  
  at  
  org.jboss.as.ejb3.remote.EJBTransactionRecoveryService.getXAResources(EJBTransactionRecoveryService.java:112)  
  ....
```



引起这个问题的根本原因是关联的逻辑循环里没有包含 `XAResource` 元素的创建，这导致队列索引错误。我们现已将这个元素移至逻辑循环里，所以定期恢复不会在和 `EJBTransactionRecoveryService` 冲突了。

### 1017673 - `TimerService.getTimers()` 里出现 `ConcurrentModificationException`

如果 Bean 试图在其他线程调用 `getTimers()` 方法的同时创建一个新的 `Timer`，`ConcurrentModificationException` 将被抛出。这是因为 `getTimers()` 方法没有调用 `Timer` 上的 `synchronized()` 引起的。

本版本修复了这个问题，`Timer` 服务实现的 `getTimers()` 方法现在可以正确地调用 `synchronized()` 了。

### 1031199 - EJB backing 缓存会保留大量在调度执行器的 `DelayedWorkQueue` 里取消的任务

JBoss EAP 6 里的 `@Stateful EJB` 缓存实现利用了调度执行器来处理 `@StatefulTimeout` 逻辑。当访问某个 Bean 时，之前过期的任务将被取消，在调用完成后新任务将重新调度。

在默认情况下，从执行器取消任务并不会从队列里删除它。

因为取消的任务会一直保留在队列里，这最终会导致内存溢出。

在这个版本里，您可以配置调度执行器在取消任务时从队列里删除任务。这可以避免内存溢出。

### 1045105 - 远程 `ejb` 客户代码将密码里的 '\$\$' 转换为 '\$' 进行处理

以前的 JBoss EAP 6 版本里有一个程序错误，它导致 `PropertiesBasedEJBClientConfiguration` 试图扩展包含了双美元符号（\$\$）的密码，就好像这是一个表达式。这可能会导致在服务器和客户间传递错误的密码。

本版本已经修改了 `PropertiesValueResolver`，默认情况下它不会再扩展密码了。

如果需要扩展，我们可以设置 `jboss-ejb-client.expandPasswords` 为 `true` 来启用它。

### 1055896 - 无法作为 `pass-by-reference` 获取异常

本版本解决了一个程序错误，它阻止了 `Servlet` 将 `EJB` 异常作为 `pass-by reference` 引用获取，即使是这样配置的。

### 901324 - 当在类里发现有多多个用 `@AroundInvoke` 注解的方法时，`AroundInvokeAnnotationParsingProcessor` 会失败。

以前的 JBoss EAP 6 版本不会检查类里是否有多多个用 `@AroundInvoke` 注解的方法。

因此，第一个被发现的方法将被使用，而其他的则会忽略。

在这个版本里，如果有多多个用 `@AroundInvoke` 注解的方法，部署会失败，并为开发人员显示警告信息。

### 1056214 - 由于分配了大量无用的字符串导致的 `EJB` 调用的性能问题

这个 JBoss EAP 6 版本里解决了一个性能问题。对于每次调用，`StatefulComponentInstanceInterceptor` 执行 `DEBUG` 日志里字符串的连接，从而导致性能问题。随后的分配可能导致在进行 `stateful EJB` 调用时垃圾收集活动的增加，从而导致更高的处理负荷。本版本合并了上游的补丁，降低了调用负荷，从而解决了这个问题。

## EJB, Remoting

**1098879** - 如果用于 EJB 调用的群集无法为群集节点创建连接, EJB 客户一开始就会失败, `ClusterNode{}` -> 操作失败, 状态为 `WAITING`。

在以前的 JBoss EAP 6 里, 如果为初始连接指定了多个群集节点, EJB 客户连接群集就会有问题。

这是 Windows 平台上特有的由于错误线程同步引起的问题, 会导致第一个 EJB 调用间歇性地失败。

这个版本已经解决了这个问题。

## Hibernate

**1023994** - `java.util.Calendar` 转换成 `java.util.Date` 失败 - HHH-8643

本 JBoss EAP 6 版本包含对一个程序错误的修复, 这个错误在设置 JPA 查询里的 `TemporalType.DATE` 或 `TemporalType.TIMESTAMP` 参数时导致抛出 `IllegalArgumentException`。这个参数本来是使用 `Calendar` 或 `Date` 值, 然而这种互换性并未如期操作。这个问题现已被纠正, 在查询里使用 `TemporalType` 参数时不会再出现异常了。

**1048709** - 执行 Native SQL 上的 `query.list` 时出现 NPE

在以前的 JBoss EAP 6 里, 当使用 Hibernate 查询缓存时, 使用查询里的 `scalar` 会导致用户遇到 `NullPointerException`。下面是会产生异常的查询示例:

```
query.addScalar("emp_first_name");
```

这个问题是试图自动识别处理 `scalar` 所需的 Hibernate 类型 (如 `String` 类型) 的代码引起的。在这个版本里, 我们修改了相关的代码以正确地识别所需的处理程序。

对于 6.3.0 以前的版本, 在查询 `scalar` 里显性地定义类型将避免 NPE:

```
query.addScalar("emp_first_name", new StringType());
```

**1070423** - HHH-8983 数据库驱动可能在语句句柄上附加警告信息, 积累后可能消耗大量内存。

在前一个 JBoss EAP 6 版本里, 我们发现数据库驱动会在语句句柄上附加警告信息, 积累后可能消耗大量内存。在使用 `Timestamp` 类型和底层的 Sybase `datetime` 映射时就会出现这个问题。后面的 Sybase 驱动发出的警告可能在批量更新里开始增大内存印记。本版本里已经解决了这个问题。

**1073076** - HHH-3482: `UnsupportedOperationException` with `StatelessSession`

本版本已经解决了当使用 `Stateless` 会话并保存 `ManyToOne` 时抛出 `UnsupportedOperationException` 实例。这个错误是因为调用了不是为 `StatelessSession` 实现的 `getTimestamp()` 方法而引起的。 `UnsupportedOperationException` 现在不会再出现了。

**1057742** - PostgreSQL & 对 `count-distinct` 查询的复合属性使用括号导致 H2 方言不正确

我们在 JBoss EAP 6 里进行了更新, 允许在 MySQL 和其他数据库的 `count-distinct` 查询里在复合属性周围使用括号。以前的版本是不允许这样做的, 在 PostgreSQL 里会抛出异常。本版本允许方言使用括号而不会抛出异常。

## HornetQ

### 1056216 - 修改群集连接的初始连接行为

如果群集的 HornetQ 实例丢失了和其他群集节点的连接，重新连接会导致无限循环。对于静态的群集配置，任何初始的连接都会无限进行，忽略 `reconnect-attempts` 参数。对于动态的群集配置，如果节点在它接收到节点成为群集拓扑一部分的通知和初始连接之间断开连接，重新连接将会无限地尝试。我们已经解决了这个问题，群集逻辑对于初始连接和重新连接都现在使用 `reconnect-attempts` 参数。

### 1089838 - 每次调用 `ScheduledDeliveryHandlerImpl$ScheduledDeliveryRunnable.run()` 都进行不必要的完整的 `scheduledReferences` 遍历

在以前的 JBoss EAP 6 版本里，如果大量的消息以很小的时间间隔进行调度，将会导致过度的 CPU 负荷。

其根本原因是每个实例消息都会被消费，全部 `scheduledReferences` 链接列表被不必要地遍历了。

这个版本已经解决了这个问题。

### 1063864 - 往后移植 HornetQ-1278 到 JBoss EAP

在以前的 JBoss EAP 6 版本里，Scheduled Delivery Handling 会对 Scheduled Deliveries 列表进行完整的有序搜索

这可能导致性能问题，因为系统处理很多 Scheduled Deliveries 会占用很高的 CPU 资源。

本版本实施了一个排序列表并在超时前进行正确的搜索，从而让这个列表搜索现在运行得更快，不会有很高的 CPU 负荷。

### 1089843 - 在 `XmlDataImporter` 删除临时文件后出现错误的 WARN 消息

在以前的 JBoss EAP 6 版本里，当 `XmlDataImporter` 导入大型消息时，日志里会记录错误的 WARN 消息。

其根本原因是因为当导入超大消息时，`XmlDataImporter` 会创建一个临时文件并在导入完成后进行删除。

而当导入另外一条大型消息时，`XmlDataImporter` 功能试图再次删除临时文件，而这个文件已不存在，从而在日志里登记了 WARN 消息。

这个版本已经解决了这个问题。

### 1096942 - 客户无法发送消息 - `HornetQException[errorCode=100 message=HQ119016: queue has been removed cannot deliver message, queues should not be removed when grouping]`

在以前的 JBoss EAP 6 里，用户会遇到群集分组的问题，因此路由消息可能会被中断并给出无效的应答。

在这个版本里，我们改进了节点间的通讯和组的获取过程，从而避免了这个消息及在群及分组时可能出现的其他错误消息。

### 1089844 - Policy Fail 在发送异常给客户前会丢弃消息

在以前的 JBoss EAP 6 版本里，我们发现 HornetQ 里的 "fail" address-full-policy 多数情况下会丢弃消息而不是发送异常给客户。

这导致每当地址已满时消息被丢弃而没有抛出异常。

我们已经解决了这个问题，当地址已满时 blocking sends 总是抛出异常，而当需要从服务器请求额外的凭证时，non-blocking sends 也会抛出异常。

#### 1089846 - ClientSessionImpl 里所有实现 XAResource 的方法可能抛出 non-XAException 到 TM

在以前的 JBoss EAP 6 里，如果涉及 HornetQ 的 XA 事务超时，HornetQ 可能抛出 non-XAException 到事务管理者。

我们在 `javax.transaction.xa.XAResource` 的实现里确认了这个问题，本版本里已得到解决。

#### 1089849 - 在某些重新连接的场景里，消息总是单独地应答。

在以前的 JBoss EAP 6 版本里，客户消费者在某些场景里将单独应答消息，例如在发生故障后。这个过程效率是非常低的。

这个版本已经解决了这个问题。

#### 1089851 - 在 depage 过程中避免可能的 NPE

在以前的版本里，在消息递送和分页过程中可能抛出 `NullPointerException` (NPE)。

这个版本已经解决了这个问题。

#### 1089841 - 修复 `totalIterator()` 的顺序

在以前的 JBoss EAP 6 版本里，列出消息的任何管理操作会错误地将分页消息类在已发送的消息之前。

在重递送后也可以确认其他的顺序问题。

通过调整 `Iterator` 在发送消息到管理操作前获取消息的顺序，我们修复了这个问题。

#### 1089835 - 如果 `ServerLocator` 的 `consumerMaxRate <= 0`，`ClientConsumer` 的 `max rate` 不会起作用

在以前的 JBoss EAP 6 里，如果 `ConnectionFactory` (或 `ServerLocator`) 已将 `maxRate` 设置为小于或等于零的默认值，Core API 方法 `createConsumer(String queueName, String filter, int windowSize, int maxRate, boolean browseOnly)` 会忽略参数 `maxRate`。

其根本原因是 `createConsumer` 方法里的逻辑错误。本版本已经解决了这个问题。

#### 1089842 - 如果 `filter` 属性包含 `\n` 字符，`listMessagesAsJSON` 方法无法列出消息。

在以前的 JBoss EAP 6 版本里，如果消息的字符串或其他属性里包含新行符 (`\n`)，`filter` 无法正确解析消息。

这是由于 `listMessagesAsJSON` 方法里的错误逻辑引起的，它无法解析新行符。

这个版本已经解决了这个问题。

## IIOP

### 1064644 - 当 `org.jboss.as.jacorb.rmi.InterfaceAnalysis` 分析 `javax.ejb.EJBObject` 时抛出 `StackOverflowError`

在以前的 JBoss EAP 6 版本里，我们发现由于依赖线程上下文切换的时机，启用了 IIOP 的 EJB 可能无法正确部署，从而抛出了 `StackOverflowError`。

这个问题是由于 `org.jboss.as.jacorb.rmi.WorkCacheManager` 里的线程同步不正确而引起的。

这个版本已经解决了这个问题，`StackOverflowError` 不会再出现了。

### 1052237 - 向后移植 JacORB #904 `CDRInputStream.read_string` 应该优雅地处理长度为 0 的字符串

`CDRInputStream.read_string` 函数错误地处理了空字符串，将其长度计算为 0，导致编组异常。我们添加了 QoS 来允许 `CDRInputStream.read_string` 可以和这些没有正确对空字符串编码的 ORB 互用。

## 安装程序

### 1034062 - 域的主机文件里端口偏移量配置的值翻倍了

当使用 **Configure an offset for all default port bindings** 选项来设置端口偏移量时，一个程序错误导致这个值翻倍了。本版本已经修复了这个问题。

### 1062602 - 安装程序接受不带字母的管理性用户的密码

在以前的 JBoss EAP 6 版本里，我们发现图形化安装程序工具没有遵循控制创建用户的用户名和密码限制。

这个版本已经解决了这个问题，GUI 安装程序现在会遵循密码和命名限制了。

## JCA

### 1088470 - 如果 TSR 在 `SemaphoreArrayListManagedConnectionPool` 里抛出 `IllegalStateException` 和 `NPE`，`ConnectionListener` 将会溢出。

如果 TSR 在 `SemaphoreArrayListManagedConnectionPool` 里抛出 `IllegalStateException` 和 `NPE`，以前版本里的一个程序错误将导致 `ConnectionListener` 的溢出。

这个版本已经解决了这个问题。

## JDR

### 1069850 - 阻止 JDR `CommandLineMain` 里抛出 `NullPointerException`

在以前的 JBoss EAP 6 版本里，每当 JDR `CommandLineMain` 抛出一个异常，`NullPointerException` 就会被抛出。本版本已修复了这个问题。

## JMS

### 1033495 - 域模式下 commitPreparedTransaction CLI 操作不可用

在以前的 JBoss EAP 6 版本里, hornetq-server 资源上的管理操作在域模式下不可用。本版本里已修复了这个问题。

## JPA

### 1040733 - JBoss AS / Hibernate JPA 集成时出现内存溢出

当使用管理 CLI 来获取使用 JPA 命名查询的应用程序部署的 JPA 统计时可能发生内存溢出。本版本里已解决了这个问题。

## JSF

### 1029387 - WFLY-2493 EL 不能访问非 public 类的 public 方法/字段

在以前的 JBoss EAP 6 版本里, BeanELResolver 没有尝试覆盖对 Method.invoke 访问非 public 类的 public 方法/字段的访问控制。

这导致了当试图通过 Expression Language(EL) 访问非 public 类的 public 方法/字段时, 会出现下列错误信息:

