

UniEAP V4.6 应用监控用户手册

更改履历

| 版本号 | 更改时间 | 更改的图表和 章节号 | 状态 | 更改简要 描述 | 更改申请编 号 | 更改人 | 批准 人 |
|-----|------------|---------------|-----|------------|------------|-----|---------|
| 1.0 | 2014-8-8 | | N | | | 王辉 | |
| 1.1 | 2014-8-14 | | A_M | | | 于明光 | |
| 1.2 | 2014-10-23 | | A_M | | | 王辉 | |
| 1.3 | 2014-10-22 | | A_M | | | 张冬威 | |
| | | | | | | | |
| | | | | | | | |

注：状态可以为 N-新建、A-增加、M-更改、D-删除

版权声明

《UniEAP V4 WorkShop 用户手册》的版权归东软集团（大连）有限公司所有。未经东软集团（大连）有限公司的书面准许，不得将本手册的任何部分以任何形式、采用任何手段（电子的或机械的，包括照相复制或录制）、或为任何目的，进行复制或扩散。

Copyright© 1997-2011 东软集团(大连)有限公司。版权所有，翻制必究。**Neusoft东软**[®]是东软软件股份有限公司的注册。

前言

*Gartner 在《应用性能监控魔力象限》分析报告指出 IT 运维管理日渐趋向于以应用为中心……大多数 IT 运维团队仍认为基础设施的管理工作(真实和虚拟服务器、网络 and 存储)是其
主要业务，但他们已经认可，基础设施只有通过支持应用才能为企业带来价值。*

面向的受众

在阅读本手册前请务必先阅读 UniEAP V4.6 用户手册的第二章 UniEAP WorkShop 的安装与配置。其次，本手册面向有一定 UniEAP WorkShop 开发经验的开发人员。

| | |
|--|----|
| 更改履历 | 1 |
| 版权声明 | 3 |
| 前言 | 4 |
| 面向的受众 | 4 |
| 第 1 章 UniEAP 应用监控简介 | 7 |
| 1.1 了解应用监控 | 7 |
| 1.1.1 应用监控的功能定位 | 7 |
| 1.1.2 UniEAP Platform 应用监控方案的特点与优势 | 7 |
| 1.2 UniEAP Platform 应用监控方案体系架构 | 7 |
| 1.2.1 基本元素概念 | 7 |
| 1.2.2 快速部署模式 | 8 |
| 1.2.3 可伸缩部署模式 | 8 |
| 1.3 监控数据存储 | 9 |
| 1.3.1 简要说明 | 9 |
| 1.3.2 应用监控数据存储方案 | 9 |
| 第 2 章 UniEAP 应用监控安装部署 | 11 |
| 2.1 部署规划 | 11 |
| 2.1.1 快速部署规划 | 11 |
| 2.1.2 可伸缩部署规划 | 12 |
| 2.1.3 推荐配置 | 12 |
| 2.2 安装过程指引 | 13 |
| 2.2.1 快速部署模式下的安装过程 | 13 |
| 2.2.2 可伸缩部署模式下的安装过程 | 14 |
| 2.2.3 监控代理的使用指引 | 16 |
| 第 3 章 监控总线与监控控制台使用说明 | 29 |
| 3.1 接入管理 | 29 |
| 3.1.1 应用注册管理 | 29 |
| 3.1.2 已订阅管理 | 32 |
| 3.2 实时信息 | 33 |
| 3.2.1 监控信息订阅 | 33 |
| 3.2.2 业务事务监控 | 34 |
| 3.2.3 SQL 语句监控 | 39 |
| 3.3 智能预警 | 47 |
| 3.3.1 预警项管理 | 47 |
| 3.3.2 告警历史查看 | 50 |

| | | |
|-------|-----------------------------|----|
| 3.4 | 外部资源..... | 50 |
| 3.4.1 | 资源管理..... | 50 |
| 3.4.2 | WebLogic DataSource 监控..... | 51 |
| 3.4.3 | WebLogic Jvm Heap 监控 | 52 |
| 第 4 章 | 应用监控日常运维..... | 60 |
| 3.5 | 应用监控启动..... | 60 |
| 3.6 | 应用监控停机..... | 61 |
| 3.7 | 监控数据管理..... | 61 |
| 附录 A | MongoDB 的安装部署 | 62 |

第1章 UniEAP 应用监控简介

1.1 了解应用监控

1.1.1 应用监控的功能定位

在生产环境下实时追踪应用运行轨迹、监控人机交互过程，辅助快速定位与分析系统故障与性能瓶颈，最终达到保证应用服务质量、提升用户体验的效果目标。

1.1.2 UniEAP Platform 应用监控方案的特点与优势

- **无侵入：**对目标系统透明，基本做到完全无侵入。
- **轻负载：**资源占用对生产系统影响可忽略，保证系统业务正常运行。
- **实时性：**在生产环境中实时测量，第一时间反馈监控结果，还原真实上下文场景。
- **可扩展：**面向服务架构设计，易扩展、可定制更多类型资源监控。

1.2 UniEAP Platform 应用监控方案体系架构

1.2.1 基本元素概念

整个应用监控体系中存在三种元素概念：

- **监控代理**
嵌入到被监控应用中，透明地收集应用运行时数据。足够轻量，对应用性能及资源占用方面的影响可以忽略。
- **监控总线**
发挥监控消息队列作用，负载高并发下的消息处理压力。通过消息发布订阅功能，支撑灵活、高可伸缩性的部署架构体系。
- **监控控制台**
真正处理监控消息的逻辑运算单元，包括：数据持久化、数据分析视图、监控指

标告警等功能。



图表 1-1 基本元素概念示意图

1.2.2 快速部署模式

针对小型的、数量很小的应用监控需求（被监控应用产生的监控数据并不过多，数据处理压力并不过大），建议采取将监控总线嵌入到监控控制台的部署形式（监控总线与监控控制台在同一个应用实例中）。



图表 1-2 快速部署结构示意图

可见，快速部署模式采用的是一种“点对点模式”，其优点在于部署灵活轻便，缺点是在处理高负载、高并发的消息请求时可能会出现瓶颈问题。

1.2.3 可伸缩部署模式

监控代理数量随着应用实例数量的变化而增减。而监控控制台也会随着运维工作的工作量、资源投入以及管理制度的变化而改变实例数量与权责划分。可以根据具体的工作需求，使用不同的监控控制台来区分监控不同的应用，或者共享相同的监控对象再细分复杂的运维职能。这种灵活的伸缩性与制度性支撑功能，将由监控总线扮演的消息中转路由角色来保障。



图表 1-3 可伸缩部署结构示意图

可见，可伸缩部署模式是一种“总线模式”，其优点是通过分布式部署监控控制台应用可以高效的处理高负载、高并发的情况，缺点是部署过程相对复杂。

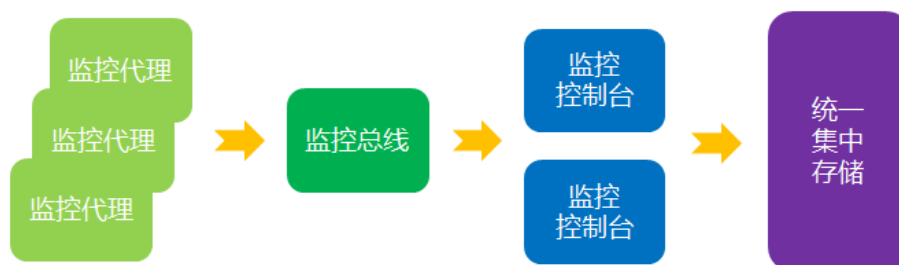
1.3 监控数据存储

1.3.1 简要说明

应用监控数据存储采用了 MongoDB 解决方案。考虑到数据压缩率、磁盘与内存成本方面的因素，准确地说是采用了 TokuMX 的数据存储解决方案。关于 TokuMX 的相关介绍请参考：<http://www.tokutek.com/tokumx-for-mongodb/>。

1.3.2 应用监控数据存储方案

1) 集中统一存储模式



图表 1-4 集中统一存储模式示意图

在集中统一存储模式下，所有的监控控制台均将接收到的监控数据存储到统一的 TokumX 存储介质中。此种模式通常用于多个监控控制台订阅同一个应用的场景。

2) 分散存储模式



图表 1-5 分散存储模式示意图

在分散存储模式下，监控控制台与 TokumX 存储介质形成了一对一的对应关系。不同的监控控制台将接收到的监控数据存储到不同的存储介质中，确保了数据的隔离。可见，此种模式通常适用于不同的监控控制台订阅不同的应用的场景。

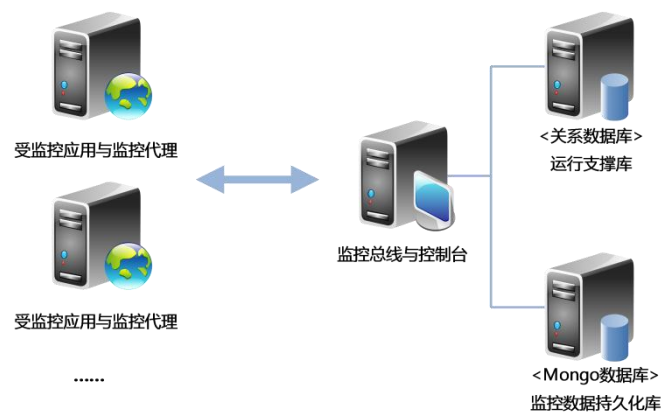
第2章 UniEAP 应用监控安装部署

2.1 部署规划

对应前文所述两种部署模式划分，部署规划也分为两种形式：“快速部署规划”与“可伸缩部署规划”。关于两种模式的选取，这里给出参考性建议：

- 若出于开发与调试目的，请使用快速部署模式
- 若被监控应用的数量少于三个，请使用快速部署模式
- 若运维工作制度更接近集中式管理，请考虑使用快速部署模式
- 不符合上述条件的，请考虑使用可伸缩部署模式

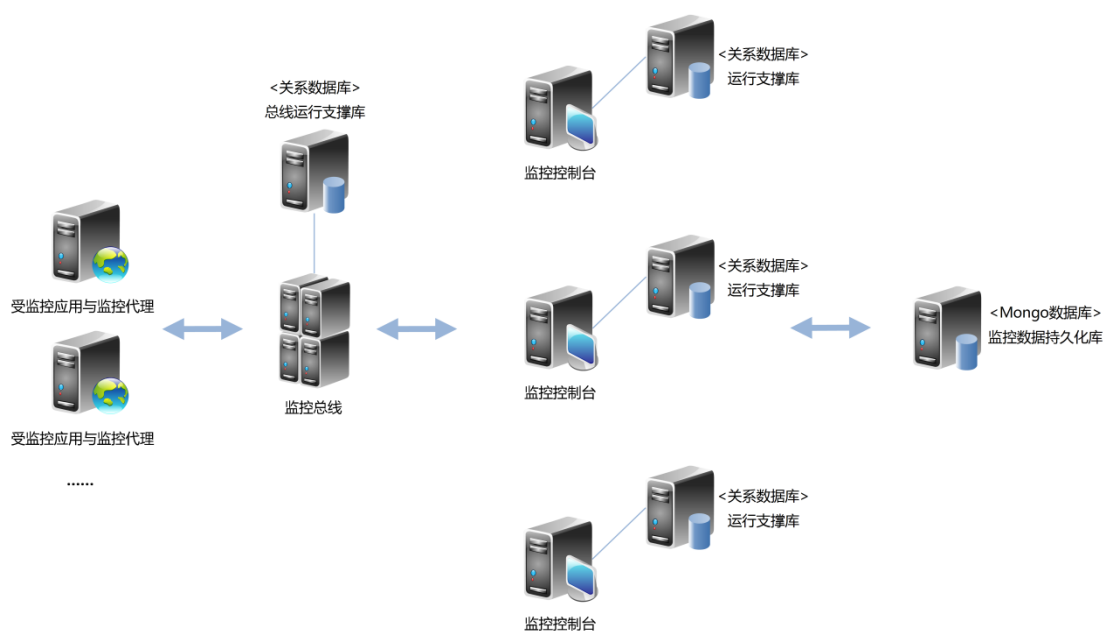
2.1.1 快速部署规划



图表 2-1 分散存储模式示意图

如上图所示，该模式下需保证每个“监控代理”与“监控控制台”间的双向网络联通。

2.1.2 可伸缩部署规划



图表 2-2 分散存储模式示意图

如上图所示，该模式下需保证每个“监控代理”与“监控总线”的双向网络联通；保证“监控总线”与每个“监控控制台”间的双向网络联通；并且保证对于每个监控控制台，Mongo 服务的可用性。