```
"java.lang.IllegalAccessException: Class javax.el.BeanELResolver can not access a member of class X with modifiers "private"
```

我们通过调用 EL 实现里的 setAccessible(true) 解决了这个问题。现在可以通过 EL 访问非 public 类的 public 方法/字段了。

### 1017242 - 在根上下文应用程序里 FacesMessages 无法正常工作

在以前的 JBoss EAP 6 里, 我们发现如果应用程序绑定到根上下文, JSF 的 Flash scope 在重定向时没有正确恢复。这意味着 FacesMessages 在根上下文应用程序里无法正常工作。

本版本里的 Mojarra 组件升级到了最新的上游版本, 从而解决了这个问题。现在 FacesMessages 可以在根上下文应用程序里正常工作了。

### 1052265 - JAVASERVERFACES-3080: 当多个 f:viewParam 包含在 f:metadata 里时出现问题

有有上游版本里的一个程序错误, 在以前的 JBoss EAP 6 里, **f:metadata** 无法包含多个 **f:viewParam**。升级 JSF 组件后可以解决这个问题, 现在 **f:metadata** 可以包含多个 **f:viewParam** 了。

### 1054051 - JSF slot 配置无法用于 default 配置

当额外的 JSF slot 安装在 **modules** 目录的根目录下时, *main* JSF slot 没有被添加为有效的 JSF 配置。尝试使用 *main* JSF 实现的 JSF 应用程序将失败并显示如下信息:

```
org.jboss.as.server.deployment.DeploymentUnitProcessingException:  
JBAS012656: Default JSF implementation slot 'main' is invalid
```

我们解决了这个问题, 现在 "main" slot 总是被当作有效的 JSF 实现。因此, 当额外的 JSF slot 安装在 **modules** 目录的根目录下时, 使用 *main* JSF 实现的 JSF 应用程序将可以成功部署了。

## 日志

### 1066597 - formatter 属性在每次被处理时都会改变

通过合并上游的补丁，以前的 JBoss EAP 6 里导致 `HandlerOperations.equalValue()` 方法总是返回 `false` 的程序错误已被修复。因为在比较值时，这个方法使用了错误的属性名，从而出现了这个程序错误。本版本里已不会再出现这个错误了。

### 1080991 - Messages#getBundle() in jboss-logging 缺少了 privileged action

在以前的 JBoss EAP 6 版本里，启用安全管理者而没有足够的权限来运行会导致获取消息束的错误。

因此，如果没有启用类加载器权限，当试图获取消息束会抛出异常。

在这个版本里，获取消息束会在私有块里完成。启用安全管理者时获取消息束不会再抛出异常了。

### 1088618 - throwable 信息的缓存字符串表示形式

在以前的 JBoss EAP 6 里无法保证正确存储栈跟踪信息。因此，在序列化 log4j 日志事件时会导致数据丢失。

为了修复这个错误，我们修改了 Log4J 来缓存反序列化时的栈跟踪信息（也就是 throwable 信息）。这确保了在序列化时不会丢失栈跟踪信息。

### 1017881 - /subsystem=logging/logger=org.jboss.as.quickstarts.logging:assign-handler 在批处理模式下无法运行

在以前的 JBoss EAP 版本里，composite logging 操作里的 name 属性被添加到模型并从中读取。

因此，composite 操作的 last name 是所使用的唯一处理程序名称，导致相同的处理程序名称被多次添加。

在这个版本里，name 属性不会再复制到模型里，并从操作本身读取。再 composite 操作里添加处理程序现在可以正常进行了。

### 1095516 - 当手动从 JBoss Config XML 里删除 POJO 对象时，它们的引用并没从 logging.properties 里删除。

在以前的 JBoss EAP 6 版本里，当 POJO 的定义从服务器的配置文件里删除时，对 POJO 的引用没有从 `logging.properties` 文件删除。

如果在之后创建具有相同名称的 POJO，JBoss EAP 将表示在 `logging.properties` 里有重复的条目。

这个问题已得到解决，对 POJO 的引用将从 `logging.properties` 文件里删除。

### 1073053 - EAP 6.2 审计日志显示的是 EAP 版本而不是 AS 版本号

在以前的 JBoss EAP 6 版本里，审计日志将显示错误的版本号。

这个版本已经解决了这个问题，现在可以显示正确的版本号了。

### 1066606 - 将 log4j appender 用作 custom-handler 如果有需要时应该调用 activateOptions

在以前的 JBoss EAP 6 版本里，如果 appender 实现了 `OptionHandler`，修改作为 log4j appender

的 custom-handler 里的属性不会调用 appender 上的 `OptionHandler.activateOptions()`。它要求重启 loggin 资源使这个修改生效。在这个版本里，如果 log4j appender 上的属性有修改则会调用 activation 方法，这样 `OptionHandler` appender 就不需要重启了。

### 1070452 - 当使用 per-deployment 日志时 System.out.println() 无法工作

以前的 JBoss EAP 6 版本里有一个程序错误，在使用 per-deployment 日志时，它阻止了 `System.out.println()` 方法输出到日志文件里。通过上游补丁，我们解决了这个问题。

## Naming

### 1014414 - 对于不同的原因，Remote Naming 却抛出相同的异常。

在以前的 JBoss EAP 6 版本里，当 Remote Naming 客户无法连接到任何主机时，对于许多连接错误都抛出相同的异常。这个行为是次优的，它并没有为用户提供实际的错误指示。

抛出的异常是：

```
javax.naming.NamingException: Failed to connect to any server. Servers
tried: [remote://localhost:4447]
```

在这个版本里，抛出的异常更贴近于实际的错误原因。

如果主机或端口细节是不正确的，下面的 `CommunicationException` 将被抛出，指示连接超时：

```
javax.naming.CommunicationException: Failed to connect to
any server. Servers tried: [remote://localhost:4447 (Operation failed
with status WAITING after 5000 MILLISECONDS), remote://localhost2:4321
(Operation failed with status WAITING after 5000 MILLISECONDS)] [Root
exception is java.net.ConnectException: Operation failed with status
WAITING after 5000 MILLISECONDS]
```

如果可用服务器中的一个有响应，但随后的验证失败，那 `AuthenticationException` 将被抛出：

```
javax.naming.AuthenticationException: Failed to connect to
any server. Servers tried: [remote://localhost:4447 (Authentication
failed: all available authentication mechanisms failed),
remote://localhost2:4321 (Operation failed with status WAITING after
5000 MILLISECONDS)] [Root exception is
javax.security.sasl.SaslException: Authentication failed: all available
authentication mechanisms failed]
```

现在对于连接列表里每台服务器的故障，都会给出合适的信息。

### 1061609 - InitialContext 吞没了原始的异常原因

在以前的 EAP 6 版本里，InitialContext 的实例化可能失败并显示如下信息：

```
javax.naming.NamingException: JBAS011843: Failed instantiate
InitialContextFactory com.sun.jndi.ldap.LdapCtxFactory from classloader
ModuleClassLoader for Module
"deployment.externalContextBindingTest.jar:main" from Service Module
Loader
```



然而，在这些信息里并看不到底层的原因，使得故障解除无法进行。为了解决这个问题，现在显示了底层的原因。现在如果发生问题，错误信息将显示具体原因：

```
javax.naming.NamingException: JBAS011843: Failed instantiate
InitialContextFactory com.sun.jndi.ldap.LdapCtxFactory from classloader
ModuleClassLoader for Module
"deployment.externalContextBindingTest.jar:main" from Service Module
Loader [Root exception is javax.naming.CommunicationException:
127.0.0.1:10389 [Root exception is java.net.ConnectException: Connection
refused]]
```

### 1059836 - Remote Naming 的通讯异常应该在 `ConnectException` 上抛出

在以前的 JBoss EAP 6 版本里，当 `java.net.ConnectException` 发生时，通用的 `javax.naming.NamingException` 而不是更具体的 `javax.naming.CommunicationException` 将被抛出。

这个版本里进行了修改，可以确保在连接异常发生时 `javax.naming.CommunicationException` 被抛出。

`CommunicationException` 是 `NamingException` 的子类，所以之前捕获 `NamingException` 的代码也可以正常工作。

## 其他

### 901210 - Cleanup 部署目录 - AS7-6031

JBoss EAP 6.3 确保了 `JBOSS_HOME/tmp` 和 `JBOSS_HOME/tmp/vfs` 里创建的文件和目录在可能干扰新启动的 EAP 实例前被删除。

在以前的 JBoss EAP 版本里，服务意外关闭后可能会留下一些旧的文件（作为关闭过程的一部分，JBoss EAP 会删除 `JBOSS_HOME/tmp` 和 `JBOSS_HOME/tmp/vfs` 里的文件）。

这个版本提供了一种失效保护措施来减缓这种状况。如果 JBoss EAP 6.3 服务器没有优雅地关闭，服务器将没有机会清除这些临时文件。然而，在重新启动后，服务器现在会检查上述目录，如果存在来自前一实例的文件，服务器将初始化一个重命名/删除过程，并允许为新的实例创建新的文件（旧的目录将被重命名来避免和新创建的文件冲突）。这些过程是并发进行的。

在重新启动 JBoss EAP（优雅地或以其他方式）后，旧的临时文件将被删除（在服务关闭或重启时）以确保它们不占用不必要的磁盘空间。

**注意** 用户应该避免使用 `-Xrs JAVA_OPT`，因为这个会限制信号的处理并导致 `tmp/vfs` 目录的大小继续增长。

## 补丁

### 1110117 - 关闭时查询 `patching MBeans` 导致 `IllegalStateException`

在本版本的 JBoss EAP 6 里，`patching` 子系统 `PatchResource` 虽然没有对 `InstallationManagerService` 的依赖关系，但在关闭时会试图使用它。

这会导致当关闭 hook 试图查询 `patching` 子系统 MBean 时抛出 `IllegalStateException`。

我们期望在下一个版本里解决这个问题。

### 1108952 - 应用大型补丁时出现 `OutOfMemoryError`

在以前的 JBoss EAP 版本里，从主机控制器传递到从主机控制器的附件数据会被全部读入内存。

因此，当安装大型补丁时，子主机控制器上可能发生 `OutOfMemoryError`。在通过域控制器使用默认的内存设置安装 CP04 时可能就会出现这种情况。

本版本的附件数据会保存在临时文件里，并不会消耗内存，所以子主机控制器上不会发生 `OutOfMemoryErrors`。

## PicketLink

### 1084596 - 移植 PLINK-396

在以前的 JBoss EAP 6 版本里，我们发现 PicketLink 的 `IDPWebBrowserSSOValve` 和 `IDPFilter` 解码了 `relaystate`，这和 SAML 规格是相反的。本版本已解决了这个问题。

## RESTEasy

### 1037753 - 选择的变体并不总是最合适的

在以前的 JBoss EAP 6 版本里，我们发现虽然 RESTEasy 遵循了 RFC 2616 规格，但并没有总是返回最合适的媒介处理程序，其品质因素相等但特征是不一样的。

例如，当给定 `application/json`，`Accept` header 和变体值 `["application/xml", "application/json"]`，RESTEasy 的 `Request.selectVariant()` 会选择 `application/xml` 而不是 `application/json`。

在这个版本里，专有的 `Accept` header 值会优先于非专有的具体相同品质值的变体值（例如都具有 `q=1.0` 或 `q=0.5`）。

### 1014393 - RHEL 5, 6 里 IBM jdk 16, 17 上的 `resetStream` 抛出 `Stream closed` 异常

在以前的 JBoss EAP 6 里，IBM JDK 16 / 17 提供的 `xercesImpl` 和 `resteasy-jaxb-provider` 使用的 `jaxb` `Unmarshaller` 有冲突。

当用户直接使用 EAP 6 提供的 `xercesImpl` jar 就会出现这个问题。

这些冲突导致了在将 IBM JDK 16, 17 或 `xerces:xercesImpl:2.9.1-redhat-x`（EAP 6 所提供的）用作基于 `resteasy 2.3.6.Final-redhat-x` 的项目的依赖关系时会出现 `java.io.IOException: Stream closed` 错误。

这个版本已经解决了这个问题。

## RPM

### 1086157 - 执行 `yum update` 时出现 RHEL6/RHEL5 Web Server 和 `httpd-manual` 错误

在以前的版本里，在运行在 RHEL 系统上的 JBoss EAP 6 里执行 `yum update` 会产生错误。这是因为 `jbappplatform-6-i386-server-6-rpm` 频道缺少了 RPM 依赖关系 (`httpd-manual`)。我们已在这个频道里添加了 `httpd-manual` 软件包，现在执行 `yum update` 不会再出现错误了。

## Remoting

### 1052204 - 可序列化的类和不同的不可序列化超类之间的协议不兼容性

在以前的 JBoss EAP 6 版本里，当 IBM 和 Oracle JVM 通过序列化 `StringBuilder` 或 `StringBuffer` 类来通讯时可能会抛出异常。

这是因为编组协议错误地序列化了可序列化类的第一个不可序列化超类的类描述符。

本版本通过升级 JBoss Marshalling 组件解决了这个问题。

### 1102271 - JBoss Marshalling 要求类的字段不能为 null

在以前的 JBoss EAP 6 里，如果对象包含的某个字段的值为 null 且这个字段的对象类型没有存在反编组队列里，试图反编组对象都会失败并抛出 `ClassNotFoundException`。

这个版本已经解决了这个问题。

### 1069075 - 使用 `jboss-client.jar` 调用 EAP 上的 EJB 时 Tomcat 上的线程溢出并抛出 `OutOfMemoryError`

升级 JBoss Remoting 可以解决以前版本里的 `OutOfMemoryError` 问题。这是当 Tomcat 上的 Web 应用程序调用 EJB 时发生的线程溢出引起的。

### 1011831 - JBREM000205: Failed to accept a connection: `java.nio.channels.ClosedChannelException` at server shutdown

在服务器关闭过程中，remoting 子系统里的 **DEBUG** 信息错误地记录成 **WARN** 信息。记录的信息类似于：

```
02:46:15,512 WARN [org.jboss.remoting.remote] (Remoting
"node1:MANAGEMENT" read-1) JBREM000205: Failed to accept a connection:
java.nio.channels.ClosedChannelException
```

本版本的 JBoss EAP 6 升级了 remoting 子系统，已将信息重新归类为 **DEBUG** 级别。

### 1080429 - 将 JBREM000200 从 **ERROR** 修改为 **DEBUG**，既然它是无害的。

Windows 平台上以前的 JBoss EAP 6 的用户在关闭 JMX 连接时会遇到下列 `IOException`：