2.1.3 推荐配置

| 项目名称 | 推荐配置 |
|---------------------------------|----------------------------------|
| “监控总线应用”应用服务器 “监控控制台应用”应用服务器 | (免费开源) Tomcat (商业产品) Weblogic |
| 关系数据库 | (免费开源) MySQL (商业产品) Oracle |
| MongoDB 数据库宿主操作系统 | (免费开源) CentOS 64 位版本 |

在硬件方面，具体配置需视应用访问流量而定，原则上仅需 PC-Server 级别服务器即可。需注意的一点是，必须保证各相关节点间的网络联通性以及至少千兆的通信带宽。

2.2 安装过程指引

2.2.1 快速部署模式下的安装过程

下面介绍在快速部署模式下的“总线—控制台”应用的安装过程：

前提准备：

- **JAVA Web 应用服务器：**用于运行“监控总线&监控控制台”应用
- **关系数据库服务器：**为“监控总线&监控控制台”提供运行时支撑库服务
- **MongoDB（或 TokumX）服务：**为“监控总线&监控控制台”提供监控数据持久化服务

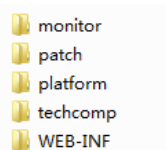
步骤 1：通过服务支持人员获取资料如下：

1. 应用部署包：“监控总线&监控控制台.zip”
2. 平台组件支撑库初始化脚本：“platform.sql”
3. 监控控制台组件支撑库初始化脚本：“platform_monitor.sql”
4. 监控总线组件支撑库初始化脚本：“platform_bus.sql”

备注：以上所列资料文件名称有可能会发生变动，以具体交付时为准，但资料数量与对应意义应该是一致的。

步骤 2：使用资料 2、3、4 初始化支撑数据库。

步骤 3：解压资料 1，解压后所得结果是一个标准的 Java Web 应用目录，如下图所示：



图表 2-3 监控总线&监控控制台应用部署包目录结构与内容

步骤 4：修改支撑库链接配置文件“WEB-INF\conf\platform\core\EAPConfig.properties”，使得应用启动时能够正确初始化支撑库连接。

将上述配置文件中的以下字段，修改为“步骤 2”中初始化好的支撑库对应值。

- unieap.datasource.jdbcUrl

- unieap.datasource.user
- unieap.datasource.password

步骤 5： 修改 MongoDB 链接属性配置文件“WEB-INF\classes\com\neusoft\unieap\monitor\repository\durability\mongo”，使得应用能够正确地将监控数据持久化。

修改上述配置文件中的“serverAddresses”字段，修改为“**前提准备**”中准备好的 MongoDB 服务对应值。

比如，若为单实例服务情形，则类似如下设置：

serverAddresses=10.4.46.13:27017

若为集群服务情形，则类似如下设置：

serverAddresses=10.4.46.13:27017,10.4.46.14:27017,10.4.46.15:27017

步骤 6： 保证支撑库服务与 Mongo 服务均可用的前提下，使用“**前提准备**”中准备好的应用服务器，启动已修改为正确配置的“监控总线&监控控制台”应用。

步骤 7： 安装完毕，访问应用服务器提供的地址即可。

2.2.2 可伸缩部署模式下的安装过程

下面介绍在可伸缩部署模式下的总线应用和控制台应用的安装过程：

前提准备：

- **JAVA Web 应用服务器：** 用于运行“监控总线”应用和“监控控制台”应用
- **关系数据库服务器：** 为“监控总线”应用和“监控控制台”应用提供运行时支撑库服务（注意：这里是指两个库实例）
- **MongoDB 服务：** 为“监控控制台”应用提供监控数据持久化服务

以下：步骤 1 至步骤 6，描述了“监控总线”应用的安装部署过程；步骤 7 至步骤 12，描述了“监控控制台”应用的安装部署过程。

步骤 1： 通过服务支持人员获取资料如下：

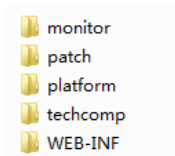
1. 应用部署包：“监控总线.zip”
2. 应用部署包：“监控控制台.zip”
3. 平台组件支撑库初始化脚本：“platform.sql”
4. 监控控制台组件支撑库初始化脚本：“platform_monitor.sql”
5. 监控总线组件支撑库初始化脚本：“platform_bus.sql”

备注：以上所列资料文件名称有可能会发生变动，以具体交付时为准，但资料数量与对应意义应该是一致的。

步骤 2： 使用资料 3、4 初始化“监控控制台”应用支撑数据库，使用资料 3、5 初始化

“监控总线”应用支撑数据库。

步骤 3: 解压资料 1，解压后所得结果是一个标准的 Java Web 应用目录，如下图所示：



图表 2-4 监控总线应用部署包目录结构与内容

步骤 4: 修改支撑库链接配置文件 “WEB-INF\conf\platform\core\EAPConfig.properties”，使得应用启动时能够正确初始化支撑库连接。

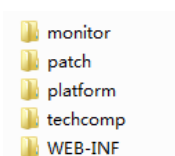
将上述配置文件中的以下字段，修改为“**步骤 2**”中初始化好的“监控总线”应用支撑数据库对应值。

- unieap.datasource.jdbcUrl
- unieap.datasource.user
- unieap.datasource.password

步骤 5: 保证“监控总线”应用支撑库服务可用的前提下，使用“**前提准备**”中准备好的应用服务器，启动已修改为正确配置的“监控总线”应用。

步骤 6: 至此“监控总线”应用已安装部署完成，访问应用服务器提供的地址即可。

步骤 7: 从该步骤起，以下所有步骤均为安装“监控控制台”应用的操作过程描述。解压资料 2，解压后所得结果是一个标准的 Java Web 应用目录，如下图所示：



图表 2-5 监控控制台应用部署包目录结构与内容

步骤 8: 修改支撑库链接配置文件 “WEB-INF\conf\platform\core\EAPConfig.properties”，使得应用启动时能够正确初始化支撑库连接。

将上述配置文件中的以下字段，修改为“**步骤 2**”中初始化好的“监控控制台”应用支撑数据库对应值。

- unieap.datasource.jdbcUrl
- unieap.datasource.user
- unieap.datasource.password

步骤 9： 修改 MongoDB 链接属性配置文件 “WEB-INF\classes\com\neusoft\unieap\monitor\repository\durablity\mongo”，使得应用能够正确地将监控数据持久化。

修改上述配置文件中的“serverAddresses”字段，修改为“前提准备”中准备好的 MongoDB 服务对应值。

比如，若为单实例服务情形，则类似如下设置：

serverAddresses=10.4.46.13:27017

若为集群服务情形，则类似如下设置：

serverAddresses=10.4.46.13:27017,10.4.46.14:27017,10.4.46.15:27017

步骤 10： 修改总线链接属性配置文件 “WEB-INF\classes\ config.properties”，使得 “监控控制台” 应用能够正确建立与 “监控总线” 应用之间的网络连接。

修改涉及的属性为：bus.location 属性与 controller.location 属性。bus.location 属性是指总线应用的可达访问地址，比如：

bus.location = http://10.4.46.21:8080/framework/feed

其中 “10.4.46.21:8080” 是 “总线应用” 的部署地址与服务端口，需设置为实际值；“framework” 是 “总线应用” 的上下文路径，需设置为实际值；“feed” 是固定的访问路径，不要改变。

controller.location 属性是 “监控控制台” 应用的可达访问地址，比如：

controller.location = http://10.4.46.14:8080/framework/controllerEntry

其中 “10.4.46.14:8080” 是 “监控控制台应用” 的部署地址与服务端口，需设置为实际值；“framework” 是 “总线应用” 的上下文路径，需设置为实际值；“controllerEntry” 是固定的访问路径，不要改变。

步骤 11： 保证 “监控控制台” 应用支撑库服务与 Mongo 服务均可用的前提下，使用 “前提准备” 中准备好的应用服务器，启动已修改为正确配置的 “监控控制台” 应用。

步骤 12： “监控控制台” 应用已安装完毕，访问应用服务器提供的地址即可。

2.2.3 监控代理的使用指引

阅读本章节之前请首先确定已阅读并理解 “UniEAP V4.6 用户手册”。

2.2.3.1 导入监控代理相关组件

目前，“监控代理” 的引用时机仅限于开发期，但这对于应用开发是透明的，并不会对应用开发造成侵入影响。

“监控代理”的形态是一个 UniEAP 应用开发组件，在应用开发期引入该组件的过程与普通的平台开发组件导入过程无异。

以下是“监控代理”组件导入过程的详细描述。

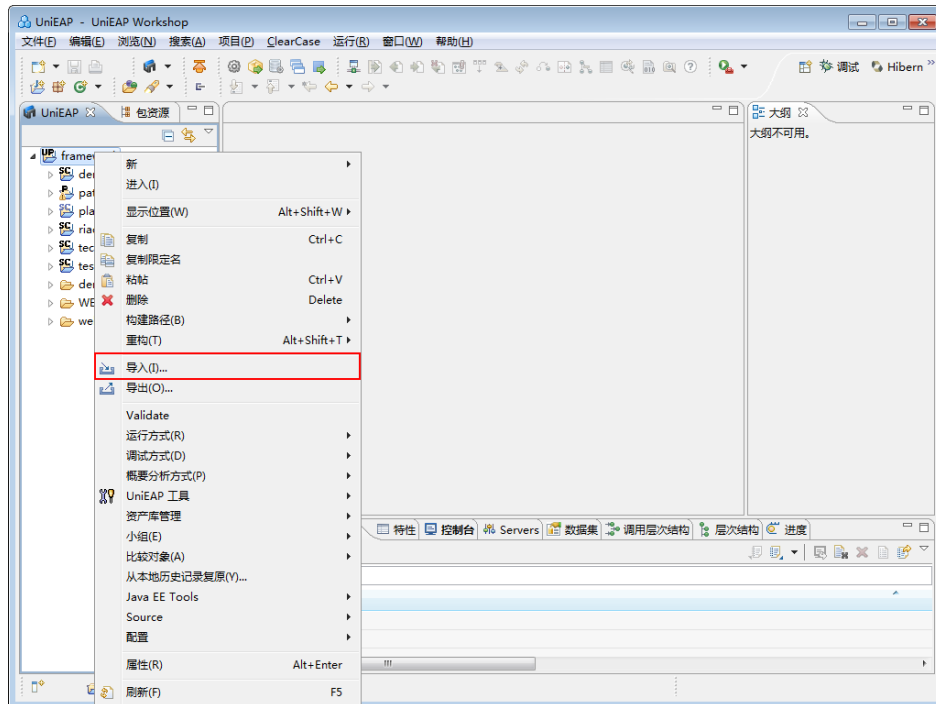
前提准备：

- UniEAP WorkShop
- 应用开发工程

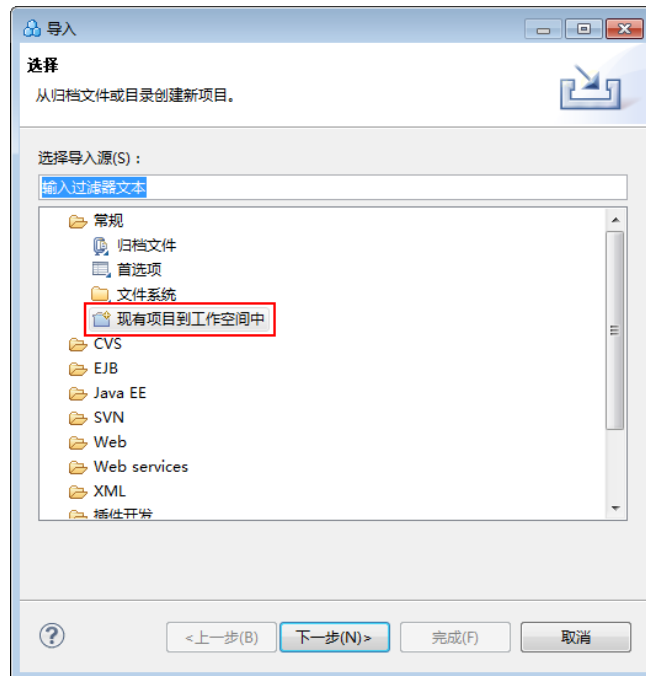
步骤 1: 通过服务支持人员获取资料如下：

1. 软件组件 monitor SC
2. 开发组件 agent DC

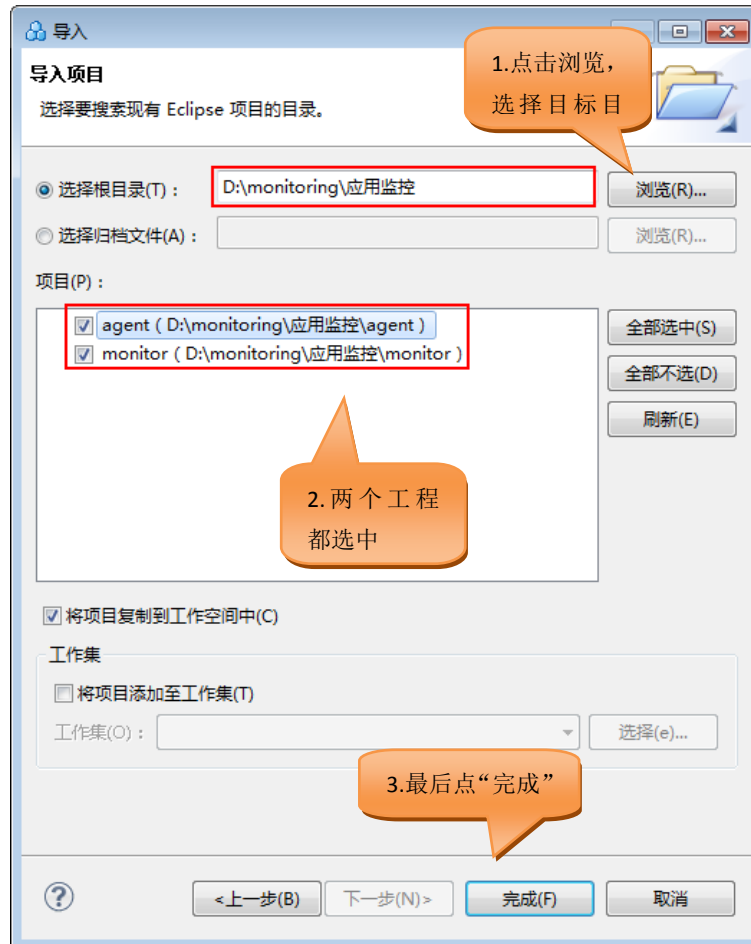
步骤 2: 打开 UniEAP WorkShop，在目标工作空间下，切换到 UniEAP 透视图，在目标开发工程上导入“步骤 1”中获得资料 1、2.过程图示如下：



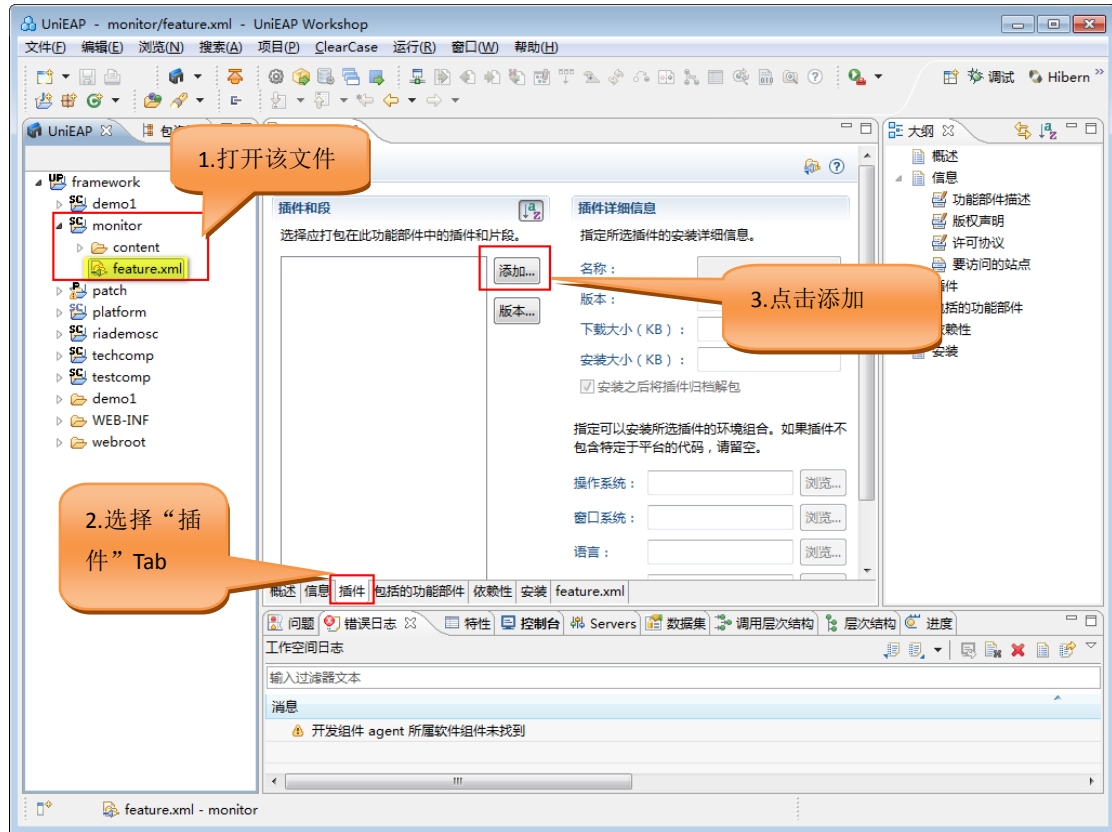
图表 2-6 选择导入菜单



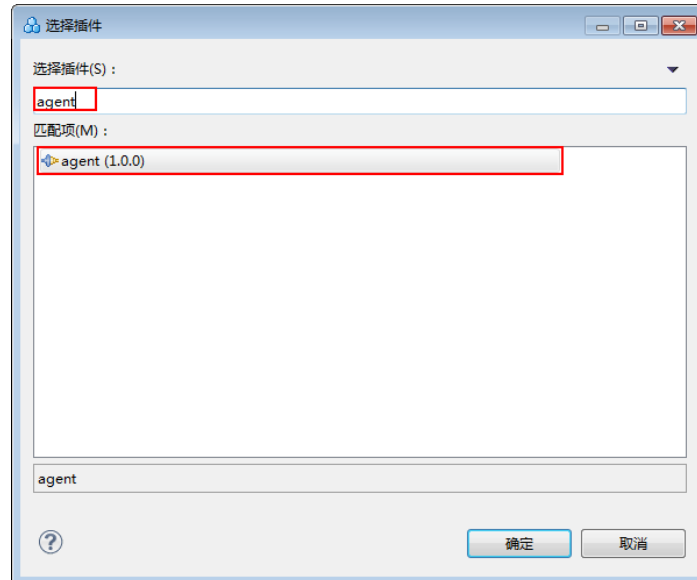
图表 2-7 导入现有项目



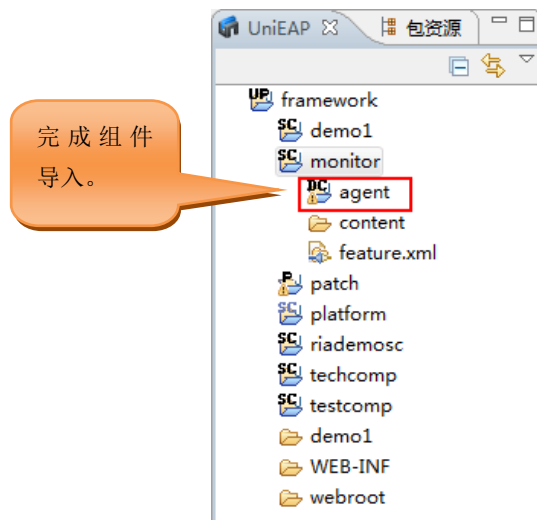
图表 2-8 选择组件



图表 2-9 配置组件关系



图表 2-10 为 Monitor SC 添加 agent DC



图表 2-11 组件导入完成

步骤 3： 监控代理属性配置修改。找到并打开监控代理的配置文件，文件位置：“agentDC/src/MonitorConfig.properties”。该配置文件是“监控代理”能否正常工作的关键，所以需要谨慎配置，尤其是红色标注部分需要特别注意。本文中以“宿主应用”代指被监控

应用。

```
#-----  
### 监控项开关配置  
#-----  
#BO、DAO Bean方法监视  
enableBeanMethodMonitoring = true  
#Web请求监视  
enableWebRequestMonitoring = true  
#用户活动监视（登入、登出）  
enableUserActivitiesMonitoring = true  
#WebSession监视（创建、销毁）  
enableWebSessionMonitoring = true  
#SQL语句执行监视  
enableSqlStatementMonitoring = true  
#方法参数及返回值（注意：打开此开关后，可能会产生特别大的数据量，所以非特殊情况，不建议打开）  
enableBoParameterMonitoring = false  
#SQL ResultSet返回值阈值，如果不想进行阈值检测，不要设置该值  
#sqlResultSetThreshold = 50000  
#Log日志监视  
enableLogMonitoring = false  
#服务提供监视，注意打开此开关的话要同时打开enableBeanMethodMonitoring开关，否则此开关无效  
enableServiceProviderMonitoring = false  
#服务调用监视，注意打开此开关的话要同时打开enableBeanMethodMonitoring开关，否则此开关无效  
enableServiceConsumerMonitoring = false  
#扩展开关  
enableExtendMonitoring = false  
  
#  
#用户可通过覆盖自定义配置覆盖缺省选项。
```

```
#-----  
### Agent自描述信息配置  
#-----  
#  
#该组配置均为“必须设置选项”!  
#  
#应用标识  
agent.appId = zdwApp  
  
#应用服务期信息  
agent.deployment.server = tomcat  
  
#Agent IP, 如不填写则Agent会自动获取网卡IP  
#agent.ip=10.4.46.21  
  
#Agent端口, 为Agent接收监控总线注册响应的端口, 如不填写默认为7999  
#agent.port=7999  
  
#Agent上下文路径, 为Agent接收监控总线注册响应的上下文路径, 如不填写默认为/jetty  
#agent.path=/jetty  
  
#-----  
### Agent追踪数据接收服务地址与认证信息配置  
#-----  
#  
#该组配置均为“必须设置选项”!  
#  
delivery.target.url=http://10.4.44.62:9999/framework/monitorPubEntry  
  
delivery.target.auth.name=admin  
  
delivery.target.auth.pwd=1  
  
#-----  
### Agent追踪数据发送配置。
```

```
#-----  
#  
#发送编码配置，缺省为plain，可自定义覆盖为gzip。  
#delivery.content.encoding=plain  
  
#发送批量数据包大小设置，缺省为20000。  
#delivery.content.batchSize=20000  
  
#监控数据队列中的最大存储数据量，队列容量超过该值后将不再接收监控数据，不设置的话则  
#该队列无上限。  
nonblockingQueue.maxSize=300000  
  
#-----  
### Agent注册相关配置  
#-----  
#  
#注册服务地址，为“必须设置选项”！  
signIn.target.url=http://10.4.44.62:9999/framework/monitorRegister  
  
#注册请求拒绝后的重试延迟，缺省为60分钟  
signIn.rejectDelay = 0.5  
  
#注册请求被搁置后的重试延迟，缺省为30分钟  
signIn.waitDelay = 0.2  
  
#注册请求发生网络失败后的充实延迟，缺省为5分钟  
signIn.timeoutDelay = 0.3  
  
#数据传输连续失败次数阈值。数据传输连续失败次数超过该值后，将重新发起新的注册。  
#缺省为3次。  
#signIn.failureCeil = 3  
  
#-----  
### monitor4jdbc代理驱动设置  
#-----
```



```
#被代理的真实驱动类，缺省为oracle.jdbc.driver.OracleDriver，也可以配置多个数据源，以分号分割。
#如若采用DBCP、C3P0等数据源，该项为"必须设置项"。如采用应用服务器自带的JNDI数据源，该项可不填写
monitor4jdbc.realDirverClass =
oracle.jdbc.driver.OracleDriver;com.mysql.jdbc.Driver
#日志过滤相关设置，logger.include代表该包下的日志需要采集，
#logger.exclude代表该包下的日志不需要被采集，如果有多个包，用分号分割
logger.include = logsearchdemosc.logsearchdemodc.bo.impl.*
logger.exclude = com.neusoft.*
```