```
JBREM000200: Remote connection failed: java.io.IOException: An existing
connection was forcibly closed by the remote host
```

这是因为 Windows 强制地关闭连接引起的。它并没有负面的影响，日志级别 **ERROR** 可以修改为 **DEBUG**，以避免出现在更底层级别的日志里。

### 1052258 - 应用程序在 IBM-I 的 IBM JDK 上运行时出现 `segfault` 和其他 `xnio` 问题

这个 JBoss EAP 6 版本纠正了运行在 IBM 系统上的应用程序因为 `segmentation fault` 而运行失败的问题。

题。

这是因为没有为 IBM 操作系统进行优化的 IBM JDK 的 NIO 实现引起的。JBoss EAP 里的 I/O 层试图检测和利用这些实现。然而，在某些操作系统（如 IBM-I），这些实现会导致 segfault。

这个版本里会检测这些操作系统并使用安全 fall-back。这解决了 segmentation fault 问题，应用服务器不会再意外地崩溃了。

## Remoting, Web

### 1032552 - 由于大量的 `org.apache.tomcat.util.net.JIoEndpoint$Poller` 对象出现 OOM

本版本纠正了线程溢出可能导致的 `OutOfMemoryError`。这个溢出可追溯到 `JIoEndpoint` 类。我们更新了相关的代码，现在 OOM 错误不会再出现了。

## 脚本和命令

### 1062595 - 如果用户名和密码相同（非交互模式），`add-user` 将抛出 `RuntimeException`。

如果对 `add-user` 工具的非交互式调用失败（如有问题的用户名/密码组合），以前的 JBoss EAP 6 会抛出 `RuntimeException`。这个异常的目的是警告脚本故障已经发生。然而，这种类型的异常不应该被处理就进行传播。我们在这个版本里添加了一个自定义异常以表示这个异常是有意为之，而不是表示 `add-user` 里有程序错误。

### 1027165 - `add-user.sh` 要求控制台输出

以前的 JBoss EAP 6 版本里有一个问题，添加用户到 EAP 服务器的脚本（`add-user.sh`）如果没有控制台（非交互模式）则无法执行。

这是因为脚本（`add-user.sh`）依赖于控制台（`java.io.Console`）来进行操作。

执行这个脚本（`add-user.sh`）会抛出下列异常并中止：

```
java.lang.IllegalStateException: JBAS015232: No java.io.Console
available to interact with user.
```

这个版本已经解决了这个问题。

### 1063888 - `add-user` 脚本使用 `help` 参数时显示错误的密码信息

在以前的 JBoss EAP 6 版本里，`add-user` 工具的 `help` 参数输出只显示关于密码的单个限制条件（密码不能和用户名一样）。在添加用户时这会造成混淆，因为为了确保使用密码有效，需要多个限制条件。本版本从 `help` 输出文本里删除了使用交互时模式时的单个限制条件以及其他适用的限制。

### 1062611 - `add-user`: 检验用户名时 '@' 没被当作非字母字符

当输入无效的用户名时，以前的 JBoss EAP 6 里的 `add-user` 会显示误导的错误信息。这个信息指出用户名必须只包含字母，而实际上它可以接受特殊字符的子集。我们重写了这些错误信息使其包含可接受的字符的列表，现在当输入无效的用户名时，用户可以看到更为准确的错误信息了。

### 1020677 - 域模式的服务器脚本使用了错误的字符串来进行检验 JBoss 是否已启动。

在以前的 JBoss EAP 6 版本里，我们发现在确认服务器是否已正确启动时域和独立服务脚本都使用了错误的字符串。

错误的变量。

当试图检验服务器的状态时，这会导致不正确的结果。

这个版本已经解决了这个问题。

### 956281 - 在带有 32位 JVM 的 Windows 上启动 EAP 6.1 会导致启动 JVM 失败

在带有 32-bit JVM 的 Windows 8 上运行以前的 JBoss EAP 6 版本时，默认的 `t max perm gen space` 设置会导致不能创建 JVM，然后阻止了服务器的启动。

这个版本已经解决了这个问题。

### 1057127 - 当补丁升级了一些依赖关系时，`jconsole.sh` 无法工作。

以前的 JBoss EAP 6 版本里有一个程序错误，每当应用 CP1 补丁时，它都会阻止 `jconsole.sh` 脚本的运行。这个补丁更新 `jconsole.sh` 里一些硬编码的模块，同时也改动了原始的模块 JAR 文件。

这意味着不可能用 `jconsole.sh` 连接到 EAP 控制台。

使用 `bin/client` 里的 `jboss-cli-client.jar` 可以修复这个问题，它包含了所有需要的依赖关系。

### 1062592 - `add-user` 工具给出的令人混淆的错误信息

本版本的 JBoss EAP 6 已经修改了 CLI `add-user` 工具的一条错误信息以避免混淆。

在以前的版本里，如果输入的密码只包含数字字符时，将会显示表示用户密码需要至少包含一个字母数字字符的错误信息。本版本里相关的错误信息会表示只有数字的密码必须包含至少一个字母字符。

### 1057625 - `add-user.sh` 语法针对 Cygwin 的修改（用于 EAP 6.3.0）

在以前的 JBoss EAP 6 版本里，我们发现在 Cygwin 环境里 `add-user.sh` 脚本无法运行。

这是由于脚本里一行错误格式的代码引起的。

这个版本已经解决了这个问题，然而在 Cygwin 环境里运行时这个脚本仍带有一个未解决的问题。详情请参考本文档的『已知问题』里的 Ticket 1069252。

### 928486 - 对密码的要求应该只显示一次

在以前的 JBoss EAP 6 版本里，用户在使用 `add-user` 工具时输入无效密码只会接收关于第一次违反密码规则的错误信息。

如果用户违反了多条规则，在选择有效密码前将需要多次尝试创建密码。

在这个版本里会显示密码限制的完整列表，减少了尝试创建密码的次数。

## 安全性

### 1023084 - JBossJSSESecurityDomain.java 里的程序错误 - 试图使用错误的供应商

以前的 JBoss EAP 6 版本里有一个程序错误，它导致 `JBossJSSESecurityDomain.java` 试图使用密钥库/信任库供应商来获取信任管理者的实例。这种行为是错误的，因为 `"trust-manager-factory-provider"` 设置无法用在安全域的 JSSE 里。使用这个设置（即使正确地配置）将在启动时抛出异常。本版本已

经纠正了这个程序错误，现在可以用 "trust-manager-factory-provider" 设置来设置 `trustManagerFactoryProvider` 了。

### 1065476 - AdvancedLdap 登录模块无法处理在 UID 里包含斜杠字符的用户

在以前的 JBoss EAP 6 版本里，如果请求 ID (UID) 包含斜杠字符 (/)，验证请求将会失败。这是因为 AdvancedLdap 登录模块没有正确处理这个字符。在这个版本里，我们更新了这个模块，在绑定前从用户 DN 里删除了引号。

### 1069127 - RBAC + LDAP 需要和 <local/> 登录机制一起使用

在以前的 JBoss EAP 里，如果一个验证的用户无法映射到 LDAP 帐号时 LDAP 组加载将会失败。这是因为使用安全区的验证过程首先会在客户和服务器间协商一个机制，然后再加载用户的组信息。因为 local 验证系统代表带有人工用户名的用户，而如果 LDAP 服务器无法映射这个用户名到某个用户，这个过程的第二部分会失败。

在这个版本里，我们在用于本地验证的 <local /> 元素里添加了新的属性 `skip-group-loading`。当这个属性被设置为 `true` 时，组加载将在本地验证发生后被跳过，从而避免了错误。如果使用了不同的机制，组加载则进行正常处理。

### 1066488 - 管理安全区：LDAP referrals 无法工作

以前的 JBoss EAP 6 有一个程序错误，所有的用户和组都必须在相同的 LDAP 服务器上定义且是可搜索的。

导致 referral 的任何用户和组条目都不能和 JBoss EAP 6 一起使用。

这是因为在根据 LDAP 验证用户或使用 LDAP 加载组时，安全区里的 LDAP 搜索没有包含处理 referral 的逻辑。

这个版本已经解决了这个问题，现在 LDAP referral 可以按预期那样工作了。

### 1030053 - NegotiationAuthenticator 丢失了 POST 数据

在以前的 JBoss EAP 6 版本里，我们发现如果 `NegotiationAuthenticator` 和 PicketLink 及 HTTP\_POST 绑定一起使用时会丢失任何 SAMLRequest 参数。这导致了即使在验证成功后，用户仍会停留在 IDP 登入页面。在这个版本里，我们已安装了 `NegotiationAuthenticator` 的补丁，这个问题不会再出现了。

### 1065486 - LdapExtended 登录模块不会处理在 UID 里包含斜杠字符的用户

在以前的 JBoss EAP 6 版本里，如果用户 ID (UID) 包含斜杠字符 (/)，用户验证将会失败。这是因为 LdapExtended 登录模块没有正确处理这个字符。在这个版本里，我们更新了这个模块，在绑定前从用户 DN 里删除了引号。现在用户可以按预期进行验证了。

### 974324 -用于管理的 EAP 6 域模式日志不存在

在以前的 JBoss EAP 6 版本里，我们没有添加 TRACE 和 DEBUG 日志到安全区的 LDAP 交互里。这使得诊断 LDAP 的验证问题极为困难。我们现在添加了 DEBUG 日志到使用了 LDAP 的安全区里。客户现在可以使用这些日志来诊断和 LDAP 相关的问题。

### 1069885 - JAAS 缓存没有缓存 SecureIdentityLoginModule (以及 ConfiguredIdentityLoginModule) 结果

在以前的 JBoss EAP 版本里，当使用 `SecureIdentityLoginModule` 而没有缓存加密的数据源密码时会出现性能问题。这是当使用数据源的应用程序未被保护时 JAAS 缓存不允许缓存键为 `null` 而导致的。

在这个版本里，密码库用来加密数据库密码，绕过了 JAAS 登录模块并解决了性能问题。

#### 1067610 - 如果 `DatabaseRolesMappingProvider` 的 `rolesQuery` 返回一个空集，验证会失败。

在以前的 JBoss EAP 6 版本里，我们发现如果 `DatabaseRolesMappingProvider` 返回一个 `null` 值，验证将会失败。这个问题是由于如果这个值为 `null`，验证无法提供角色给被验证的用户而导致的。在本版本里，安全系统将承认成功的验证且不会试图在返回值为 `null` 的实例里应用角色。

#### 1000185 - JASPI auth-module 配置不支持 "module" 属性

在以前的 JBoss EAP 6 版本里，安全子系统里的 JASPIC auth-module 配置会忽略 "module" 属性。这个属性告诉 PicketBox 到哪去加载自定义的 auth-module 类。

因此，我们不能配置自定义的 JASPIC auth-module，因为 PicketBox 无法确定应该使用的 jboss 模块。

因为 `module` 属性已经存在于安全子系统模式里，我们需要在安全子系统里添加代码来处理这个属性，允许 PicketBox 正确地加载自定义模块。

因此，用户现在可以用 "module" 属性来配置自定义 JASPIC auth-module，指定包含自定义模块类的 `jboss-module`。

## 服务器

#### 955818 - 没有正确处理 WARs-in-EAR 的 Class-Path manifest 条目

在以前的 JBoss EAP 6 版本里，当 EAR 里的多个子部署通过 Class-Path manifest 条目引用单个非模块 JAR 文件时，它会被添加到第一个子部署的类加载器里。

这导致类加载问题，因为 Utility Jar 里的类将被加载到错误的类加载器里。

本版本为 Utility JAR 创建了一个新的部署模块，所有依赖的部署都使用它。因此，类加载问题不会再出现了。

#### 1060269 - 主域服务器重载的反向 DNS 查找

以前的 JBoss EAP 6 版本的用户可能已发现，如果受管服务器没有重启，服务器在重新加载后会无法重新连接到域控制器。

如果没有正确配置反向 IP 查找，而服务器用控制器的主机名（非 IP 地址）进行重新连接，这个问题就可能会发生。这种情况下连接会失败。

通过重用保存在 `InetSocketAddress` 对象里的数据，本版本已经解决了这个问题。这避免了反向 IP 查找的需要并允许服务器如期进行重新连接。

#### 1036872 - 无法配置禁用在 JRE 的 `java.security` 文件里指定的策略的安全策略文件设置

使用特殊前缀 `=` 来禁用默认的策略文件会出现问题。底层的原因时当主机控制器启动服务器时，它 `java.security.policy` 设置提供了 `null` 值，导致使用了 `java.security` 文件里指定的策略文件以及可能的权限问题，从而阻止了服务器的引导。

通过修改系统属性的解析，我们解决了这个问题。以 "=" 开始的系统属性的值不再在主机控制器启动服务器时被重置为 `null`。

#### 1049999 - Class-Path: . can cause JBAS011046: 这个模块里已经定义了名为 'TestBean' 的组件

在以前的 JBoss EAP 6 版本里，WEB-INF/lib 里的一些 JAR 库文件包含了带有 "." 条目的 Class-Path manifest 属性。

这个问题导致了一些资源或组件被两次处理，造成日志里出现警告信息。

本版本忽略了 Class-Path manifest 属性里的 "." 条目，所以资源不会再被处理两次了。

#### 924562 - 依赖关系替换导致的部署重启无法进行

这个 JBoss EAP 6 版本里解决了导致 `NullPointerException` 的一个问题。当部署部分重启某些已被清除以节省内存的数据结构时（例如当替换某个依赖关系时），这个异常可能出现。这个问题键该阻止重部署的完成。为了解决这个问题，本版本不再允许部分重部署。如果依赖关系被替换，部署将完全重启，而这个异常不会再出现了。

### 事务管理者

#### 1038993 - 无法通过 CLI 命令将对象库类型从 `hornetq` 修改为 `jdbc`

在以前的 JBoss EAP 6 里，通过 CLI 界面修改对象库类型（从 `HornetQ` 到 `JDBC`，反之亦然）没有正确填充。

因此，对象库将保持初始设置（可能缺少所需的选项）。

在这个版本里，我们加强了写处理程序 `use-hornetq-store` 和 `use-jdbc-store` 来禁用其他选项，

#### 1034650 - 增大 `com.arjuna.ats.jta.orphanSafetyInterval` 的默认值

如果事务的处理时间和定期恢复进程的活动重叠，以前的 JBoss EAP 6 版本会导致大量的回滚 `XAExceptions` 被登记日志，本版本已修复这个问题。`orphanSafetyInterval` 的超时间隔已增至 20 秒，这显著地减少了遇到异常的可能性。

#### 1027126 - 使用 `standalone-xts.xml` 和 `mysql 5.5` 上运行的 `jdbc` 对象库时服务器无法启动

在以前的 JBoss EAP 6 里，如果没有配置事务管理者运行 XTS 事务，而使用了 MySQL 5.5 上的 `JDBC` 对象库，驱动会放入 `deployments`，导致服务器无法启动。

这个版本已经解决了这个问题。

#### 1107569 - 单阶段提交优化：Bean 调用实际失败了，但抛出的 `XAException` 并未被捕获。

以前的 JBoss EAP 6 版本里的一个程序错误阻止了用户看到表示单阶段提交失败的异常。

因为资源管理者可能执行 `XAR::end` 失败但执行 `XAR::rollback` 成功，而不会报告异常给用户，从而出现了这个错误。

本版本会抛出正确的异常，以确认单阶段提交的结果。

#### 1092198 - `LogStoreProbeHandler` 覆盖了 `LogStoreResource` 根模型



在以前的 JBoss EAP 6 版本里，我们发现 `LogStoreProbeHandler` 替代了 `LogStoreResource` 持有的 `delegate`。这会从现有委托的 `model` 字段（如 "type" 属性）删除任何数据。

因此，调用 `subsystem=transactions/log-store=log-store` 上的 `probe` 操作将导致资源的 `type` 属性的值被修改为 `default`，即使事务管理者实际上使用了 `hornetq`。实际的运行时行为不会受影响，但报告的值将是错的。

在这个版本里，当前委托模型的内容会在被删除前复制到新委托里，当日志库类型实际上是其他类型时，在运行 `probe` 操作后，`type` 属性不会再是 `default`。

## Web

### 1027272 - 使用群集 SSO 时会话失效会抛出 `ContextNotActiveException`

在以前的 JBoss EAP 6 版本里，当过期的会话和 SSO 关联时 SSO Valve 没有设置上下文。

因此，在会话被销毁后 `ClusteredSingleSignOn` 将调用 `WeldListener.sessionDestroyed(event)`，导致会话失效上出现 `ContextNotActiveException`。

在这个版本里，当过期会话和 SSO 关联时 SSO Valve 会设置上下文，会话失效时也避免了 `ContextNotActiveException`。

### 1050204 `WAIT_FOR_BEFORE_START` 无法用于 "/" 上下文应用程序

在以前的 JBoss EAP 6 版本里，`WAIT_FOR_BEFORE_START` 属性无法用于根上下文应用程序。如果用户将 `WAIT_FOR_BEFORE_START` 设置为 "/" 并部署一个根应用程序，连接器将无法启动。

这个版本已经修复了这个问题。当设置 `WAIT_FOR_BEFORE_START` 为 "/" 并部署根上下文应用程序时，连接器现在可以成功启动了。

### 1105160 - 对 `JSSupport keySizeCache map` 的并发访问导致高 CPU 占用率

以前的 JBoss EAP 6 版本没有正确地同步 `JSSupport` 里的 `keySizeCache`

这意味着对 `JSSupport keySizeCache` 的并发访问会导致产生高 CPU 占用率的 Hash 表循环。

在这个版本里，对 `JSSupport` 里的 `keySizeCache` 的访问被同步，不会出现对 `keySizeCache` 的并发访问。

### 1036197 - 如果 `org.apache.tomcat.util.Constants.ENABLE_MODELER` 被设置为 `true`，Native HTTP 连接器将无法使用。

在以前的 JBoss EAP 6 里，`JBossWeb Coyote` 的 `*Protocol` 类的 `start()` 方法使用 `getName()` 的值构建了一个 MBean 名称，它很可能包含一个冒号而 MBean 对象名不能包含任何冒号。

如果设置了 `-Dorg.apache.tomcat.util.Constants.ENABLE_MODELER=true`，当 JBoss 试图启动时会报告下列错误：