关于各个配置文件中各个配置项的相关含义，请参见下表：

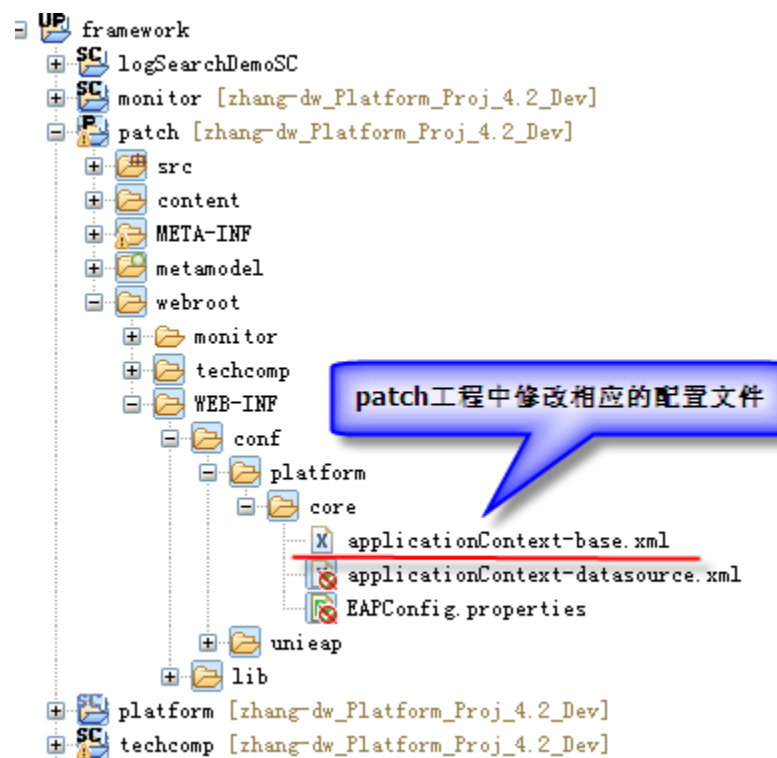
| | |
|---------------------------------|--|
| enableBeanMethodMonitoring | true 代表采集 BO、DAO 业务方法产生的监控数据，false 代表不采集该部分数据，推荐设置为 true。 |
| enableWebRequestMonitoring | true 代表采集 web 请求相关的数据，false 代表不采集该部分数据，推荐设置为 true。 |
| enableUserActivitiesMonitoring | true 代表采集在线用户登录登出的相关数据，false 代表不采集该部分数据，推荐设置为 true。 |
| enableWebSessionMonitoring | true 代表采集 Session 创建、销毁的相关数据，false 代表不采集该部分数据，推荐设置为 true。 |
| enableSqlStatementMonitoring | true 代表采集 sql 语句相关数据，false 代表不采集该部分数据，推荐设置为 true。 |
| enableBoParameterMonitoring | true 代表采集 BO 方法对应的参数信息，false 代表不采集该部分数据。由于 BO 方法对应的参数信息可能涵盖较大的数据量，对带宽和存储资源都有一定的消耗，所以如非特殊情况，建议将该值设置为 false。 |
| enableLogMonitoring | true 代表采集 log 日志相关数据，false 代表不采集该部分数据。对于不需采集 log 日志的应用，建议将该值设置为 false。 |
| enableServiceProviderMonitoring | true 代表采集 soap 服务执行的相关信息，false 代表不代表不采集该部分信息。该功能需要与 UniEAP V4 提供的 ESB 组件结合使用，所以默认情况下建议设置为 false。 |
| enableServiceConsumerMonitoring | true 代表采集 soap 服务调用的相关信息，false 代表不代表不采集该部分信息。该功能需要与 UniEAP V4 提供的 ESB 组件结合使用，所以默认情况下建议设置为 false。 |
| enableExtendMonitoring | 扩展开关，目前有两种场景需要将其设置为 true：（1） |

| | |
|----------------------------|---|
| | 使用 UniEAP V4 提供的 ESB 功能，并且需要对 ESB 进行服务监控；（2）自定义扩展 agent 组件的相关监控采集功能。其余的情况都建议设置为 false。 |
| agent.appld | 代理端应用标识，作为监控总线注册订阅的标识，必填项，并且要保证唯一性。 |
| agent.deployment.server | 应用服务器类型，必填项。 |
| agent.ip | 监控代理 IP 地址，监控总线需要通过该 IP 地址完成监控注册的过程，所以务必填写真实、可通讯的 IP。如不填写会自动获取网卡的 IP 地址发送给监控总线。 |
| agent.port | 监控代理端口，注意不是应用的端口，监控总线需要通过该端口完成监控注册的过程，如不填写会自动分配一个 7999 端口号用于完成注册。所以不建议手动修改该值。 |
| agent.path | 监控代理上下文路径，注意不是应用的上下文路径，监控总线需要通过该路径完成监控注册的过程，如不填写会自动分配一个 /jetty 为默认上下文路径。所以不建议手动修改该值。 |
| delivery.target.url | 发送监控数据的目标 url 地址，是 restful 的调用方式，以 http://10.4.44.62:9999/framework/monitorPubEntry 为例，其中 http://10.4.44.62:9999/framework 要修改为当前监控总线对应的“ http://IP:端口/应用上下文 ”，而 /monitorPubEntry 是固定的，不需改变。 |
| delivery.target.auth.name | 认证相关，调用目标请求的用户名信息。 |
| delivery.target.auth.pwd | 认证相关，调用目标请求的密码信息。 |
| delivery.content.batchSize | 单次异步发送监控数据的上限值，默认为 20000。 |
| nonblockingQueue.maxSize | 无阻塞队列的上限值，设置过大或不设置，可能会出现 OOM 问题；设置过小可能会丢失一些监控数据。默认值为 300000。 |
| signIn.target.url | 发起注册的目标 url 地址，是 restful 的调用方式，以 http://10.4.44.62:9999/framework/monitorRegister 为例，其中 http://10.4.44.62:9999/framework 要修改为当前监控总线对应的“ http://IP:端口/应用上下文 ”，而 /monitorRegister 是固定的，不需改变。 |
| signIn.rejectDelay | 注册请求拒绝后的重试延迟，缺省为 60 分钟。 |
| signIn.waitDelay | 注册请求被搁置后的重试延迟，缺省为 30 分钟。 |

| | |
|------------------------------|--|
| signIn.timeoutDelay | 注册请求发生网络失败后的延迟，缺省为 5 分钟。 |
| signIn.failureCeil | 数据传输连续失败次数阈值。数据传输连续失败次数超过该值后，将重新发起新的注册。缺省为 3 次。 |
| monitor4jdbc.realDirverClass | 被代理的真实驱动类，当使用 DBCP、C3P0 数据源的时候，如若需要监控 sql 语句的执行情况，需要配置该项，默认值为 oracle.jdbc.driver.OracleDriver。 |

2.2.3.2 配置 AOP 动态代理

如若需要对于 UniEAP V4 中的 BO、DAO 方法执行情况进行监控，则需要在配置 Spring 动态代理的时候添加相应配置，如下所示：



修改的内容如下：

```
<!-- UniEAP V4 提供的自动代理 -->
<bean id="unieapAutoProxy"
```

```
class="com.neusoft.unieap.core.autoproxy.UniEAPAutoProxyCreator">
  <property name="beanNames">
    <list>
      <value>*_bo</value>
      <value>eventPublisher</value>
      <value>securityContextLogoutHandler</value>
    </list>
  </property>
  <property name="interceptorNames">
    <list>
      <value>baseTxAdvisor</value>
      <value>jtaTxAdvisor</value>
      <value>auditAdvisor</value>
      <value>beanInvokingCollector</value>
    </list>
  </property>
</bean>
```

通过添加一个新的拦截器来实现对 BO、DAO 方法的信息采集，如果不配置该拦截器，则不会采集到这部分信息。

2.2.3.3 配置 JDBC 驱动代理

应用监控往往需要对产生的 SQL 语句进行采集，此时需要对 EAPConfig.properties 或 applicationContext-datasource.xml 数据源配置文件进行一定的修改，建议在 Patch 工程中进行修改。现以使用 DBCP 数据源为例，修改 patch 工程下的 EAPConfig.properties 文件内容如下：

```
#unieap.datasource.driverClass=oracle.jdbc.driver.OracleDriver
#unieap.datasource.jdbcUrl=jdbc:oracle:thin:@10.4.44.32:1521:oracle
unieap.datasource.driverClass=com.neusoft.unieap.monitor.jdbc.monitor4jdbc.DriverSpy
unieap.datasource.jdbcUrl=jdbc:monitor4jdbc:oracle:thin:@10.4.44.32:1521:oracle
unieap.datasource.user=v44test
unieap.datasource.password=v44test
unieap.datasource.initialSize=20
```

```
unieap.datasource.maxActive=1000
unieap.datasource.maxIdle=5
unieap.datasource.maxWait=10
unieap.hibernate.dialect=org.hibernate.dialect.Oracle9Dialect
unieap.quartz.delegate=org.quartz.impl.jdbcjobstore.oracle.OracleDelegate
```

其中被注释的绿色部分是原来配置的真实数据库驱动和 jdbcUrl，这里修改为 UniEAP V4 提供的特定驱动类型 com.neusoft.unieap.monitor.jdbc.monitor4jdbc.DriverSpy，jdbcUrl 是在原来的 jdbc: 后面加入 monitor4jdbc:，此两处结合 2.2.3.1 表格中的 monitor4jdbc.realDirverClass 配置的真实数据库驱动，可实现在 DBCP、C3P0 数据源下的 SQL 语句监控功能。

另外，在生产系统中经常使用的是应用服务器提供的 JNDI 数据源（如 Weblogic JNDI Datasource），此时修改 EAPConfig.properties 文件就毫无用处了。配置的方式是要修改 applicationContext-datasource.xml 文件，具体可参考附录 F 中的相关内容。

配置完毕后，使用 UniEAP 一键发布后，就可以启动应用了。启动应用时，监控代理会自动向监控总线发送注册请求，等待监控总线审核。至此，agent 端配置完成。

第3章 监控总线与监控控制台使用说明

本章节内容主要涵盖“监控总线”应用与“监控控制台”应用的系统操作说明。其中 3.1 小节中的“接入管理”属于“监控总线”应用功能范畴，3.2-3.4 小节内容属于“监控控制台”应用功能范畴。“快速部署”模式下，所有的功能菜单将出现在一个应用控制台中；而“可伸缩部署模式”下，两个范畴的功能菜单会体现在各自的应用系统中。

3.1 接入管理

3.1.1 应用注册管理

1) 应用注册

- a) 前面提到，宿主应用启动后，“监控代理”会自动向监控总线发送注册请求。宿主应用只有在被注册后，监控代理才会实时采集监控信息发往监控总线，监控系统才能对此应用进行监控。

- b) 注册方法: 进入监控控制台→接入管理→待注册管理→待处理注册请求→选择要注册的应用

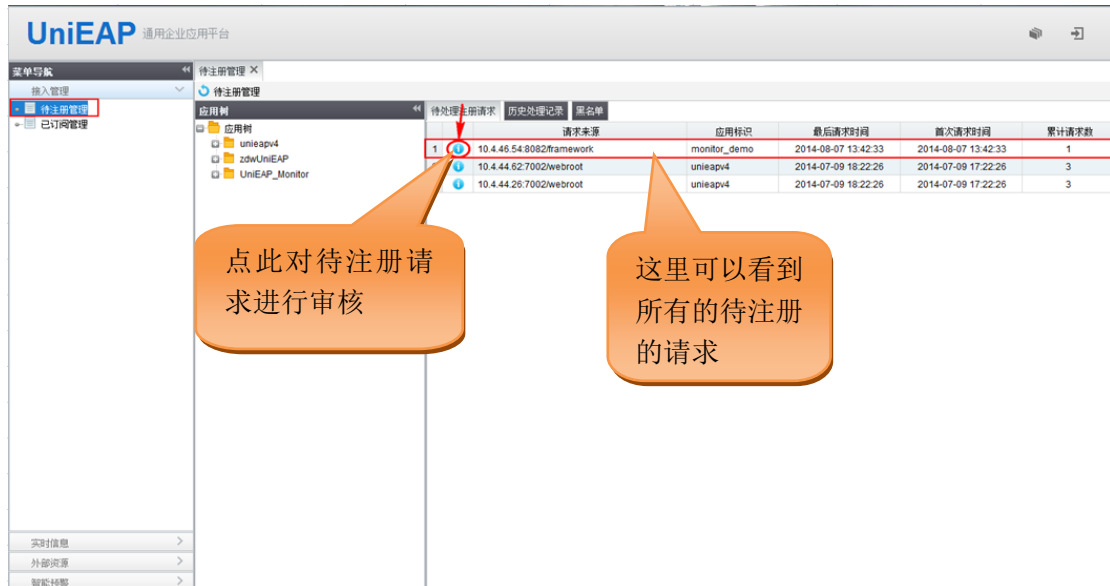



图 3-1 应用注册

点击“ info 图标进入审核处理页面，页面包括两部分：“注册信息”、“审核处理”；审核处理中有三个选项，这里选择“接受，并注册到应用 monitor_demo”，点击确认会提示注册成功。然后就可以在左侧的“应用树”中看到刚注册的应用“monitor_demo”。

待注册管理 × 已订阅管理 ×

待注册管理 > 审核处理

注册信息

IP地址: 10.4.46.54

端口号: 8082

应用路径: framework

最后请求时间: 2014-08-07 13:42

首次请求时间: 2014-08-07 13:42

累计请求次数: 1

审核处理

☒ 接受, 并注册到应用 monitor_demo

☐ 拒绝, 并添加到黑名单

☐ 忽略

确认 取消

图 3-2 注册请求审核



图 3-3 注册成功

2) 应用树:

一个应用可以有多个实例，每个实例的 url 都是有区别的，将这些实例都注册在同一个应用下，就组成了一个应用树。



图 3-4 应用树

3.1.2 已订阅管理

已注册的应用需要订阅后才能查看其实时信息（具体的订阅步骤参见 3.2.1），在已订阅管理中可以查看订阅者的基本信息。



图 3-5 已订阅管理

3.2 实时信息

3.2.1 监控信息订阅

1) 进入监控控制台，选择实时信息导航菜单。



图 3-6 监控信息订阅步骤 1

2) 选择监控信息订阅子菜单，可以看到已订阅的应用。然后点击更新订阅。

| | | | | |
|----------|----------------|--------|--------|---------------------|
| 监控信息订阅 X | | | | |
| 监控信息订阅 | | | | |
| 已订阅 更新订阅 | | | | |
| | 应用名 | 历史数据流量 | 最后活跃时间 | 开始订阅时间 |
| 1 | zdwUniEAP | 0 | - | 2014-07-17 11:46:52 |
| 2 | unieapv4 | 0 | - | 2014-07-03 10:55:47 |
| 3 | UniEAP_Monitor | 0 | - | 2014-08-05 11:50:00 |

图 3-7 监控信息订阅步骤 2

3) 选择更新订阅 tab 后，可以看到如下信息。在此页可以对应用的监控信息进行订阅或者退订。选择“monitor_demo”应用点击“订阅”。

| | | | |
|----------|----------------|------|----|
| 监控信息订阅 X | | | |
| 监控信息订阅 | | | |
| 已订阅 更新订阅 | | | |
| | 应用名 | 订阅状态 | 操作 |
| 1 | unieapv4 | 已订阅 | 退订 |
| 2 | zdwUniEAP | 已订阅 | 退订 |
| 3 | UniEAP_Monitor | 已订阅 | 退订 |
| 4 | monitor_demo | 未订阅 | 订阅 |

图 3-8 监控信息订阅步骤 3

4) 显示如下消息，表示订阅成功。



图 3-9 订阅成功提示信息

3.2.2 业务事务监控

- 1) 进入监控控制台，选择实时信息导航菜单，选择业务事物监控子菜单，可以看到如下画面。



图 3-10 业务事物监控画面

- 2) 选择好应用、时间后，点击刷新就可以看到应用的实时监控信息了。坐标系的横轴为调用时刻，纵轴为执行耗时（ms），可以再坐标系中看到很多点，这些点反映的就是系统工作中的实时信息。

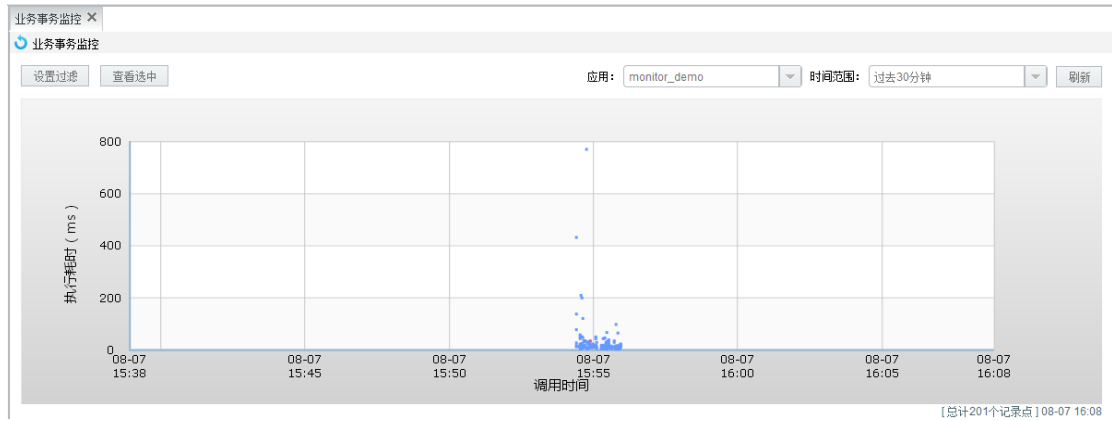


图 3-11 系统实时信息散点图

- 3) 想要查看某一区域系统执行的详细信息, 只要在散点图中拖动鼠标左键选中想要查看的区域, 点击“查看选中”按钮就可以查看详细信息。



图 3-12 选择区域

选中的区域的详细信息:



图 3-13 选中的记录

4) 单击一条记录，可以查看该记录的详细信息



图 3-14 记录详细信息

5) 调用树信息

| 名称 | 执行时间 | 查看详情 |
|---|------|------|
| com.practice.demo1.demo1_1.bo.impl.StaffBOImpl.insert | 769 | 查看详情 |
| com.practice.demo1.demo1_1.dao.impl.StaffDAOImpl.insert | 769 | 查看详情 |
| [SQL statement] | 30 | 查看详情 |
| [SQL statement] | 40 | 查看详情 |
| [SQL statement] | 33 | 查看详情 |
| [SQL statement] | 21 | 查看详情 |
| [SQL statement] | 19 | 查看详情 |
| [SQL statement] | 42 | 查看详情 |
| [SQL statement] | 20 | 查看详情 |
| [SQL statement] | 20 | 查看详情 |
| [SQL statement] | 20 | 查看详情 |
| [SQL statement] | 21 | 查看详情 |
| [SQL statement] | 39 | 查看详情 |
| [SQL statement] | 41 | 查看详情 |
| [SQL statement] | 48 | 查看详情 |
| [SQL statement] | 21 | 查看详情 |
| [SQL statement] | 19 | 查看详情 |
| [SQL statement] | 154 | 查看详情 |
| [SQL statement] | 58 | 查看详情 |
| [SQL statement] | 49 | 查看详情 |

图 3-15 调用树

6) 点击查看详情

详细信息

执行时间 / 耗时占比:

33ms / 4%

影响行数:

1

SQL:

insert
into
t_employee
(employee_idcode, employee_name, employee_nation,
employee_minzu, employee_sex, employee_birth_date,
employee_join_date, employee_province, employee_idcardno,

图 3-16 调用树详情

7) 异常信息

业务事务监控

散点分布图 > 查看选中

所有选中记录

| 名称 | 数量 |
|---|----|
| com.practice.demo1.demo1_1.bo.impl.StaffBOImpl.addStaff | 2 |

提示：共选中[2]条记录，涉及到[1]种业务方法

系统异常记录

| 名称 | 数量 |
|---|----|
| com.practice.demo1.demo1_1.bo.impl.StaffBOImpl.addStaff | 2 |

提示：所选范围内包含[2]条系统异常记录，涉及到[1]种业务方法

业务异常记录

| 名称 | 数量 |
|----|----|
|----|----|

提示：所选范围内包含[0]条业务异常记录，涉及到[0]种业务方法

图 3-17 异常记录



图 3-18 查看记录

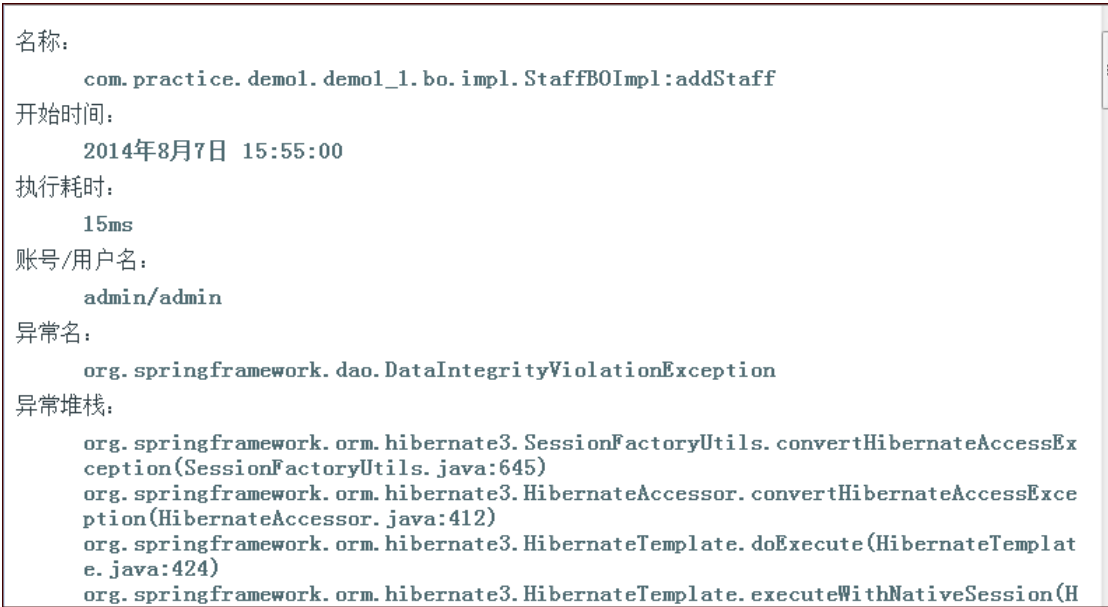


图 3-19 异常堆栈信息

3.2.3 SQL 语句监控

如果想要进行 SQL 语句监控，则必须要在监控代理端打开“SQL 语句执行监视”开关。
SQL 语句监控操作与业务事物监控类似，有些地方不再赘述。

- 注：如果业务系统使用 JNDI 数据源，则需要进行特殊配置。详见附录 F
1. 进入监控控制台，选择实时信息导航菜单，选择“SQL 语句监控”子菜单
选择筛选条件，点击“刷新”。选中某一区域，点击“查看选中”。