```
JBWEB003044: Threadpool JMX registration failed:
javax.management.MalformedObjectNameException: Invalid character ':' in
value part of property
```

这个版本使用 `getJmxName()` 而不是 `getName()` 来构建 MBean 的名称。

因此，当设置 `-Dorg.apache.tomcat.util.Constants.ENABLE_MODELER=true` 时并启用 Modeller 时，异常不会再出现了。

## Web 控制台

### 1079948 - 在 JBoss EAP 6.2 管理控制台里无法看到所有主机

在以前的 JBoss EAP 6 版本里，主机选择器里没有滚动条。这使得选择未显示的主机变得很困难。

我们在这个版本的主机选择器里添加了一个滚动条，现在可以很容易地选择所有主机了。

### 1014219 - RBAC: 控制元素对于具有多个 `scoped` 角色的用户的可见性问题

分配有多个角色的用户将在控制台里看到他们并没有访问权限的操作。例如，具有 `host-master-administrator` 和 `host-slave-monitor` 角色的用户只应该看到从主机上下文里的控制元素（如服务器配置页面里的 **Add** 按钮）。在主主机上下文里这个按钮不应该可见（然而它是可见的）。

如果执行不应出现的操作将会失败，因为在操作执行时会强制实施正确的访问控制。这不会造成安全漏洞。

本版本已修复了这个问题。和用户角色不相关的控制元素将变灰并处于不活动状态。

### 900849 - 当数据大于 64k 时 EAP6 CLI 出现错误

任何返回超过 65535 个字符的字符串的 CLI 命令都无法运行并出现下列错误：

```
Communication error: java.util.concurrent.ExecutionException: Operation failed
```

这是由于使用了 JBoss DMR 库里的 `java.io.DataOutput.writeUTF()` 方法引起的。如果处理的字符串超过 65535 个字符，这个方法会抛出 `UTFDataFormatException`。我们已更新了 JBoss DMR 库来使用其他技术以正确处理超过这个长度的字符串。现在，返回超过 65535 个字符的字符串的 CLI 命令可以正常运行了。

### 901227 - Web 控制台里的 `node-timeout`, `worker-timeout`, `flush-wait` 和 `ttd mod_cluster` 属性问题

以前的 JBoss EAP 6 版本里基于 Web 的管理控制台里存在一个问题，它阻止了用户查看或设置 `mod_cluster` 属性的值为 `-1`。

- `node-timeout`
- `worker-timeout`
- `flush-wait`
- `ttd`

### 1103747 - EAP 6.x 的管理控制台的 "Profiles" 标签页里无法列出所有的主题/队列

在以前的 JBoss EAP 6 版本里，用户发现他们无法在管理控制台的 Profiles 标签页里列出所有的主题和/或队列。可见的队列的最大数目是 8。本版本已经解决了这个问题，现在所有的队列都是可见的了。

### 1029687 - 登出加密管理控制台 (SSL) 时重定向到 HTTP 地址

在以前的 JBoss EAP 6 里，用户登出加密的管理控制台 (HTTPS) 时会被错误地重定向到 HTTP 地址且登出会失败。

这是因为重定向到 HTTP 地址是硬编码的。

在这个版本里，我们更新了重定向，使其考虑用户是通过 HTTP 还是 HTTPS 访问，从而正确地进行重定向。

### 1048211 - 安全域在管理控制台显示错误的安全策略

以前的 JBoss EAP 6 版本里一个程序错误导致了安全域显示错误的安全策略。

本版本的 JBoss EAP 6 已经修复了这个问题，相关细节现在可以根据所选的安全策略而更新了。

### 999813 - [可用性] 当在管理控制台里冲刷 JMS 目的地度量时 Cancel 操作被忽略

在 JBoss EAP 6.1.0 和 6.1.1 里，在通过 Web 管理控制台冲刷 JMS 目的地之前，会显示一个确认对话框，要求对这个行为进行确认。而不管用户的选择是什么，JMS 目的地都会被冲刷。其根本原因是确认的结果被忽略了。我们已经解决了这个问题，现在 JMS 目的地只有在点击“OK”关闭确认对话框后才被冲刷。

### 1012490 - 在 Runtime Metrics 页面列出了消息目的地所有的 JNDI 名称

在以前的 JBoss EAP 6 里，我们发现消息目的地 (Runtime > JMS 目的地) 的 Runtime Metrics 页面只列出 JMS 目的地的单个 JNDI 名称。JMS 目的地的其他 JNDI 名称被省略号 ("...") 代替了。

这意味着 JMS 目的地的完整 JNDI 名称列表只能通过访问 EAP 配置集里的完整目的地列表来获取。

在这个版本里，所有 JMS 目的地的 JNDI 名称的完整列表在消息目的地的 Runtime Metrics 页面里借助信息提示的方式可见。

如果 JNDI 名称列表太长，它将被截取并显示省略号，但当鼠标悬停于 JNDI 名称条目之上时信息提示会显示完整的列表。

### 1073537 - 所有服务器都要求通过 DS 连接测试

在以前的 JBoss EAP 6 版本里，DS 连接测试使用了域里所有的服务器。这会对性能造成负面影响。

在本版本里，只有当前选择的配置集里的服务器会用于连接测试，这会产生更好、更可靠的性能。

## Web Services

### 1060001 - Spring 依赖关系注入在端点实例里无法工作

以前的 JBoss EAP 6 版本里的一个程序错误阻止了 JBoss WS-CXF 里创建的 bean 注入到端点，本版本通过组件升级解决了这个问题。

### 1032593 - 在应用程序卸载后数据仍保留在 jboss-eap-6.2/standalone/data/wSDL 里

即使应用程序从 JBoss EAP 服务器卸载，以前的 JBoss EAP 6 仍将数据保存在 Web Services Description Language (WSDL) 目录里 (EAP\_HOME/standalone/data/WSDL)。

只要在没有设置自动清理文件的服务器上卸载应用程序，WSDL 数据就默认会被保存。

我们通过更新资料库 (jbossws-cxf-4.2.x 和 trunk) 里的文件, 修改维护和发布 WSDL 日志的行为, 从而解决了这个问题。

这阻止了在 JBoss EAP 6.3 服务器里卸载应用程序后将数据保存在 WSDL 目录里。

### 1032439 - javax.management.InstanceNotFoundException: jboss.ws:service=ServerConfig 引起 jbossws testcase 间歇性地失败

在以前的 JBoss EAP 6 里, WS 服务器配置并不总是通过 JMX 可用。

这是由于已舍弃的对 WS ServerConfigService 里 MBean 服务器的 OPTIONAL 服务依赖关系引起的。

在这个版本里, 每当 JMX 子系统可用时, MBean 服务器依赖关系被设置为 REQUIRED 而不是 OPTIONAL。

因此, 当 JMX 子系统可用时, WS 服务器配置总是通过 JMX 可用。

### 1069349 - CXF 里的 Schema 导入在用于获取它们的 URL 里可能出现名称冲突

以前的 JBoss EAP 6 版本里的一个程序错误在导入 CXF 里的 Schema 时导致名称冲突。本版本已解决了这个问题。

### 900634 - JBossWS-CXF 没有发送错误消息到 FaultTo 端点

如果 WS-Addressing 的 **FaultTo** 元素被设置为 WS 客户, WS 服务器不会发送错误消息给 **FaultTo** 目的地。然而, 如果设置了 **ReplyTo** 元素, WS 服务器会发送响应到 **ResponseTo** 目的地。本版本的 JBoss EAP 6 带有更新的 Apache CXF, 从而修复了这个问题。

### 1040732 - JAXBDataBinding 无法处理带有普通对象的异常, 如 ObjectWithGenerics<Boolean, Integer>

在以前的 JBoss EAP 6 版本里, 当 **Exception** 类包含了一些带有下列展示的类型参数的成员时会出现问题。

我们合并了从 Exception 类生成的 WSDL, 现在 SOAP 错误信息不会再出现了。

```
@javax.xml.ws.WebFault
public class GenericsException extends Exception {
    private static final long serialVersionUID = 1L;
    private ObjectWithGenerics<Boolean, Integer> obj;

    public ObjectWithGenerics<Boolean, Integer> getObj() {
        return obj;
    }
    public void setObj(ObjectWithGenerics<Boolean, Integer> obj) {
        this.obj = obj;
    }
}
```

我们在上游开发里解决了这个问题并合并到了本版本里。

### 1079084 - Webservices DUP 没有根据 @WebService 注解扫描所有可见的类

我们发现在以前的 JBoss EAP 6 版本里, Webservices DUP 有一个程序错误, 它没有扫描所有可见的

用 `@WebService` 注解的类。当包含 `web.xml` 和引用 JAX-WS 端点（类用 `@WebService` 注解）的 `<servlet-class>` 的 `war` 归档位于 `ear` 归档里且包含这个类的 `jar` 位于 `ear` 归档的 `lib/` 目录里时会出现问题。我们已经在这个版本里修改了相关的代码，以更彻底地扫描用 `@WebService` 注解的类，错误信息也不会再出现了。

#### 1077259 - 对于带有 WSA ReplyTo prop set 的请求，没有创建 HttpServletRequestSnapshot。

对于启用了 WS-Addressing 的请求，如果也指定了 ReplyTo header 属性为非通用的地址，以前的 JBoss EAP 6 版本里的一个程序错误阻止了 `HttpServletRequestSnapshot` 的创建。

这个程序错误导致了立即的 HTTP 202 响应且容器过早地回收了 servlet 请求。这阻止了端点访问 servlet 请求的上下文信息。

通过升级 CXF 组件，这个版本已经解决了这个问题。

#### 1079043 - 当 JAX-WS client 从 JAX-WS 端点实现里调用时会丢失 MessageContext

在以前的 JBoss EAP 版本里，当 JAX-WS 客户在端点里被调用时，端点的 `MessageContext` 从 `ThreadLocal` 里被删除且在客户调用结束后无法被替换。

这导致在进行任何 JAX-WS 调用后 `MessageContext` 对于端点不可用。

这个版本已经解决了这个问题。

#### 1031642 - WebServices 子系统的 modify-wsdl-address 属性的 false 值被忽略了

以前的 JBoss EAP 6 版本里有一个程序错误，在执行 WSDL soap:address 重写时它阻止了 WS Stack 处理 `@WebService(wsdlLocation=...)`。

当注解只被置于服务端点接口时，上面的注解属性没有被处理。

这个版本已经解决了这个问题。

#### 1060355 - Schema 检验 + 多个 schema 导入到相同的命名空间 + 类别查找无法运行

通过升级 CXF 组件，我们解决了一个导致

`org.apache.cxf.wsdl.EndpointReferenceUtils.SchemaLSResourceResolver#resolveResource` 里的算法没有正确解析 schema 并只返回它发现的第一个 schema 的问题。

### jbossas

#### 1067620 - 当启用 Java Security Manager 时无法修改 EAP 6 上应用程序的权限

我们发现了一个应用 JSM 的问题，已部署的应用程序被赋予 `AllPermission` 权限，这和策略文件是矛盾的。这个问题的根本原因是没有启动虚拟文件系统（VFS），所以基于 `vfs:/...` URL 的策略不会被加载并只会应用默认权限。这个问题已被解决，通过确保 VFS 模块的加载，现在 JSM 策略可以正确地应用了。

### mod\_cluster

#### 1008901 - 有些重要的日志消息没有 ID 来进行标识

在以前的 JBoss EAP 6 版本里，我们发现有两条日志消息没有正确地本地化。

因此，用户无法看到 MODCLUSTER{id} 消息或者翻译。

在这个版本里，我们添加了两条消息：MODCLUSTER000044 和 MODCLUSTER000045。

这些消息现在可以正常显示了。

### 1030965 - 注册的上下文的数量对 mod\_cluster 的性能有负面的影响

我们在将 mod\_cluster 配置为负载均衡器的 Apache HTTP 服务器上确定了一个性能问题。workers->nodes 表上的 httpd 共享内存操作对负载均衡器性能有着负面的影响。因此，当注册的上下文增加时，httpd 负载均衡器的性能会降低。

要绕开这个问题，请试图减少注册的上下文的数量。

为了修复这个错误，我们修改了 httpd 来利用本地内存而不是共享内存。

### 1020142 - modcluster 子系统的 XSD 没有指定 worker-timeout 属性

在以前的 JBoss EAP 6 版本里，用于检验的 mod\_cluster XSD 没有指定 worker-timeout 属性。

这意味着使用 XSD 来检验配置可能会失败，即使配置是正确的且服务器正确地进行了解析。

我们已修复了 XSD schema，在使用 worker-timeout 属性时用 XSD schema 可以通过检验了。

### 1058334 - 服务器日志里没有解析 mod\_cluster 的 \${project.version}

在以前的 JBoss EAP 6 里，版本字符串逻辑使用 ModClusterLogger.java 里的 \${project.version}。

因此，\${project.version} 被写入了服务器日志。

在这个版本里，我们在 Version.properties 里添加了一个版本字符串并在登记开始信息前读取它来修复这个逻辑。现在服务器日志里已可以正确显示版本了。

### 985101 - STOP-APP 及 ENABLE/DISABLE-APP 没有包括在 mod\_cluster-manager 页面上

以前的 JBoss EAP 6 版本里没有开放 mod\_cluster-manager 页面上 STOP-APP 命令。

这个版本已经解决了这个问题，mod\_cluster-manager 页面上的 STOP-APP 命令是可用的了。

### 980246 - mod\_cluster-manager 可能将单个 VirtualHost 的别名显示多次，造成页面乱码。

在以前的 JBoss EAP 6 里，有用户报告当部署多个带有独立虚拟服务器的应用程序、且每个虚拟服务器都有多个别名时，mod\_cluster\_manager 可能错误地多次显示系统的虚拟主机（每个别名一次）。

我们已经解决了这个问题，现在所有的虚拟机主机及别名在 manager 页面上都只显示一次。

### 1098576 - ModClusterService stop 命令总是消耗会话

在以前的 JBoss EAP 6 版本里，使用 CLI 界面里的 ModClusterService stop 或 stopContext 命令无法在消耗活动会话后移动上下文至 STOPPED 状态。这意味着这些命令无法按需要（无需消耗活动）快速停止上下文。通过升级 mod\_cluster 组件，我们解决了这个问题。

### 1044594 - [WFLY-2663] mod\_cluster 度量属性从未被应用到类里

在以前的 `mod_cluster` 子系统里，配置的加载度量以及指定的属性会从 XML 里解析，但并未应用到类里。

这意味着配置加载度量的属性不会有任何作用。

在这个版本里，属性已经如预期地应用到对象上了。

### 1052185 - MODCLUSTER-365: Reset MCMP 被发送到所有可用的代理

以前的 JBoss EAP 6 版本里有一个问题，它导致 `mod_cluster` 发送 Reset MCMP (Mod\_Cluster Manangement Protocols) 到它的 `proxy-list` 里的所有 `httpd` 服务器（在其中一个服务器重启后）。这个行为可能对设置 `auto-enable` 上下文为 `false` 的系统有负面的影响。

正确的行为是只发送 Reset 请求到重启的服务器。这个问题的出现是因为 `DefaultMCMPhander.status` 调用了默认发送到所有代理的 `sendRequests`。

通过升级 `mod_cluster` 组件，这个问题已得到解决。

## 7.2. 增强

### 群集

#### 1025023 - 添加配置选项来启用/禁用 Infinispan 统计

这个 JBoss EAP 6 版本包含了对群集系统的增强。Infinispan 子系统可以自动收集使用统计信息。您可以针对每个缓存容器及每个缓存显性地启用或禁用统计。

这些统计信息有助于配置和调试应用程序和缓存，然而它们会增加少许负荷。默认情况下它们是被禁用的，可以按需要启用。

### 域管理

#### 1052187 - 使 `auditlog syslog` 的 `FACILITY` 可以在管理模型里进行配置

在以前的 JBoss EAP 6 版本里，审计日志 `syslog-handler` 的 `Facility` 在管理控制台里是不可配置的，默认的 `facility` 将被使用。本版本的 `facility` 将作为 `core-service=management/access=audit/syslog-handler=* 资源的一部分开放，可以按照需要进行配置。`

#### 1053355 - 使 `auditlog syslog` 的 `appName` 可以在管理模型里进行配置

以前的 JBoss EAP 版本不允许在管理模型里配置 `appName`。

因此将使用默认的 `appName`。

在这个版本里，`appName` 已经作为 `core-service=management/access=audit/syslog-handler=* 资源的一部分开放，且是可配置的了。`

#### 976228 - 需要重载服务器的操作应该检查是否有改动配置

在以前的 JBoss EAP 6 版本里，某些没有有效地修改服务器的状态的操作会将服务器置于需要重载的模式。

这是因为系统没有检查操作是否有效地修改了配置。

在这个版本里，如果确认了操作没有修改服务器的配置，重载就不是必需的。但这种修改并没有涵盖每种可能的情况。

### 1044772 - ldap 组加载没有记录任何日志

在以前的 JBoss EAP 6 版本里，我们没有添加 TRACE 和 DEBUG 日志到安全区的 LDAP 交互里。因为没有日志的存在，这使得诊断 LDAP 的验证问题极为困难。

在这个版本里，我们添加了详细的 TRACE 级别的日志到 `org.jboss.as.domain.management.security` 类别。由于 DEBUG 级别过于详尽，我们限制了这些新日志的级别为 TRACE。

客户现在可以使用这些日志来诊断和 LDAP 相关的问题。

## EJB

### 985793 - 允许 MDB 里通过 `ctivationConfigProperty` 注解进行属性替换

这个版本的 JBoss EAP 6 允许 MDB 里通过通过 `ctivationConfigProperty` 注解进行属性替换。

更多信息请参考 EAP 6.3 《开发人员指南》里的 Message-Driven Beans 章节。

## Hibernate

### 996767 - [HHH-8440] 在 Hibernate 里添加 `SQLServer2012` 方言

我们更新了 JBoss EAP 6 里的 Hibernate 组件来提供对 MySQL 2012 里引入的功能的支持。用户现在可利用一些功能，如创建 SEQUENCE 对象。

## HornetQ

### 1089853 - 对非事务性消息的发送实现计数器的同步

这个 JBoss EAP 6 版本增加了检查次数以确保页面计数器正确地同步。添加这个改进的原因是，我们发现在以前的版本里，页面计数器会和分页消息及保存在日志里的内容不同步，可能出现负数。额外的检查可以避免这种情况的发生。

## 安装程序

### 1035325 - 不要在控制台输出 `IzPack` 变量

在这个 JBoss EAP 6 版本里，安装程序不会再输出所有 `IzPack` 的内部变量到控制台上。它们只会登记在日志文件里。

### 1027655 - 允许用户为安全域配置多个模块选项

在以前的 JBoss EAP 6 版本里，只能添加单个登录模块到新的安全域里。

本版本对此进行了增强，允许创建任意个数的登录模块。用户现在可以在新的安全域里创建多个登录模块。



## 1048942 - 添加选项以自动配置 Maven settings 来使用公共资料库

在这个 JBoss EAP 6 版本里，我们已经重新配置了 Maven settings.xml 配置面板的用户界面。

默认界面更加直观，现在面板可以显示更清楚的信息了。

## JMS

### 1019069 - Messaging XSD 缺少了属性的描述

在这个 JBoss EAP 6 版本里，用于 Messaging 子系统的 XML Schema 用属性和类型的描述来注解。这个增强的目的是帮助开发人员编写对应的 XML 配置。

### 1014099 - 启动时出现 WARN HQ222180: There are no queues bound into Dead Letter Address jms.queue.DLQ. Messages will ...

这个 JBoss EAP 6 版本增强了 Messaging 子系统配置，现在它包含了 Dead letter 队列和 Expiry 队列。

在以前的版本里，默认的地址设置会发送过期的消息到 Expiry 地址，而不可递送的消息将被发送到 Dead Letter 地址。然而，这些地址没有绑定队列，这导致消息丢失，所以管理员无法检查这些消息及其他潜在的问题。

现在，过期的消息将保存在 DLQ Queue 而未递送的消息将保存在 ExpiryQueue 里。

## 日志

### 994661 - 自定义日志格式器

这个 JBoss EAP 6 版本让用户可以创建自定义日志格式器，用于使用命令行界面的处理程序或 XML 配置。处理程序现在覆盖了 `setFormatter(...)` 并使用自定义的格式器。

## PicketLink

### 1084594 - 向后移植 PLINK-313

这个版本的 JBoss EAP 6 里的 PicketLink 组件已经引入了新的功能。现在可以配置 IDP 来签注 assertion。

### 1084601 - 向后移植 PLINK-405

这个版本的 JBoss EAP 6 里的 PicketLink 组件已经引入了新的功能。发送给 AttributeManager 的 principal 现在是可配置的。

当 PicketLink 和 JBoss Negotiation 一起使用时，发送给 AttributeManager 的 principal 是一个杂乱的字符串而不是用户名。如果 Mapping 提供者使用这个 principal 来查找任何属性，这会导致查找失败。

通过配置 principal 可以改善这个问题。

### 1084589 - 向后移植 PLINK-146

这个版本的 JBoss EAP 6 里的 PicketLink 组件已经引入了新的功能。XMLSignatureUtil 现在允许 KeyInfo 在需要时使用 X509。

### 1085534 - 先后移植 PLINK-407

本版本的 JBoss EAP 6 引入了上游的 PicketLink 功能，它允许在 `IDPWebBrowserSSOValve` 里请求里使用 `FormAuthenticator characterEncoding` 参数。

### 安全性

这个版本的 JBoss EAP 6 添加了允许用户提高配置文件里使用的密码和其他敏感性字符串的安全性的功能。

这些功能包括使用自己的 `SecurityVault` 实现来遮盖配置文件里的密码和其他敏感属性，以及对 CLI 控制台配置文件 (`jboss-cli.xml`) 使用的 SSL 连接的密码库支持。

关于使用这些新功能的文档，请参考《安全指南》。

### Web

#### 900786 - 从 socket-binding 添加 redirect-port 到 web 连接器的配置里

6.3 以前的版本无法使用 `socket-binding` 来定义 `redirect-port`。

这个版本添加了相关功能，现在用户可以使用 `connector` 元素的新属性 `redirect-binding` 来重定向端口而不是使用 `redirect-port` 属性。这个新属性以套接字绑定的名称为值。

### Web Services

#### 1088916 - 对 WS-I Basic Security Profile 1.1 的支持

本版本的 JBoss EAP 6 现在支持 WS-I Basic Profile 1.1。

#### 1006638 - 对 WS-I Basic Profile 1.2 和 2.0 的支持

除了 WS-I Basic Profile 1.1，本版本的 JBoss EAP 6 现在还支持 WS-I Basic Profile 1.2 和 2.0。

### mod\_cluster

#### 987259 - 用于 mod\_cluster 的 hot-standby

本版本的 JBoss EAP 6 在 `mod_cluster` 里引入了 `hot-standby` 功能。

## 7.3. 已知的问题

### CDI/Weld

#### 1054876 - 当超类没有无参数构造器时创建代理失败

CDI 定义类必须有一个公共的构造器，要么用 `@Inject` 注解，要么就没有参数。Weld 在部署期间会检查这个要求。然而，本 JBoss EAP 版本里的 Weld 实现执行了更严格的检查，所有超类都必须有无参数的构造器。如果没有，部署会失败并抛出 `UnproxyableResolutionException`。

变通办法是在 Bean 类的所有超类里都包含一个公共的无参数构造器。

### 1086555 - Weld - 通用 Bean 的 Specialization 抛出不合适的异常

在这个版本的 JBoss EAP 6 里，设置通用 bean 的 specialization 会抛出 **DefinitionException**。

这可能并不合适，我们正在进行调查。

## CLI

### 1054874 - jboss-cli.sh 对 cygwin 的支持

本版本的 JBoss EAP 6 有一个程序错误，在 Cygwin 环境里使用产品时可能出现这个问题。

POSIX 和 Microsoft Windows 路径实现的不同可能导致 **jboss-cli.sh** 脚本里的路径变量无法正常解析。

我们期望在下一个版本里解决这个问题。

## 群集

### 900483 - 节点关闭使用 DIST SYNC 时接收到过时的数据

在测试过程中，当某个节点关闭且使用了 **DIST SYNC** 或 **DIST ASYNC** 缓存模式时，某些情况下会接收到过时的会话数据。我们仍在调查这个问题。

### 900378 - 带有 SET 复制触发器的 CDI bean 没有复制

由于 Weld 组件里的一个程序错误，**setAttribute** 方法没有被正确地调用。这导致了带有 **SET** 复制触发器的 CDI bean 复制失败。变通方法是使用 **SET\_AND\_NON\_PRIMITIVE\_GET** 触发器。我们将在以后的版本里解决这个问题。

### 900946 - IllegalStateException: Cache is in 'TERMINATED'/'STOPPING' state.

这个 JBoss EAP 6 版本带有一个程序错误，如果在某个节点上卸载应用程序后 5 秒内又部署到群集里的另一个节点上，这个错误可能会导致抛出 **IllegalStateException**。

我们正在进行调查这个问题，希望在下一个版本里解决它。

### 901162 - TimeoutException: Unable to acquire lock

这是本版本的 JBoss EAP 6 里的一个已知问题，它在某些情况下会抛出 **TimeoutException: Unable to acquire lock**。

我们期望在以后的版本里解决这个问题。

### 917635 - 无法加载会话 : NullPointerException

这是本版本的 JBoss EAP 6 里的一个已知问题，在某些情况下部署应用程序后它会导致 **NullPointerException** 并遇到 *Failed to load session* 信息。

我们期望在下一个版本里解决这个问题。

**993041 - org.jboss.as.web.session.ClusteredSession.access 里的 RuntimeException**

在罕见的情况下，当分布式 Web 会话被访问时，如果另外一个节点突然离开群集，获取锁可能失败。这种情况发生时会导致这样的异常信息：

```
RuntimeException: JBAS018060: Exception acquiring ownership of <session-id>
```

这个问题的根本原因是锁的获取并没有考虑群集节点可能恰好同时离开群集，导致获取锁失败。我们正在调查这个问题并期望在以后的版本里解决它。目前还没有变通方法。

**959951 - CacheException: java.lang.RuntimeException: Failure to marshal argument(s) at server shutdown**

服务器关闭时下列消息会被记录：

```
CacheException: java.lang.RuntimeException: Failure to marshal argument(s) at server shutdown
```

这个信息是因为 Infinispan 还不支持干净关闭，可被安全地忽略。我们正在调查这个问题，但目前还没有变通方法。

**922699 - IllegalStateException: AtomicMap stored under key X has been concurrently removed!**

在偶然的情况下，当过期和钝化的例行处理访问了并删除的会话时 `IllegalStateException` 会被抛出。这导致异常抛出并登记日志。

这个异常可以忽略。然而，禁用会话钝化可以避免这个问题。

**域管理****1111575 - subnet-match 完全无法工作**

这个版本的 JBoss EAP 6 的一个程序错误会阻止 `subnet-match` 解析 `value` 属性。

我们正在调查这个问题，期望在以后的版本里解决它。

**1105677 - 不存在的 ldap 组导致安全区里的验证失败**

当试图查找未存在于本地 LDAP 服务器上的组时，本版本的 JBoss EAP 6 里的一个程序错误会导致 `LdapGroupSearcher` 运行失败。在这些情况下，搜索会返回 **HTTP 500** 错误代码，阻止用户的登录。

我们期望在下一个版本里解决这个问题。

**1015524 - RBAC: 无法部署已经被不同 server-group 作用域的用户部署的相同部署**

当启用了基于角色的访问控制时，带有 `server-group` 作用域角色的管理用户可能无法部署新的部署，错误信息为：