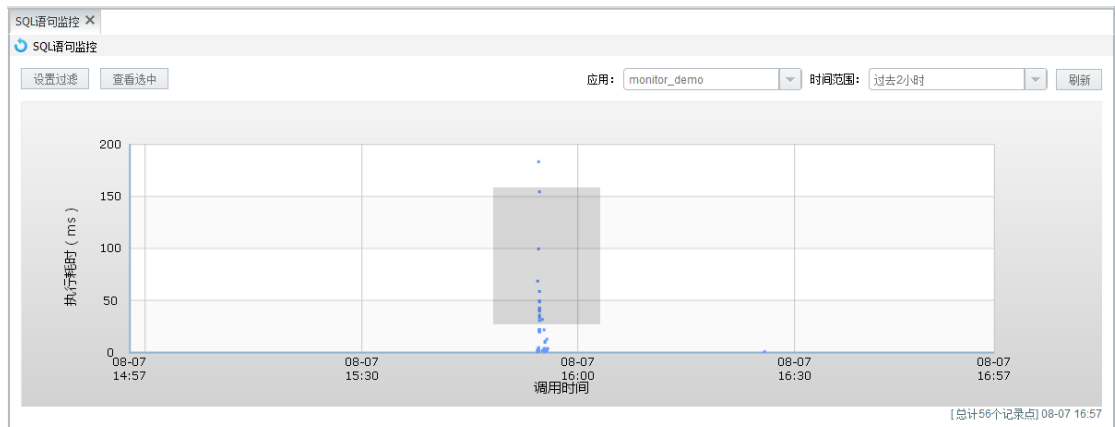


图 3-20 SQL 语句监控

SQL语句监控

SQL语句监控

散点分布图 > 查看选中

所有选中记录

| 名称 | 数量 |
|--|----|
| insert into t_employee (employee_idcode, employee_name, employee_nation, employee_minzu, employee_sex, employee_birth_date, employee_join_date, employee_pro | 1 |
| insert into t_employee (employee_idcode, employee_name, employee_nation, employee_minzu, employee_sex, employee_birth_date, employee_join_date, employee_pro | 1 |
| insert into t_employee (employee_idcode, employee_name, employee_nation, employee_minzu, employee_sex, employee_birth_date, employee_join_date, employee_pro | 1 |
| select distinct resourceau0_resource_id as col_0_0_ from up_sec_resource_authority resourceau0_ where resourceau0_resource_type=? and resourceau0_ | 1 |
| select templovee0_id as id65_ templovee0_employee_idcode as employee2_65_ templovee0_employee_name as employee3_65_ templovee0_employee_nati | 1 |
| insert into t_employee (employee_idcode, employee_name, employee_nation, employee_minzu, employee_sex, employee_birth_date, employee_join_date, employee_pro | 1 |
| insert into t_employee (employee_idcode, employee_name, employee_nation, employee_minzu, employee_sex, employee_birth_date, employee_join_date, employee_pro | 1 |
| insert into t_employee (employee_idcode, employee_name, employee_nation, employee_minzu, employee_sex, employee_birth_date, employee_join_date, employee_pro | 1 |
| insert into t_employee (employee_idcode, employee_name, employee_nation, employee_minzu, employee_sex, employee_birth_date, employee_join_date, employee_pro | 1 |
| insert into t_employee (employee_idcode, employee_name, employee_nation, employee_minzu, employee_sex, employee_birth_date, employee_join_date, employee_pro | 1 |
| insert into t_employee (employee_idcode, employee_name, employee_nation, employee_minzu, employee_sex, employee_birth_date, employee_join_date, employee_pro | 1 |
| insert into t_employee (employee_idcode, employee_name, employee_nation, employee_minzu, employee_sex, employee_birth_date, employee_join_date, employee_pro | 1 |
| select templovee0_id as id65_ templovee0_employee_idcode as employee2_65_ templovee0_employee_name as employee3_65_ templovee0_employee_nati | 1 |

提示：共选中[14]条记录，涉及到[14]条SQL语句

图 3-21 被选中的记录

2. 选择单条记录查看详细信息

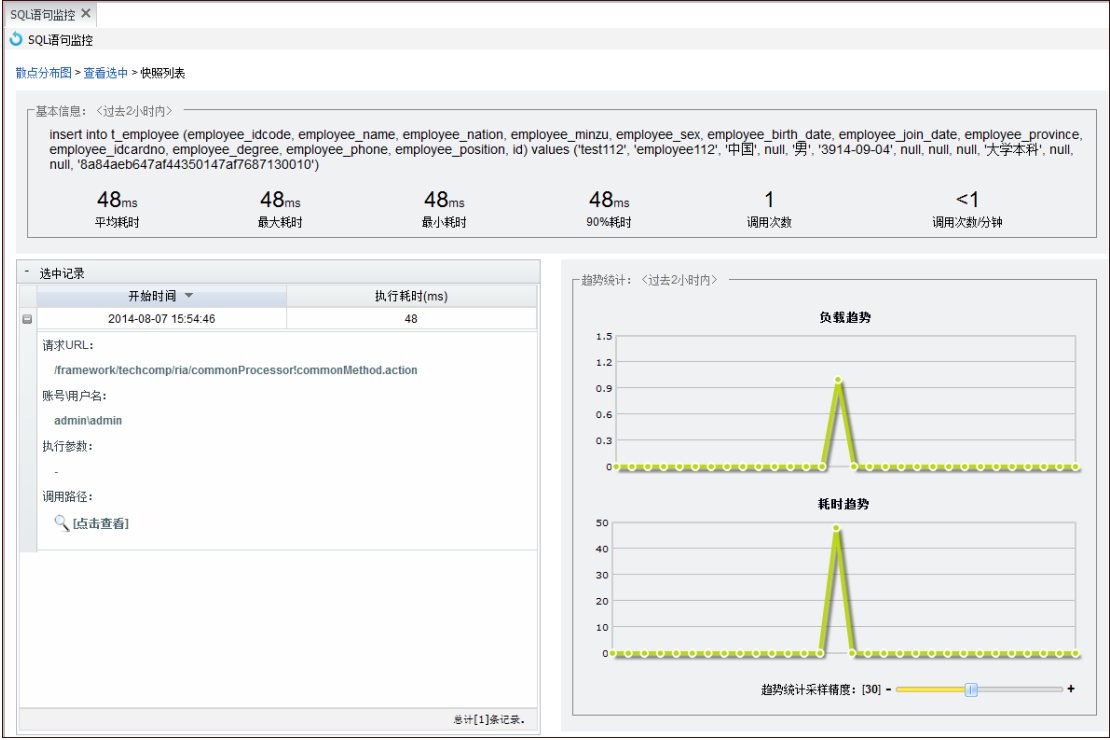


图 3-22 快照

3. 调用路径及查看详细

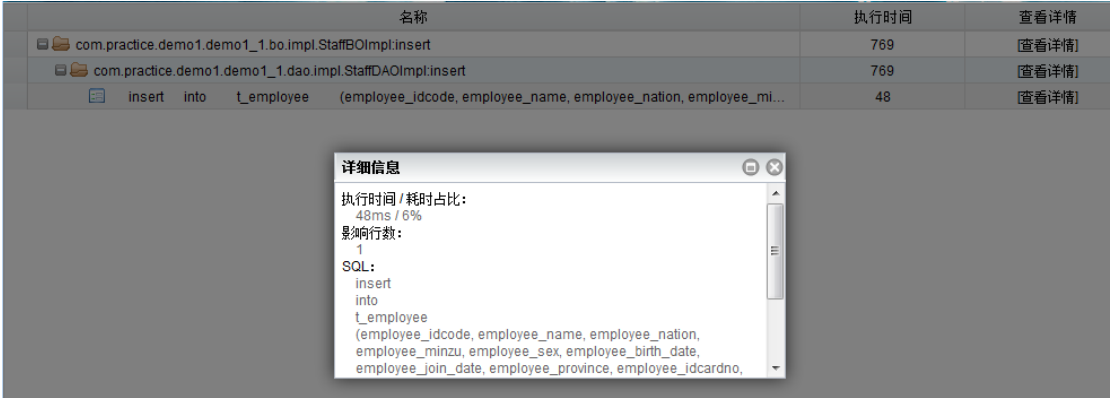


图 3-23 调用路径

3.2.4 日志查询

日志查询提供了标准日志数据的查询以及用户自定义的日志数据查询功能。UniEAP 监控平台能够将用户基于 SLF4j、Log4j 等方式创建的日志数据进行收集，日志查询提供日志的查询功能。为了满足运维人员针对业务系统中的业务数据进一步分析的需求，日志查询扩展了标准的日志接口，为开发人员提供了更多类型的数据传递以及查询功能。

基于 UniEAP V4 开发的业务系统产生的标准日志信息，都会通过 UniEAP 监控平台收集到 Mongoddb 数据库中，但是针对日志索引的创建是保存在 Solr 中（通过 Solr 构造的索引在查询上比仅仅使用 Mongoddb 建立索引性能上要好很多，Solr 的安装方式请参照[附录 B](#)）。日志查询提供了过滤功能，使用者可以通过：日志级别、日志内容、日志类型、账号、用户名、类名、方法名、会话 ID 的查询信息进行过滤查询。

| | 日志触发时间 | 级别 | 日志内容 | 日志类型 | 账号 | 用户名 | 会话ID |
|----|-------------------------|------|----------------------|-------|-------|-------|-------------------|
| 1 | 2014/10/23/ 11:48:36 am | INFO | 测试用户，用户名称为 bbb | 用户 | admin | admin | A538ACC66B8A5352C |
| 2 | 2014/10/23/ 11:48:36 am | INFO | 测试Test, Test名称为 test | TestF | admin | admin | A538ACC66B8A5352C |
| 3 | 2014/10/23/ 11:48:36 am | INFO | 测试Test, Test名称为 test | TestC | admin | admin | A538ACC66B8A5352C |
| 4 | 2014/10/23/ 11:48:36 am | INFO | 测试Test, Test名称为 test | TestN | admin | admin | A538ACC66B8A5352C |
| 5 | 2014/10/23/ 11:48:36 am | INFO | 测试Test, Test名称为 test | TestO | admin | admin | A538ACC66B8A5352C |
| 6 | 2014/10/23/ 11:48:36 am | INFO | 测试Test, Test名称为 test | TestI | admin | admin | A538ACC66B8A5352C |
| 7 | 2014/10/23/ 11:48:36 am | INFO | 测试Test, Test名称为 test | TestI | admin | admin | A538ACC66B8A5352C |
| 8 | 2014/10/23/ 11:48:36 am | INFO | 测试Test, Test名称为 test | TestA | admin | admin | A538ACC66B8A5352C |
| 9 | 2014/10/23/ 11:48:36 am | INFO | 测试Test, Test名称为 test | TestH | admin | admin | A538ACC66B8A5352C |
| 10 | 2014/10/23/ 11:48:36 am | INFO | 测试Test, Test名称为 test | TestG | admin | admin | A538ACC66B8A5352C |
| 11 | 2014/10/23/ 11:48:36 am | INFO | 测试Test, Test名称为 test | TestK | admin | admin | A538ACC66B8A5352C |
| 12 | 2014/10/23/ 11:48:36 am | INFO | 测试Test, Test名称为 test | TestE | admin | admin | A538ACC66B8A5352C |

平台扩展了日志接口，为用户提供了更加丰富的日志信息采集的功能，以 Info 为例接口，提供了两种扩展接口，具体如下所示：

```
/**
 * INFO级别的日志信息记录
 * @param arg0 日志内容，支持{}的方式进行参数传递
 * @param arg1 传递到日志内容的信息
 * @param logType 日志类型
 */
```

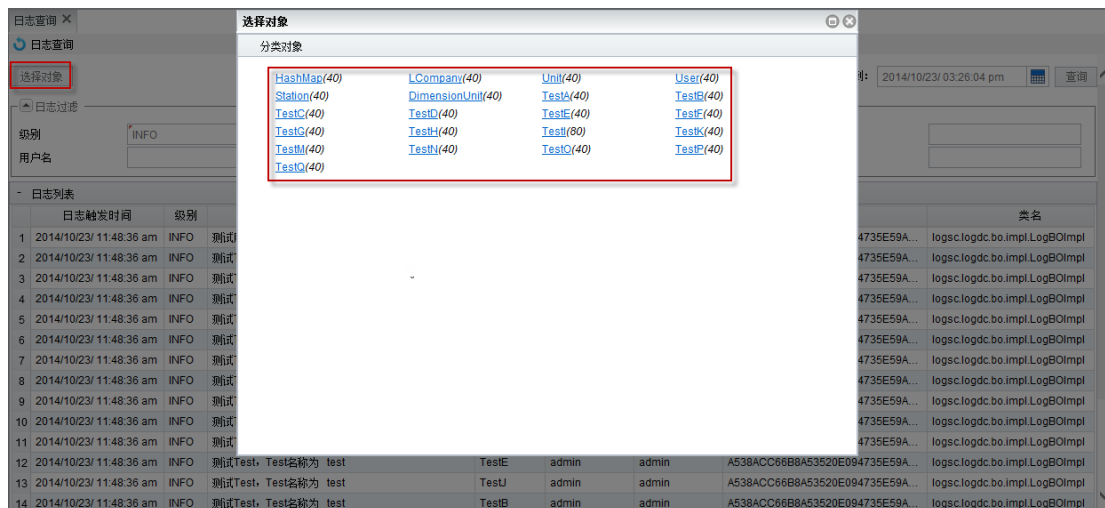
```
public void info(String arg0, Object[] arg1, String logType);

/**
 * INFO级别的日志信息记录
 * @param arg0 日志内容，支持{}的方式进行参数传递
 * @param arg1 传递到日志内容的信息
 * @param logType 日志类型
 * @param extendObj 用户自定义的扩展对象
 */
public void info(String arg0, Object[] arg1, String logType, Object extendObj);
```

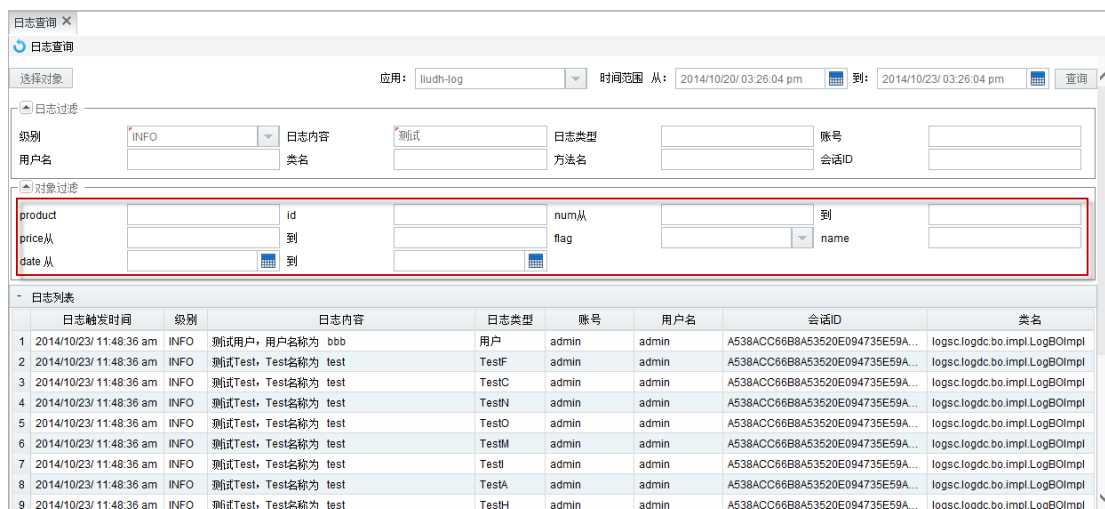
举例来说，扩展接口调用的方式大概如下 **loggerProxy.info("创建公司成功，公司名称为 {}",new Object[] {companyName},"创建公司",company);**

- 第一个参数是日志内容，记录该操作是创建公司，并提供“{}”的方式将第二个参数的值传递进来；
- 如果需要传递多个值，只需要在第一个参数中设置多个“{}”，在第二个参数中传递相应的值即可；
- 第三个参数是日志类型，用来记录该日志执行的是什么业务操作，
- 最后一个参数是用户自定义的扩展对象，会将该扩展对象记录在日志中，该例中记录的是公司对象，会将公司对象中的所有属性记录在日志中。

日志查询会对扩展对象进行统计，并提供基于扩展对象属性的日志查询功能。如下所示，点击“选择对象”，会将该应用下，在一段时间范围内产生的扩展对象进行分组以及数量统计：



选择一个对象可以进行进一步的查询，日志查询会根据对该对象属性的分析自动生成查询条件：



以 LCompany 对象为例，LCompany 的属性包括：product、id、num、price、flag、date。其中：

- 字符串类型的数据有 product、id、name，这三个属性的查询条件输入域是 Textbox；
- 数值类型的数据有 num、price，这两个属性的查询条件输入域是 NumberTextBox，查询方式是按照范围的方式进行查询；
- 布尔类型的数据有 flag，该属性的查询条件是 Combobox，里面包含了“TRUE”和“FALSE”两个选择项；

- 日期类型的数据有 **date**，该属性的查询条件是 **DateTextBox**，查询方式是按照范围的方式进行查询。

3.2.5 在线用户信息查看

在线用户统计功能可以监控应用的在线人员的基本信息，包括其登录、登出被监控系统的时间、IP 地址、MAC 地址、计算机名、历史登录信息、在线时长、访问过的菜单及业务对象等；并且支持查询功能等。同时，还可以查看用户的历史登录数据以及用户登录系统后的行为。

在线用户统计的使用步骤如下：

- 1) 进入【应用监控】控制台，选择【在线用户】→【在线用户信息】。



图 3.2.5-1 在线用户菜单

- 2) 【在线用户信息】页面：
 - 【设置过滤】按钮：条件查询开关。默认关闭，此时，查询的是所选应用下的所有在线用户。
 - 【选择应用】：选择需要查看的应用，必选。
 - 【查询】按钮：查询所选应用下的所有在线用户。



图 3.2.5-2 在线用户信息页面

- 3) 条件查询：点击【设置过滤】按钮，打开查询条件域。
- 在线状态：
 - 【在线】：查询所选应用下的所有的在线用户信息。
 - 【所有】：查询所有登录过的所选应用的用户的用户的信息，换句话说就是当前系统的登录履历。
 - 【帐户】、【用户名】为模糊查询。
 - 可选的登录时间范围。



图 3.2.5-1 设置过滤

- 4) 查看用户登录后的访问详情
- 点击【查看】超链接，查看用户登录后的访问详情，包括访问的菜单、访问的 BO 对象及方法、方法执行的时间。

在线用户查看

刷新过滤

选择应用: liudh-log 查询

在线用户过滤

帐户: 1 用户名: 登录时间: 2014-10-22 14:46:00 到 2014-10-22 15:17:00 在线状态: 在线 所有

| 在线用户列表 | 帐户 | 用户名 | 登录时间 | 登出时间 | 在线时长 | 最后访问菜单 | 最后访问业务对象 | 计算机名 | IP地址 | MAC地址 | 访问详情 |
|--------|-------|-------|---------------------|---------------------|-----------|--------|---|-------|------------|-------------------|--------------------|
| 1 | liudh | liudh | 2014-10-22 14:46:28 | 2014-10-22 15:17:00 | 0小时30分31秒 | 业务角色管理 | com.neusoft.unieap.techcomp.ria.menu.bo.impl.M... | Snail | 10.4.46.12 | 00-21-86-9F-6B-65 | 查看 |

退出系统前用户最后的行为

查看用户登录后的访问详情

图 3.2.5-2 过滤后的用户信息

【访问详情】页面

| 访问详情 | | | | |
|------|--------|--|--------------------------|---------------------|
| | 访问菜单 | 访问BO | 调用方法 | 调用时间 |
| 1 | 组织管理角色 | | | |
| 2 | | com.neusoft.unieap.techcomp.org.bo.impl.AdminRoleBOImpl | | |
| 3 | | | getAdminRolesByParentId | 2014-10-22 14:46:39 |
| 4 | | | getDirectAdminRoles | 2014-10-22 14:46:38 |
| 5 | | | getManagedAdminRoleUnits | 2014-10-22 14:46:37 |
| 6 | | | getUsersByAdminRoleId | 2014-10-22 14:46:39 |
| 7 | | com.neusoft.unieap.techcomp.ria.menu.bo.impl.MenuAuthBOImpl | | |
| 8 | | | getAuthorityApps | 2014-10-22 14:46:38 |
| 9 | | | getAuthorityApps | 2014-10-22 14:46:42 |
| 10 | | com.neusoft.unieap.techcomp.security.bo.impl.ResourceAuthorizeBOImpl | | |
| 11 | 业务角色管理 | | | |
| 12 | | com.neusoft.unieap.techcomp.variability.bo.impl.VarTreeBOImpl | | |
| 13 | | com.neusoft.unieap.techcomp.security.bo.impl.SecurityAdminRoleBOImpl | | |

图 3.2.5-3 用户登录后的访问详情

3.3 智能预警

3.4.1 预警项管理

1) 预警

当系统发生异常和性能问题时，可及时预警。首先得确保 monitor SC 下引入了 alert DC 工程。

2) 预警项管理操作指南

a) 查看已经建立的预警项目

进入监控控制台，选择智能预警导航菜单,选择预警项管理子菜单。

b) 在右侧可以看到预警项管理的页面。页面情况如下：

| | 名称 | 生效 | 告警级别 | 自定义告警内容 | 描述 |
|---|---------------|-----|------|----------------|-------------------|
| 1 | 业务方法执行时间超过0ms | OFF | 一般 | 业务方法执行时间超过0ms. | 业务方法执行时间超过0ms |
| 2 | test | OFF | 一般 | 出错信息 | |
| 3 | 异常提醒 | ON | 严重 | 发生异常! | 对业务执行过程中产生的异常进行预警 |
| 4 | 执行时间过长 | ON | 中等 | 业务方法执行时间过长! | 对执行时间过长的方法预警 |

新增和删除按钮。
新增：可以自定义新的预警项。
删除：删除预警项

图 3-24 预警项管理

3) 新增预警项

a) 点击右上角的新增按钮，出现如下画面，画面各项信息解释如下：

预警项管理

新增预警策略

基本属性

名称: 异常预警

描述: 对系统异常和业务异常进行预警

生效: ☒ 是 ☐ 否

条件

资源类型: 系统异常 监测对象: *

资源类型: 业务异常 监测对象: *

条件连接符: OR

自定义告警内容: 系统发生异常!

告警设置

告警级别: 严重

告警频率: ☒ 每次条件满足时 ☐ 每触发 次条件在 内

响应方式

☒ 邮件预警: hui.wang.neu@neusoft.com

☐ 短信预警:

☐ 桌面提醒:

动作频率: ☐ 第一次告警发生时 ☒ 每次告警发生时 ☐ 在告警有效期内, 每隔

确定

取消

"*" 代表所有方法

增加或删除条件

警告级别选择

警告内容

预警触发时发送邮件提醒

编辑完成、点击确定

图 3-25 新增预警项

b) 点击“确定”按钮保存后，可以在预警管理业看到新增的预警项，此时预警已经生效。

| 预警项管理 | | | | | | 新增 | 删除 |
|-------|---------------|-----|------|----------------|-------------------|----|----|
| | 名称 | 生效 | 告警级别 | 自定义告警内容 | 描述 | | |
| 1 | 业务方法执行时间超过0ms | OFF | 一般 | 业务方法执行时间超过0ms. | 业务方法执行时间超过0ms | | |
| 2 | test | OFF | 一般 | 出错信息 | | | |
| 3 | 异常预警 | OFF | 严重 | 发生异常! | 对业务执行过程中产生的异常进行预警 | | |
| 4 | 执行时间过长 | OFF | 中等 | 业务方法执行时间过长! | 对执行时间过长的方法预警 | | |
| 5 | 异常预警 | ON | 严重 | 系统发生异常! | 对系统异常和业务异常进行预警 | | |

图 3-26 预警生效

c) 预警邮件
具体的邮件信息如下图：

系统发生异常！

发件人：dpmail
收件人：hui.wang.neu
时 间：2014-08-11 11:26

隐藏

回复 | 转发

尊敬的系统监控用户，您好，
以下是系统监测到的预警信息，为了避免对系统造成影响，请及时处理

预警策略信息

名称：异常预警
预警级别：严重
预警内容：系统发生异常！
预警时间：2014-8-11 11:26:33

条件定义

系统异常产生条件：*

满足场景

异常名称：org.springframework.dao.DataIntegrityViolationException
异常描述：not-null property references a null or transient value: com.practice.demol.demol_1.entity.TEmployee.employeeName;
nested exception is org.hibernate.PropertyValueException: not-null property references a null or transient value:
com.practice.demol.demol_1.entity.TEmployee.employeeName
对应方法：com.practice.demol.demol_1.dao.impl.StaffDAOImpl.addStaff
操作人：admin
异常堆栈信息：
org.springframework.orm.hibernate3.SessionFactoryUtils.convertHibernateAccessException
(SessionFactoryUtils.java:645);org.springframework.orm.hibernate3.HibernateAccessor.convertHibernateAccessException
(HibernateAccessor.java:412);org.springframework.orm.hibernate3.HibernateTemplate.doExecute
(HibernateTemplate.java:424);org.springframework.orm.hibernate3.HibernateTemplate.executeWithNativeSession
(HibernateTemplate.java:374);org.springframework.orm.hibernate3.HibernateTemplate.saveOrUpdate
(HibernateTemplate.java:748);com.practice.demol.demol_1.dao.impl.StaffDAOImpl.addStaff
(StaffDAOImpl.java:30);sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method);sun.reflect.NativeMethodAccessorImpl.invoke
(Unknown Source);sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source);java.lang.reflect.Method.invoke(Unknown
Source);org.springframework.aop.support.AopUtils.invokeJoinpointUsingReflection
(AopUtils.java:307);org.springframework.aop.framework.ReflectiveMethodInvocation.invokeJoinpoint
(ReflectiveMethodInvocation.java:182);org.springframework.aop.framework.ReflectiveMethodInvocation.proceed
(ReflectiveMethodInvocation.java:149);com.neusoft.unieap.monitor.collector.interceptor.BeanInvokingCollector.invoke
(BeanInvokingCollector.java:121);org.springframework.aop.framework.ReflectiveMethodInvocation.proceed
(ReflectiveMethodInvocation.java:171);org.springframework.aop.framework.JdkDynamicAopProxy.invoke
(JdkDynamicAopProxy.java:204);SPProxy194.addStaff(Unknown Source);com.practice.demol.demol_1.bo.impl.StaffBOImpl.addStaff
(StaffBOImpl.java:35);sun.reflect.GeneratedMethodAccessor161.invoke(Unknown
Source);sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source);java.lang.reflect.Method.invoke(Unknown
Source);org.springframework.aop.support.AopUtils.invokeJoinpointUsingReflection