```
"JBAS014807: Management resource '[("deployment\" => \"example.war\"]]' not found"
```

这是因为域里已存在具有相同名称的部署。这是正常的行为。然而，因为管理用户限定在某个服务器组的作用域里，它们无法查看部署是否已存在于另外一个服务器组上。这使得避免这种错误相当困难。

要绕开这个限制，我们推荐给负责添加部署到域里的用户分配无作用域的角色，或者维护一个部署名称列表以避免重复。

### 995439 - JTS transactions 设置依赖于 Jacorb 的设置

当在 Web 管理控制台 ("Profile -> Container -> Transactions") 的 **Transactions** 部分启用 JTS 时，您需要设置 JacORB 子系统里的属性 **transactions** 为 **on**。预期的行为是当启用 JTS 时事务也被启用。我们仍在调查这个问题产生的原因。在解决以前，变通方法是确保 **transactions** 属性也被设置为 **on**。请注意，事务子系统只在 **full** 和 **full-ha** 配置集里可用。

## EJB

### 990102 - 并发访问超时 -- 无法在 5000 毫秒内获取锁

这个 JBoss EAP 6 版本带有一个程序错误，当 EJB 客户在 "forwarding" 群集里调用 stateful bean 上的方法时，它可能导致 **concurrent access timeout**。这个 bean 将调用转发到 "target" 群集里的 stateful bean，然后再转回来。调用是有序的；客户在获得前一个调用的响应前无法调用新的方法。当群集里的其中一个服务器被关闭时，错误就出现了。

### 1118432 - 如果 remote-naming InitialContext 应被关闭，将抛出 `java.util.concurrent.RejectedExecutionException`。

这是本 JBoss EAP 6 版本里，如果使用远程命名来查找 EJB，它会创建一个 **ejb-client** 上下文。

而如果远程命名上下文对象被关闭时，客户将记录下面的错误，因为 **ejb-client** 上下文已经关闭。

```
ERROR [org.jboss.remoting.handler-errors] Close handler threw an
exception: java.util.concurrent.RejectedExecutionException
```

这个问题将在以后的版本里得到解决。

### 1067194 - EJB3 客户代码吞没了异常信息，使跟踪产品里的问题变得很困难。

本版本的 JBoss EAP 6 有一个程序错误，除非日志级别设置为 **DEBUG**，否则 EJB3 客户将吞没异常信息。

我们期望在下一个版本里解决这个问题。

### 1123505 - 如果连接使用相互验证，那 EJB/remoting 配置不会将证书作为凭证传播来进行验证。

在这个 JBoss EAP 6 版本里，如果连接使用相互验证，那 EJB/remoting 配置不会将证书作为凭证传播来进行验证。

这个问题使得无法将用于验证的 **BaseCertLoginModule** 和用 SSL 保护的 EJB 一起使用。

我们正在调查这个问题，期望在以后的版本里解决它。

## Hibernate

**1102974 - HHH-9231 如果事务性集合在冲刷后再初始化，回滚后未提交的数据仍会保留在集合缓存里。**

本版本的 JBoss EAP 6 有一个程序错误，它导致回滚后某些修改会保留在二级缓存里。

这个问题可能导致 `javax.persistence.EntityNotFoundException` 或遇到不存在的实体。

我们期望在下一个版本里解决这个问题。

**1111706 - NonUniqueDiscoveredSqlAliasException: Encountered a duplicated sql alias**

在这个 JBoss EAP 6 版本里，对简单实体（如 Event）的原生 SQL 查询，如果多次指定没有唯一别名的列（如 "select e.id, e.title, e.id from Event e"），会失败并显示如下信息：

```
org.hibernate.loader.custom.NonUniqueDiscoveredSqlAliasException:  
Encountered a duplicated sql alias [id] during auto-discovery of a  
native-sql query
```

这是因为 Hibernate 从 `java.sql.ResultSetMetaData.getColumnLabel()` 获取列标签。而 Hibernate 无法确定具有相同名字的列是否引用了相同实体里的相同值。

Hibernate 无法进行错误的假设，所以抛出这个异常。

变通方法是为具有相同名称提供唯一别名（如 "select e.id AS id1, e.title, e.id AS id2 from Event e"）。

**1094951 - HHH-9261 相同实体的多个独立的表示方式无法用 `cascade=merge` 进行合并**

本 JBoss EAP 6 里的一个问题可能导致 `java.lang.IllegalStateException: An entity copy was already assigned to a different entity` 错误。

在 `cascade` 合并过程中如果检测到多个具有相同标识符的实体实例时，这个问题就会出现。因为它们无法自动合并，所以抛出了异常。

我们期望在下一个版本里解决这个问题。

**1105078 - HHH-9062 Envers validity audit 策略：最终修订时间戳没有保存在联合子类的表里。**

在这个 JBoss EAP 6 版本里，当使用 `validity audit` 策略和 Enver 通过 `JOINED` 继承来审计实体层次结构时，为联合子类创建的审计表结果会不包含最终修订列。

这会使联合子类的审计表的分区变得复杂。

我们正在调查这个问题。

**1085982 - Envers: 审计表 (HHH-8908) 里缺少 `Embedded` 列**

这个版本的 JBoss EAP 6 的一个程序错误会阻止 Hibernate 里从超类继承的属性被审计，除非这个类用 `@Audited` 进行标记。

解决的办法是确保在跟踪类时使用 `@Audited` 属性。

我们期望在下一个版本里解决这个问题。

**1117447 - HHH-9280 `@ElementCollection` 的联合表明使用实体类名而不是指定的实体名**

这个版本的 JBoss EAP 6 的 Hibernate 组件里有一个程序错误，它导致根据 @ElementCollection 计算的表名使用实体类名而不是指定的实体名。

我们期望在下一个版本里解决这个问题。

### 900321 - HHH-7287 当使用 NaturalIdLoadAccess 获取 naturalIdQuery 结果时缓存 natural-id-values 出现问题

当从非事务性场景更新 **natural-id** 值之后试图从二级缓存 (L2) 加载实体时，Hibernate 接口 **NaturalIdLoadAccess** 的行为不一致。

当对数据进行事务和非事务的混合访问时，Hibernate 目前不支持具有 natural-id 的实体的 L2 缓存。

## 安装程序

### 1032892 - 对于 Windows 里以盘符开始的路径无法使用 Tab 填充

在 Windows 服务器上的 JBoss EAP 6 控制台安装程序里，以盘符开始目录路径的 Tab 完成功能无法工作。

当输入以盘符开始的路径时，例如 **C:\**，并按 **Tab** 键，不会显示任何东西。而在这种情况下，本应该显示该目录的内容。

这个问题将在以后的版本里解决。

### 977805 - Solaris 里无法从启动菜单里启动 JBoss 管理控制台

当 JBoss EAP 6 使用安装程序安装后，Solaris 上的 JBoss EAP 6 管理控制台的启动菜单快捷键无法使用。

不是在浏览器里打开管理控制台，而是显示下列错误信息：

```
Error showing url: HTTP Error: Method Not Allowed
```

这是因为 Solaris 上的 **xdg-open** 没有按预期地运行。

变通方法是直接通过浏览器打开管理控制台。如果配置了默认的端口，管理控制台的 URL 是：<http://localhost:8080/console>（独立模式）、<http://localhost:9990/console>（域模式）。

## JCA

### 1109691 - 当没有在 RA 配置里的 recovery 元素指定凭证时，Recovery 无法工作。

在这个 JBoss EAP 6 版本里，当没有在 XML 里指定 recovery 元素的安全性凭证时，对于某些 JMS 资源 Recovery 将无法启动。

因为应该可以无需凭证（或类似情况）创建连接，所以这种行为不是我们预期的。

我们期望在下一个版本里解决这个问题。

### 991389 - 在服务器用 JTS 事务模式集重载后 generic-jms-ra.rar 资源适配器里的连接工厂没有激活

当配有资源适配器的 JBoss EAP 服务器使用 `generic-jms-ra.rar` 部署时，服务器重载时连接工厂会撤销注册。其原因是如果配置事务子系统在 JTS 上运行，事务通用资源适配器无法正确处理重载。我们仍在调查这个问题的根本原因。

变通方法是在 RAR 归档里添加一个指定连接属性的 JCA 描述符 XML 文件。

### 1023193 - 在定期故障恢复过程中，JMS 通用资源适配器显示 'lock owned during cleanup'。

我们已确定在定期恢复里有个小问题，导致在日志里记录了以下列文本开始的信息。

```
WARN [org.jboss.resource.adapter.jms.JmsManagedConnection] (Periodic
Recovery) Lock owned during cleanup: java.lang.Throwable: Lock owned
during cleanup
```

底层原因是通用 JMS 适配器错误地处理了第二个定期恢复调用。根据 Object Transaction Service 规格，当设置事务管理者在 JTS 上运行时，定期恢复调用将用相同的 XID 回滚两次。通用 JMS 适配器错误地处理了相同 XID 的第二次回滚调用，所以出现了上述的警告信息。请注意，尽管有这样的警告信息，但恢复仍会正常进行。

### 1107120 - 部署 MySQL JDBC 驱动到 EAP 6.x 时失败

这个版本的 JBoss EAP 6 的一个程序错误在某些情况下会阻止 MySQL JDBC 驱动的部署。

部署失败时会提示部署已经存在。而试图删除现有的部署也会失败。

变通方法是重启服务器。然后驱动就可以成功部署了。

我们正在调查这个问题。

### 901300 - 群集 SFSB 上的 Datasource 字段的 @Resource 注入失败并显示序列化错误

本版本的 JBoss EAP 6 有一个程序错误，它导致群集 SFSB 上的 Datasource 字段的 @Resource 注入失败并显示序列化错误。

变通方法是将字段标记为临时的并手动在 @PostActivate 里再次填充。

我们将在下一个版本里解决这个问题。

### 1033008 - 通用 JMS 资源适配器和 EE 规格不兼容 - 当在事务上下文里创建会话时，它\*没有\*忽略参数。

当在事务上下文里创建会话且参数被传递到通用 JMS 资源适配器时，NPE (Null Pointer Exception) 将会发生。这是因为尝试处理参数时，当 Java EE 规格表明它们不会被处理。我们仍在调查这个问题的根本原因，目前的变通方法是像下面这个例子一样设置会话为有事务的。由此，NPE 不会再发生了。

```
connection.createSession(true, Session.SESSION_TRANSACTED);
```

### 1104227 - 当更多的数据源存在时，为用于恢复的 JDBC 驱动提供了不正确的数据源 JNDI。

在这个 JBoss EAP 6 版本里，当定义了多个数据源时，恢复过程中传递数据源的 JNDI 会出现问题。

这些数据源属于某台物理主机，但每个都指向不同的数据库（使用不同的用户进行连接）。

这个导致恢复失败并抛出异常，且显示下列错误信息：

-



```
WARN [com.arjuna.ats.jta] (Periodic Recovery) ARJUNA016027: Local
XARecoveryModule.xaRecovery got XA exception XAException.XAER_RMERR:
oracle.jdbc.xa.OracleXAException
```

我们期望在下一个版本里解决这个问题。

## JDR

### 917683 - Windows 里运行的 JDR 工具生成错误的归档条目名称

我们发现在 Windows 环境里使用 JDR 工具时有一个程序错误。它会将原始 \$JBOSS\_HOME 目录的最后一个字符附加到归档里创建的 JBOSS\_HOME 目录上。

例如，对于原始 JBOSS\_HOME 目录 *jboss-eap-6.2*，JDR 工具生成一个名为 *JBOSS\_HOME2* 的归档目录。

我们正在调查这个错误的原因，目前还没有变通方法。

## JPA

### 1114726 - 当其他子部署引用了实体类时，JPA 实体类的增强无法用于子部署。

这个版本的 JBoss EAP 6 的一个程序错误允许用户程序里的实体类被应用程序部署里的持久化提供者重写。

当并行部署子部署且在包含持久化单元的部署外的其他子部署里有对实体类的引用。在这些实例里，每个子部署都会经由不同的并行内部部署阶段。

我们期望在下一个版本里解决这个问题。

## JSF

### 1096905 - EAP 6.2 里切换到 JSF 1.2 时 WeldApplicationFactory 上出现 InstantiationException

JBoss EAP 6 允许用户通过部署捆绑的描述符选择部署的 JSF 版本。

然而，当部署也使用 CDI 时，它目前无法正确地工作。现在，如果两个不同的部署也使用 CDI，它们使用不同的 JSF 版本是不可能的。

当进行尝试时，异常将被抛出且部署失败并带有下列错误信息：

```
java.lang.InstantiationException:
org.jboss.as.weld.webtier.jsf.WeldApplicationFactory
```

虽然 JSF 1.2 和 CDI 的一起使用并没有官方支持，但是由于 JSF 1.2 是 EE5 的技术，而 CDI 是 EE6 的技术，采用下列变通方法，所有部署仍可能将 JSF 1.2 和 CDI 一起使用：

1. 在 *jboss-cli* 里执行下列命令将 JSF 1.2 设置为 EAP 里的默认版本：  
`/subsystem=jsf/:write-attribute(name=default-jsf-impl-slot,value=1.2)`
2. 通过修改

`/modules/system/layers/base/org/jboss/weld/core/main/module.xml` 和 `modules/system/layers/base/org/jboss/as/weld/main/module.xml` 里的 `<module name="javax.faces.api"/>` 行为 `<module name="javax.faces.api" slot="1.2" />` 为 Weld 模块配置 JSF 1.2 依赖关系。

3. 重启服务器。

## 日志

### 1096053 - SyslogHandler 没有正确处理多字节的字符

使用 `org.jboss.logmanager.handlers.SyslogHandler` 来登记多字节字符可能导致日志里的错误输出。例如：

```
Mar 8 17:29:09 UNKNOWN_HOSTNAME java[9896]: SyslogHandler: ??????????????
```

变通方法是用 `org.apache.log4j.net.SyslogAppender` 登记多字节字符。

这个问题部分是由于 UTF 转换算法引起的，我们期望在以后的版本里解决它。

### 1031448 - logging-profile 可用于 servlet，但不能用于 JSP

在这个 JBoss EAP 6 版本里，logging-profile 可用于 servlet，但无法用于 JSP。JSP 里的所有日志消息都会进入系统日志上下文。

这是因为被检查的类加载器是 `org.apache.jasper.servlet.JasperLoader`，其父类加载器在日志上下文里进行了注册。

我们期望在下一个版本里解决这个问题。

## 其他

### 901231 - IBM JDK 上使用 finalize() 需要更多的保护

当运行在 IBM JDK6 或 7 上时，JBoss EAP 6 可能遇到间歇性的故障。这是因为 IBM JDK 的垃圾回收比其他 JVM 要激进得多。这导致内存有时候会被意外地回收，而在其他 JVM 上测试时则注意不到这个问题。

在这个问题得到解决前，Red Hat 建议不要使用 IBM JDK6 或 7 来运行 JBoss EAP 6。

## 补丁

### 1120535 - 在关闭过程中查询 Patching MBean 导致 IllegalStateException

在本版本的 JBoss EAP 6 里，patching 子系统 PatchResource 虽然没有对 `InstallationManagerService` 的依赖关系，但在关闭时会试图使用它而导致了程序错误。

这可能导致当关闭 hook 试图查询 patching 子系统 MBean 时抛出 `IllegalStateException`。

解决的办法是在使用 `InstallationManagerService` 前先确认它是否已启动。

## PicketLink

### 1086733 - PicketLink 2.1 和 2.5 版本间的 API 修改

自版本 2.1 后, PicketLink 2.5 已经对 API 进行了一些修改, 这些都包含在以前的 JBoss EAP 6.x 发行版本里。

这些修改包括几个影响向后兼容性的类。PicketLink 2.1 应用程序需要相应地进行更新。

本文档里结尾提供了改动的完整列表。

## RBAC

### 1021607 - RBAC 的 non-addressability

某些资源对于作用域为服务器组和主机的角色是不可寻址的, 目的是提供管理模型的简化视图, 从而提高可用性。这和以不可寻址来保护敏感数据的资源是不同的。

对于作用域为服务器组的角色来说, 这意味着管理模型的 **profile, socket binding group, deployment, deployment override, server group, server config** 和 **server** 部分里的资源将是不可见的, 如果它们和为该角色指定的服务器无关的话。

对于作用域为主机的角色来说, 这意味着管理模型的 **/host=\*** 部分里的资源将是不可见的, 如果它们和为该角色指定的服务器无关的话。

然而, 在某些情况下, 这种简化的视图可以隐藏超出用户管理的范围的信息, 它可以为用户的一系列动作提供指导。一个例子是 [BZ# 1015524](#)。

在以后的版本里, 这些不可寻址的资源可能会改为可寻址但不可读的。这不会影响服务器的安全性, 因为出于安全原因它们是不可寻址的。红帽推荐您不要依赖于资源的不可寻址来隐藏信息, 除非这种不可寻址性是在敏感性约束里定义的。

## RESTEasy

### 899666 - RESTEasy: 空的 `cfg. param javax.ws.rs.Application` 导致异常

当 `WEB-INF/web.xml` 文件里的 `_javax.ws.rs.Application_` 配置选项为空时, 这个版本的 JBoss EAP 6 附带的 RESTEasy 组件抛出 `java.lang.StringIndexOutOfBoundsException: String index out of range: 0`。

这个行为是次优的, 我们期望在以后的版本里解决这个问题。

### 1088956 - Windows 上的命名空间测试抛出 `MalformedByteSequenceException`

在这个 JBoss EAP 6 版本里, 当在客户请求的主体里指定编码时, RESTEasy 会返回以服务器的编码而不是以原始请求编码为格式的响应。要以指定的编码接收响应, 您必须指定 `request.accept(mediaType)` 头部或使用 `@Produces` 注解。

我们期望在下一个版本里解决这个问题。

### 1078186 - 尽管在成功验证后连接已关闭, 重用的连接会处于不一致的状态。

在这个 JBoss EAP 6 版本里，当多次发送某个客户请求时，底层的 `httpClient` 没有正确处理连接。RESTEasy 使用 JBoss EAP 提供的 `org.apache.httpcomponents` 模块（目前版本是 4.2.1-redhat-1）。4.2.6 版本的 `org.apache.httpcomponents` 会修复这个问题。

#### 1122779 - DataSourceProvider 没有删除它创建的临时文件

当读取超过 4096 个字节时，`DataSourceProvider` (`org.jboss.resteasy.plugins.providers.DataSourceProvider.readDataSource(InputStream, MediaType)`) 会创建一个名为 `resteasy-provider-datasource ... .tmp` 的临时文件。然而，它不会删除这个临时文件也不会提供删除它的工具，这会导致资源的溢出。我们仍在调查这个问题并期望在下一个版本里解决它。

#### 1080420 - DataSourceProvider 使用了不安全的方法来读取输入流

RESTEasy 使用 `InputStream.available()` 来确定它是否完成了从客户读取输入流

RESTEasy 有时候不能完整地为客户读取输入数据。

我们期望在下一个版本里解决这个问题。

#### 899664 - (EAP 6.4.x) RESTEasy: Boolean 配置参数没有拒绝无效的内容

这个版本的 JBoss EAP 6 的一个程序错误允许设置无效的 Boolean 配置参数。这个行为是计划外的，无效的参数应该被拒绝且应用程序也不应该部署。

我们期望在下一个版本里解决这个问题。

## RPM

#### 1124516 - RPM 里缺少 `snmpd.conf.sample` 文件

下列版本的 `mod_snmp` RPM 里缺少 `httpd/conf.d/snmpd/snmpd.conf.sample` 文件：

- `mod_snmp-2.4.1-7.GA.ep6.el5.i386.rpm`
- `mod_snmp-2.4.1-7.GA.ep6.el5.x86_64.rpm`
- `mod_snmp-2.4.1-8.GA.ep6.el6.i386.rpm`
- `mod_snmp-2.4.1-8.GA.ep6.el6.x86_64.rpm`
- `mod_snmp-2.4.1-9.GA.ep6.el7.x86_64.rpm`

这个问题的 Bugzilla 里附加了一个 `snmpd.conf.sample` 示例文件，您可以按需要进行修改。

我们期望在下一个版本里解决这个问题。

## Remoting

#### 1104328 - 'Compatible Change' 编组不允许从序列化类的层次结构里删除类

如果序列化类的层次结构里缺少了某个类，在解组 (`unmarshalling`) 过程中可能遇到 `ClassNotFoundException`。

例如，如果某个系统编组类 A，它继承了类 B，而 B 反过来继承了类 C，当它们发送到没有在同一层次结构里包含这三个类的系统里时，会抛出关于缺少类的异常。

我们将在下一个版本里解决这个问题。

### 1089236 - AS 日子里的 JBoss Remoting 版本显示为 unknown

在这个 JBoss EAP 6 版本里，服务器日志里的 JBoss Remoting 版本显示为 (**unknown**)。

这是 Remoting 系统日志里的一个程序错误，但它对性能没有任何影响。

我们期望在以后的版本里通过组件升级解决这个问题。

## 脚本和命令

### 1069252 - add-user.sh 在 cygwin 里无法使用

在这个 JBoss EAP 6 版本里，`add-user.sh` 脚本无法运行并抛出 `IllegalStateException`。

当在 Cygwin 里执行 Java 进程时，因为没有可用的 `java.io.Console`，意味着没有 API 可以读取屏蔽的密码，从而出现了这个问题。

我们期望在下一个版本里解决这个问题。

## 安全性

### 949737 - NegotiationAuthenticator valve 中断了会话复制

在 `WEB-INF/web.xml` 文件里启用了 `NegotiationAuthenticator valve` 的应用程序里的会话复制无法运行。`NegotiationAuthenticator valve` 提供了自己的 `setNext()` 和 `getNext()` 方法，它们没有正确地添加群集会话 Valve 到 JBossWeb 请求管道里。

我们正在调查这个问题。

### 1103684 - 无法在数据源间分享标识符安全域

当定义了同一个安全域支持的多个数据源时，这个版本的 JBoss EAP 6 的一个程序错误会导致 `javax.resource.ResourceException: No matching credentials in Subject!` 错误。

我们正在调查这个问题，期望在以后的版本里解决它。

### 1052644 - LdapExtLoginModule 无法找到自定义的 LDAP 套接字工厂

在这个 JBoss EAP 6 版本里，`LdapExtLoginModule` 没有设置 TCCL 为（可配置）JBoss 模块的类加载器。这个 JBoss 模块将包含一个自定义套接字工厂。

因此，`LdapExtLoginModule` 无法使用自定义的套接字工厂来创建到 LDAP 服务器的连接。当试图使用自定义套接字工厂和 `LdapExtLoginModule` 时会抛出 `ClassNotFoundException`。

我们将在下一个版本里解决这个问题。

### 1097276 - 当获取连接时，MsSql 数据源抛出 IllegalStateException。

这个版本的 JBoss EAP 6 里的一个程序错误会导致在获取连接时 MySQL 数据源抛出 `java.lang.IllegalStateException: This ticket is no longer valid.`

我们期望在下一个版本里解决这个问题。

### 979369 - 通过程序调用 `login()` 时 `HttpSession` 的创建具有不同的行为

我们发现当使用程序登录时 `HttpSession` 的创建行为会不一致：

- 不使用 SSO：调用 `login()` 方法时不会创建会话 - 也就是随后的请求无法被验证；
- 使用非群集的 SSO：调用 `login()` 方法时会创建会话，但第一次调用不会设置验证状态 - 也就是随后的请求无法被验证；用户在第二次调用 `login()` 方法后被验证，因为会话已经存在了。
- 使用群集的 SSO：调用 `login()` 方法时会创建会话，且随后的请求会被验证。

我们正在调查这种行为不一致的原因。目前还没有变通方法。

### 1069326 - 需要 `login/logout` 审计事件

域管理请求是使用 `stageless` 协议来处理的。对于 HTTP，每次请求都会进行验证。对于原生验证，它是在连接建立时发生的。但是这里没有 *已验证的会话*。

因为没有 *已验证的会话*，`login` 和 `logout` 事件无法被审计。相反，当从用户接收操作时，审计消息被登记。

## 事务管理者

### 1124861 - 用 CMR 进行恢复时出现问题，虽然恢复正常但会依赖于 orphan 检测。

在这个 JBoss EAP 6 里，如果服务器在两阶段提交的 `prepare` 阶段崩溃，而 `Commit Markable Resource` 是 XA 资源的一部分时，XA 资源不会按预期地立即回滚。资源将保留在这个状态，直至启动了 orphan 检测。

我们期望在下一个版本里解决这个问题。

### 1017256 - 恢复进行时计算的事务统计不正确

这个版本的 JBoss EAP 6 里的一个程序错误会导致当处理有疑问的 `prepared` 事务时，如果使用了恢复，错误的事务统计信息将被显示。

在服务器崩溃前，以及在服务器重启后当恢复操作修复了有疑问的状态时，已处理的事务总数会不正确地增加。在这些情况下，事务会被统计两次。

我们正在调查这个问题，期望在以后的版本里解决它。

### 1077156 - 服务器绑定到 `::1` 时无法在 IPv6 上启动 XST 事务

在这个 JBoss EAP 6 版本里，服务器绑定到地址 `::1` 时无法在 IPv6 上启动 XST 事务。这是因为地址的一部分必须包裹在方括号里：如 `[::1]`。

我们期望在下一个版本里解决这个问题。

### 1113225 - CMR: 在崩溃后 TX 日志没有显示参与者

这个版本的 JBoss EAP 6 的一个程序错误会阻止在崩溃后在 TX 日志里显示参与者。

如果服务器在 `CommitMarkableResourceRecord.commit` 方法退出后崩溃（且在重启过程中定期恢复暂停），TX 日志将不会显示参与者。

我们正在调查这个问题。

## Web

### 1076439 - IBM 1.7 JDK 上的重载操作期间 NIO2 连接器没有正确地重载

如果这个版本的 JBoss EAP 6 在 IBM JDK 1.7 上启动并试图重载服务器，下列错误信息将出现在日志里。

```
ERROR [org.apache.coyote.http11.Http11NioProtocol] (MSC service thread 1-6) JBWEB003043: Error initializing endpoint: java.net.BindException: Address already in use
```

这个错误是由于 IBM JDK 里的一个问题引起的，它导致 NIO2 连接器没有正确重载，而原来的连接器继续运行。要绕过这个问题，请使用不同的连接器（注意每个连接器都有自己的功能）或不同的 JDK。

我们期望在下一个版本里解决这个问题。

### 918130 - JBossWeb 连接器在应用程序部署完成前启动

我们发现 JBossWeb 连接器在启动时有一个时机问题，连接器会在应用程序程序完全部署之前启动并接受请求，通过负载均衡器或直接连至 JBoss EAP 的客户连接会返回 **404** 消息。这个问题影响 JBoss EAP 6.0.1 和更高的版本。目前还没有变通方法，我们仍在调查这个问题。

## Web 控制台

### 1017655 - Web services 配置检验出错

在这个 JBoss EAP 6 版本里，Web 控制台不会拒绝使用 > Web > 配置集的 Web Services 里的 **WSDL Host**、**WSDL Port** 和 **WSDL Secure Port** 的无效值。

一些无效值会导致错误信息，而另外一些则被忽略且不会显示错误信息。这些无效值都没有保存在配置里。

我们将在下一个版本里解决这个问题。

### 1051991 - 管理控制台上没有显示 JMS 目的地的统计信息

这个 JBoss EAP 6 版本里的一个程序错误会阻止管理控制台显示 JMS 目的地的正确数量。数量总是显示为 **0**。

我们期望在下一个版本里解决这个问题。

### 1027586 - RBAC: Web 控制台对应用程序资源的控制过于粗颗粒度

在这个 JBoss EAP 6 版本里，当资源被定义为应用程序资源时，控制台可能不会体现这种定义。这是因为控制台经常在一个视图下对几个资源进行分组。如果相关的资源是可写的，视图里的控件就是可用的。然而，如果其中的任何资源被配置为应用程序资源，相关的控件会保持为禁用的。

目前的变通方法是，如果可能的话，配置所有和子系统相关联的资源类型为应用程序资源。

#### 1029851 - 管理控制台 - 即使在取消部署文件的上传后，其内容仍会被添加到 data 目录。

这是本 JBoss EAP 6 版本里的一个已知问题，它导致即使用户取消的文件的上传，这个文件仍会通过管理部署屏幕上传成功。

取消操作会导致上传的文件被添加到 data 目录，但也会阻止其添加为 `domain.xml` 文件的部署。这意味着文件无法通过 CLI 或 Web 管理界面进行删除。

我们将在下一个版本里解决这个问题。

#### 1026823 - 提供可靠的方法来清除 "Run as" 对话框设置的 "ROLE" 头部

在这个 JBoss EAP 6 版本里，当 Superuser 代替其他用户（通过 **Run As...** 链接）运行时，它们的角色在动作结束后没有重置。

变通方法是关闭浏览器窗口（而不只是活动的标签页）并重新访问 Web 控制台。我们将在下一个版本里解决这个问题。

#### 1014048 - RBAC: 在域模式下登出后再登录不会正确清除 UI

在某些情况下，当登出 Web 控制台时，在登录另外一个用户前控制台将部分地刷新。这会导致“混合”的屏幕内容，有些区域好像还是旧用户登录，而另外一些区域则是新用户登录的样子。

这个问题不是一个安全隐患，不会泄露敏感数据。

变通方法是关闭浏览器窗口（不仅仅是活动的标签页）并以新的用户登录。

## Web Services

#### 1111046 - 具有 WS-Security Policy Signature 配置集的端点抛出 SOAP Fault 时返回空的响应

本版本的 JBoss EAP 6 有一个程序错误，它导致在 WSDL 里具有 WS-Security Policy Signature 配置集的 Web 服务返回长度为零的空的 HTTP 响应，且抛出 SOAP Fault 而不是返回实际的 SOAP Fault。

这个问题只有在使用 Signature 策略时才会出现。

我们期望在下一个版本里解决这个问题。

#### 1079049 - @SchemaValidation 和 wsrn 1.1 一起使用时出现问题

如果客户发送引用 `CreateSequence` 的 WS-RM 1.1 消息到利用 WS-RM 和 schema 检验的端点，应用程序将抛出下列错误：

```
Unmarshalling Error: cvc-elt.1: Cannot find the declaration of element 'CreateSequence'.
```



`org.apache.xerces.impl.xs.XMLSchemaLoader` 里的 `fGrammarPool` 里未找到 `CreateSequence`, 因为 CXF 和 JBossWS 还未完全支持 WS-RM 1.1。

#### 1074368 - 从 exception 类生成的 Schema 无法识别 @XmlElement 注解

我们在这个 JBoss EAP 6 版本里找到一个程序错误, 从 exception 类生成的 Schema 无法识别 @XmlElement 注解。我们将在以后的版本里解决这个问题。

#### 1103847 - 当使用注解且没有独立的 Java 接口时 Java First WS-Policy 复制了 PolicyReferences 元素

JBoss EAP 6 附带的 CXF 组件里的一个程序错误导致了在使用 Java First WS-Policy 注解 (如 @Policies) 时 `PolicyReferences` 被复制。这使得在没有独立的 Java 接口时 WSDL 生成多个 `PolicyReference` 元素。

我们期望在下一个版本里解决这个问题。

#### 1121223 - [CXF-5892] 确保 EncryptedKey 元素引用它前面的 BinarySecurityToken

在这个 JBoss EAP 6 版本里, 当 WSS 时间戳没有包含在 SOAP 消息里时, CXF 将 `EncryptedKey` 元素引用的 `BinarySecurityToken` 放在 `EncryptedKey` 元素之后。

这可能导致期望 `EncryptedKey` 引用的 `BinarySecurityToken` 在 `EncryptedKey` 元素之上的接收者执行失败, 因为当他们在处理 `EncryptedKey` 元素时试图查找 `BinarySecurityToken` 但没有处理它。

我们期望在下一个版本里解决这个问题。

#### 1115215 - CXF-5679, CXF-5724 - WS-S 在升级后失败并抛出异常

`org.apache.ws.security.WSSecurityException: The signature or decryption was invalid`

CXF 2.7.10 里引入的 regression 运行失败, 错误信息

为: `org.apache.ws.security.WSSecurityException: The signature or decryption was invalid`

我们将在下一个版本里解决这个问题。

### mod\_cluster

#### 1085427 - StickySessions 无法用于未启用上下文的 ProxyPass

本版本 JBoss EAP 6 的一个程序错误阻止了 `StickySession` 用于未启用上下文的 `StickySession`。我们期望在下一个版本里解决这个问题。

#### 1044879 - 如果 Balancer 名称包含了大写字母, modcluster 不会保留 sticky sessions。

负载均衡器命名如果使用了大写字母, JBoss EAP 6 里的问题会阻止 `modcluster` 模块保留 `sticky session`。

这个问题导致特定会话的请求被导向到不同的节点, 而不是服务该会话的第一个请求的节点。

这是因为 JBoss EAP 6 不支持负载均衡器名称使用大写字母, 因为其名称和 URL 里的主机名类似。

如果模块里的所有负载均衡器名称都使用小些字母, 则 `modcluster` 模块可以保留 `Sticky Session`。

## 900047 - Linux/Unix 上的 SystemMemoryUsageLoadMetric 不正确

**SystemMemoryUsageLoadMetric** 在 Linux 或 Unix 操作系统上没有显示有用的信息。对于这些系统，**HeapMemoryUsageLoadMetric** 提供了更多有用的信息。这个问题的解决方案是修改 **SystemMemoryUsageLoadMetric** 的算法，从 **used** 数目里减去 **buffers/cache** 数目。我们仍在研究最佳的实现办法。

## Apache 服务器 (httpd) 和连接器

### 1125829 - 在 PPC64 架构的 RHEL6、RHEL7 上缺失了 HTTP 模块 mod\_snmp

这个 JBoss EAP 6 版本没有为 PPC64 平台 (RHEL6、RHEL7) 提供 mod\_snmp 库。

### 900620 - 如果使用 UAC 则不会加载 APR natives (Windows Server 2008 R2)

在 Windows 服务器上，如果启用了 UAC 则无法加载 Apache portable runtime (APR) 库。在安装了这些库后，管理员用户需要右击 JBoss 目录，并在 Properties -> Security 标签页里设置 "Full control" 为 "Allow"。这将纠正 UAC 的访问权限。

### 1125833 - RHEL7-x86\_64 平台的 ZIP 里缺失了 snmpd.conf.sample

RHEL 7 x86\_64 里的 Apache HTTP Server zip 归档里缺失了 snmpd.conf.sample。

变通方法是从 [https://bugzilla.redhat.com/show\\_bug.cgi?id=1125833](https://bugzilla.redhat.com/show_bug.cgi?id=1125833) 里的附件下载缺失的 snmpd.conf.sample。

### 1122799 - postinstall 脚本没有更新 mod\_snmp.conf.sample 里的路径

这是本版本里的一个程序错误，postinstall 脚本没有更新 mod\_snmp.conf.sample 里的路径。

安装后，路径仍然包含了 installroot 变量：

```
SNMPConf "@installroot@/etc/httpd/conf.d/snmpd"
```

变通方法是手动修改 mod\_snmp.conf.sample 文件。

我们会在下一个版本里解决这个问题。

## Apache 服务器 (httpd) 和连接器、文档

### 900580 - httpd: libaprutil-1.so.0: 文件太短

Red Hat 企业版 Linux 里的 JBoss EAP 或 JBoss 企业级 Web Server 并没有附带 Apache portable runtime (APR) 库。我们使用 Symlink 来引用基本的 Red Hat 企业版 Linux 库。如果没有安装下列软件包而又试图使用 APR 库时会导致 symlink 无法解析：**apr**，**apr-util**，**apr-devel**，**apr-util-devel**。

以后的 Red Hat 企业版 Linux APR 更新里会修复这个问题。

## CDI/Weld,EJB

### 971679 - 当卸载已经初始化 Interval EJB Timer 的应用程序时会出现问题

当关闭部署了使用 EJB Interval Timer 的 EAP 实例时，在包含 "Formatting error" 或 "Error invoking timeout for timer" 的服务器日志有时候会出现 `IllegalStateException` 或 `EJBComponentUnavailableException`。

这个问题的出现是因为 JBoss EAP 6 在关闭时不会等待所有的请求完成。

这个问题目前还没有解决办法，但是它并不会对产品有任何负面影响。

## EE,EJB

### 1114061 - ejb-jar.xml 的 interceptor-binding 不接受 "\*"

在本版本的 JBoss EAP 6，由于过于严格的检查，默认的 EJB 拦截器要求 *interceptors* 定义以及声明作为 *assembly-descriptor* 的一部分。

## EJB,Naming

### 923836 - 在 IBM JDK 里 remote:// protocol 无法使用 SSL 运行

这是本 JBoss EAP 6 的一个已知问题，在 IBM JDK 上远程命名查找会间歇性地失败并抛出下列异常：

```
org.jboss.remoting3.NotOpenException: Endpoint is not open.
```

当客户使用 IBM JDK 时会出现这个问题。当服务器使用 IBM JDK 但客户使用 JDK 时，这个问题则不会出现。

我们正在寻求解决这个问题的方案。

### 1069591 - 当查找远程队列时看到 ERROR: remote lookup Channel end notification received, closing channel Channel ID

当在 EJB 里的远程队列上执行查找时，虽然操作执行成功，但在上下文关闭后会显示下列错误：

```
Channel end notification received, closing channel Channel ID b8e969d6
(outbound) of Remoting connection 4970f4db to
DEVPC016970/10.23.132.245:5301
```

这个问题将在以后的更新解决。

## EJB, Transaction Manager

### 952746 - 修复在调用远程 EJB 资源时事务恢复操作失败的问题

在这个版本的 JBoss EAP 6 里，如果调用了可能已经崩溃的远程 EJB 资源，事务恢复操作可能会失败。

这是由于服务器和客户间的连接断开了（特别是当客户崩溃且重启），服务器和客户将不会彼此自动通讯。

在这种情况下，服务器无法知道客户已经重启，意味着 EJB 的事务恢复过程也无法知道要与哪个 EJB 节点通讯。

我们正在调查这个问题并寻求解决方案。

## 对 IPv6 的支持

### 900564 - 日志条目里的 IPv6 地址格式错误

IPv6 地址应该用方括号 ("[" 和 "]") 格式化，如 `http://[2620:52:0:105f::ffff:2]:9990`。JBoss EAP 6 的日志文件里缺失了这些括号。我们会在以后的版本里解决这个问题。

### 900609 - 当 Windows 服务器绑定 IPv6 时，IPv4 套接字仍会被打开。

如果 JBoss EAP 6 在 Windows 服务器上启动时禁用了 IPv4 栈并启用了 IPv6 栈，IPv4 套接字仍会被打开。我们仍在调查这个问题。

## Security, Web

### 952518 - Servlet 里 `init()` 和 `destroy()` 方法未使用 `run-as`

JBoss EAP 6 里存在一个问题，`run-as` 标识符未用于 `Servlet.init()`，这和 Java Servlet 2.4 规格是矛盾的。

这是因为与 JBoss EAP 6.5 一样，JBoss EAP 6 里不存在 `RunAsListener` 引起的。

要绕过这个问题，请在 `Servlet` 的 `init()` 方法里建立一个调用被保护的 EJB 的安全上下文。这可以用《JBoss Security FAQ》的 A10 和 A23 里描述的方法来完成：  
<https://community.jboss.org/wiki/SecurityFAQ>。

我们期望在下一个版本里解决这个问题。

## Web Console - UX

### 1016546 - RBAC: [可用性] 尝试配置 Auditor 角色为 Administrator 时显示不清楚的错误消息

在这个 JBoss EAP 6 版本里，如果用户试图执行他们并没有权限执行的操作，他们会看到下列错误信息：

**You don't have the permissions to access this resource!**

这样的信息会让用户感到困惑，因为他们只是执行某个操作，而不是访问某个资源。在以后的版本里我们会修改这个错误信息。

## XML 框架

### 1065128 - 处理大型文本节点时出现性能问题

我们在这个版本的 JBoss EAP 6 里发现了一个程序错误，当试图用 Xalan Transformer 转换大型文本节点时会导致性能问题。在使用 Transformer 来转换 `StreamSource` 为 `DOMResult` 时，Transformer 的性能会随着字符数据的大小增加而降低。这是一个已知的问题，我们将在以后的版本里解决。

## doc-Release\_Notes,JCA,JMS

**1002518** - 对于非事务性的操作，通用资源适配器不允许使用 TIBCO EMS 的 `XAConnectionFactory` 实例。

在这个版本的 JBoss EAP 6 里，当事务上下文里创建会话时（如 `connection.createSession(false,0);`），`IllegalArgumentException` 将被抛出。

这是因为通用的 RA 不会忽略传递到 `createSession` 的参数，其中第一个设置会话为非事务性的，而第二个设置为事务模式。

要避免这个问题，请创建一个带事务的会话：`connection.createSession(true,0);`。

## 8. PICKETLINK 2.1 和 2.5 版本间的 API 修改

下面是合并到 JBoss EAP 6.3.0 里的 API 修改的完整列表：

### API 的修改

`org.picketlink.identity.federation.core.config` 里的类移到了 `org.picketlink.config.federation` 里

`AuthPropertyType`

`ClaimsProcessorsType`

`ClaimsProcessorType`

`EncAlgoType`

`EncryptionType`

`IDPType`

`KeyProviderType`

`KeyValueType`

`MetadataProviderType`

`PicketLinkType`

`ProviderType`

`ServiceProvidersType`

`ServiceProviderType`

`SPTType`

`STSType`

`TokenProvidersType`

`TokenProviderType`

TrustType

**org.picketlink.identity.federation.core.config** 软件包里的类移到了  
**org.picketlink.config** 里

PicketLinkConfigParser

**org.picketlink.identity.federation.core.handler** 软件包里的类移到了  
**org.picketlink.config.federation.handler** 里

Handler

Handlers

**org.picketlink.identity.federation.core.parsers.config** 软件包里的类移到了  
**org.picketlink.config.federation.parsers** 里

SAMLConfigParser

STSCConfigParser

**org.picketlink.identity.federation.core.exceptions** 软件包里的类移到了  
**org.picketlink.common.exceptions** 里

ConfigurationException

ParsingException

ProcessingException

**org.picketlink.identity.federation.core.util** 软件包里的类移到了  
**org.picketlink.common.util** 里

Base64

PBEUtil

StaxUtil

StringUtil

SystemsPropertiesUtil

TransformersUtil

**org.picketlink.identity.federation.core.saml.v2.util** 软件包里的类移到了  
**org.picketlink.common.util** 里

DocumentUtil

[提交 bug 报告](#)

## 9. JBOSSWS-SPI 和 JBOSSWS-CXF 里的修改

本版本里的修改的目的是为了提高稳定性、在需要时防止并行访问出现问题并确保线程安全。

为了实现这一点，我们以向后兼容的方式重构了 SPI 里的一些类。

这些修改不会影响大部分用户，因为受影响的类是用来构建 Webservice 栈和应用程序服务器容器间的集成接口，而不是针对最终的用户使用的。

最终用户如果在程序里使用了 `org.jboss.wsf.spi.metadata.config.CommonConfig` 和 `org.jboss.wsf.spi.management.ServerConfig` 及其实现，他们只需要留意和移植相关的代码的改动。

下列的修改可能影响在程序里依赖于改动的 SPI 类的用户：

CommonConfig 现在是不可变的，因此下列方法已被删除。

- `public void setPostHandlerChains(List<UnifiedHandlerChainMetaData> postHandlerChain);`
- `public void setPreHandlerChains(List<UnifiedHandlerChainMetaData> preHandlerChains);`
- `public void setConfigName(String configName);`
- `public void setFeature(Feature feature, boolean enabled);`
- `public void setProperty(String name, String value);`

ServerConfig 有了注册和取消注册 CommonConfig 实例（在容器引导期间）的新机制，从而防止了在运行时进行不安全的修改。用户应该意识到下列方法已被舍弃。

#### 已舍弃的 ServerConfig 方法

- `public List<ClientConfig> getClientConfigs();`
- `public List<EndpointConfig> getEndpointConfigs();`

#### 新的 ServerConfig 方法

- `public ClientConfig getClientConfig(String name);`
- `public EndpointConfig getEndpointConfig(String name);`

#### 不影响普通用户的其他修改

- `org.jboss.wsf.spi.metadata.webservices.WebservicesMetaData`、`org.jboss.wsf.spi.metadata.webservices.WebserviceDescriptionMetaData` 和 `org.jboss.wsf.spi.metadata.webservices.PortComponentMetaData` 的构造器已进行修改，因为这些类的实例现在是不可变的。当使用传入 `WebservicesMetaData` 实例（可选）的 `org.jboss.wsf.spi.publish.EndpointPublisher` 时这是相关的。

[提交 bug 报告](#)

## A. 修订

**修订 6.3.0-24**

**Thursday August 7 2014**

**Sande Gilda**

Bugzilla Sync 08/01/2014 (2:21 pm) Brisbane Time (with manual edits).