图 3-27 告警邮件

3.4.2 告警历史查看

进入监控控制台，选择“智能预警”导航菜单,选择“告警历史查看”子菜单就可以查看所有的告警历史及相关信息。

报警项管理

告警历史查看


告警历史查看

| 告警资源类型 | 告警发生时间 | 告警资源名称 | 告警级别 | 策略内容 | 操作人员 |
|-----------|-----------------------|---|------|---------|-------|
| 1101 系统异常 | 2014年08月11日 10时17分... | org.springframework.dao.DataIntegrityViolationException | 严重 | 发生异常! | admin |
| 1102 系统异常 | 2014年08月11日 10时17分... | org.springframework.dao.DataIntegrityViolationException | 严重 | 发生异常! | admin |
| 1103 系统异常 | 2014年08月11日 10时17分... | org.springframework.dao.DataIntegrityViolationException | 严重 | 发生异常! | admin |
| 1104 系统异常 | 2014年08月11日 11时26分... | org.springframework.dao.DataIntegrityViolationException | 严重 | 系统发生异常! | admin |
| 1105 系统异常 | 2014年08月11日 11时26分... | org.springframework.dao.DataIntegrityViolationException | 严重 | 系统发生异常! | admin |

图 3-28 告警历史查看

3.4 外部资源

3.4.1 资源管理

- 1) 外部资源是针对 WebLogic 下部署的数据源以及 WebLogic 的 JVM 运行状况进行监控。
在启动“监控控制台”服务前，需要将工程根目录下的 lib 文件夹下的“wlfullclient.jar”文件拷贝到 tomcat 的 lib 目录下。
- 2) 查看已创建的资源
进入监控控制台，选择“外部资源”导航菜单,选择“资源管理”子菜单，可以看到所有资源的状态，点击“” info 图标可以对资源进行编辑。

资源管理

| | | 监测状态 | 资源名称 | 资源URI | | 新增 | 删除 |
|--|--|------|--------------------|---|--|----|----|
| | | 已停止 | demo1 | weblogic/dataSource/JDBC Data Source-1:10.4.46.54:7001 | | | |
| | | 已停止 | ds1 | weblogic/dataSource/JDBC Data Source-0:10.4.44.130:7001 | | | |
| | | 已停止 | JDBC Data Source-2 | weblogic/dataSource/JDBC Data Source-2:10.4.44.130:7001 | | | |
| | | 已停止 | JDBC Data Source-1 | weblogic/dataSource/JDBC Data Source-1:10.4.44.130:7001 | | | |
| | | 已停止 | JDBC Data Source-0 | weblogic/dataSource/JDBC Data Source-0:10.4.44.130:7001 | | | |
| | | 已停止 | weblogic/jvm堆内存监视 | weblogic/jvm:10.4.44.130:7001 | | | |

新增、删除

图 3-29 资源管理

- 3) 新增 Weblogic DataSource
点击新增按钮，进入以下页面：

资源管理 > 资源编辑

基本信息

资源命名: WebLogic_DS_Demo *

资源描述: 数据源

连接设置

资源URI: weblogic/dataSource/JDBC Data Source-1:10.4.46.54:7001 *

用户名: weblogic

密码:

测试连接

帮助信息

监测设置

轮询周期 (ms): 5000 *

指标值保存周期 (s): 10 *

监测状态

监测开关: ☒ 开启 ☐ 关闭

监测状态: --

保存 返回

图 3-30 资源编辑

3.4.2 WebLogic DataSource 监控

选择“WebLogic DataSource 监控”子菜单，在右侧可以看到 WebLogic DataSource 监控视图。在“数据源监控项”下拉框中选择需要查看的资源，就可以看到所选资源的监控信息。



图 3-31 WebLogic DataSource 监控

点击保存为图片按钮会将图表保存为图片

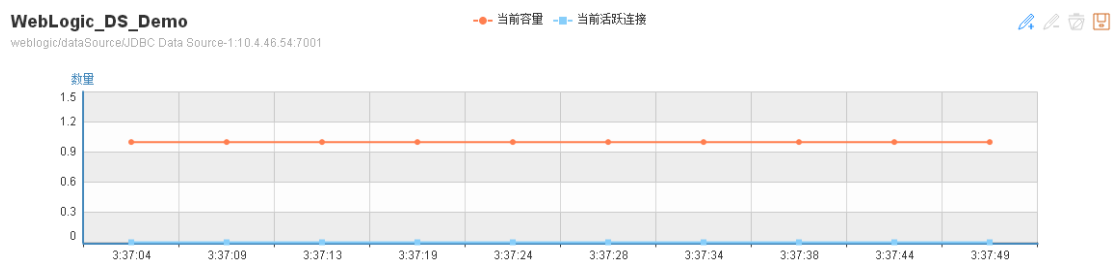


图 3-32 监控数据表保存为图片

3.4.3 WebLogic Jvm Heap 监控

WebLogic Jvm Heap 监控是对 WebLogic 下的 JVM 运行状况进行监控。选择“WebLogic JVM Heap 监控”子菜单，在右侧可以看到 WebLogic DataSource 监控视图。

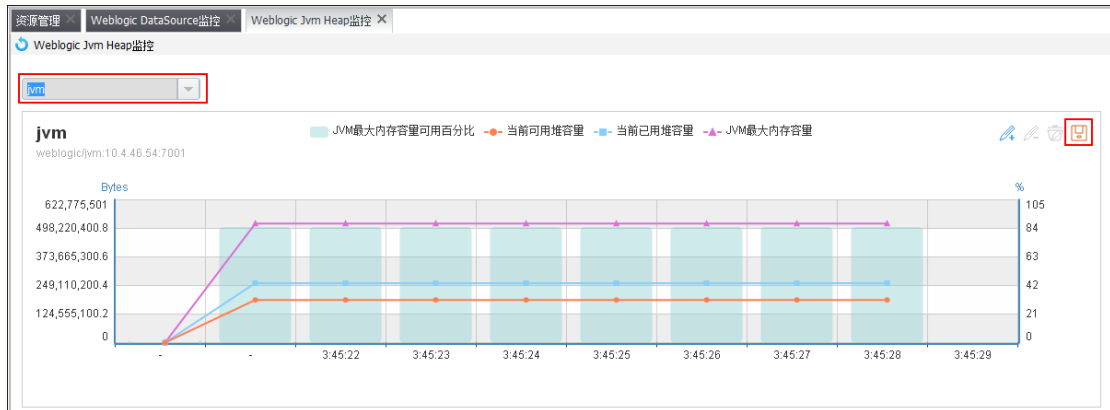


图 3-33 WebLogic Jvm Heap 监控

保存为图片：

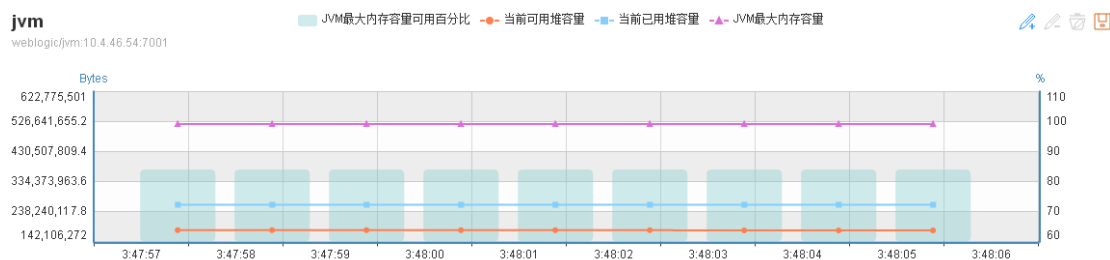


图 3-34 jvm 监控信息保存为图片

3.4.4 WebSphere 的下的 JVM 和数据源实时监控配置

监控前准备

- 1) 对 WebSphere 的下的 JVM 和数据源进行监控需要开启【全局安全性】下的【管理安全性】，用以防止那些未经授权的用户获取 WebSphere 运行时的信息。开启步骤详见[附录 c](#)。
- 2) 开启监控前，需要注意以下两点：
在安全模式先，监控后台获取监控数据时需要使用 WebSphere 的安全证书（服务器自带，无需自己生成），所以，确保以下目录中的以下证书存在。

{WebSphere_HOME}/AppServer/profiles/AppSrv01/config/cells/{cell 名}/nodes/{node 名}

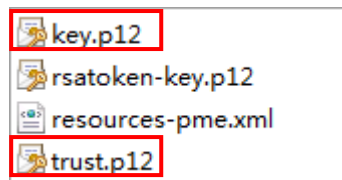


图 3.4.4-4 证书

- 3) 确保 PMI（性能监控基础结构）处于开启状态，开启方式详见[附录 E](#)

3.4.5 WebSphere 下的 JVM 的实时监控

- 1) 选择应用监控下的：外部资源→资源管理。

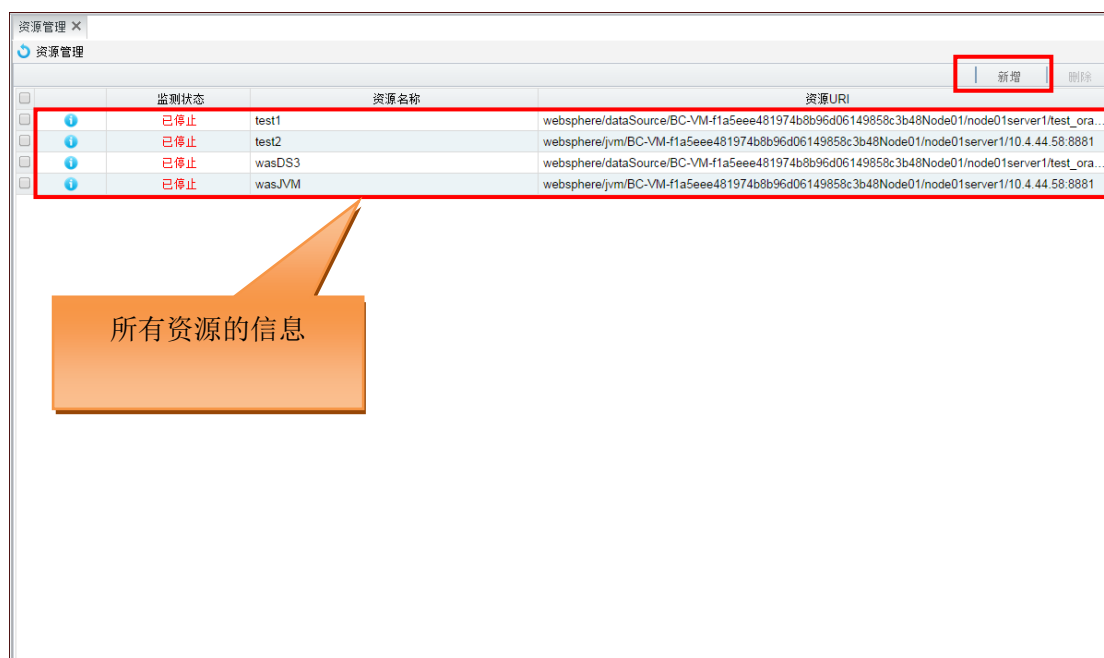


图 3.4.5-1 资源管理页面

- 2) 点击左上角的“新增”按钮,进入资源编辑页面。

注意事项:

- 【用户名】和【密码】是登录 WebSphere 控制台所使用的用户名和密码。

- 【帮助信息】，会提示各种资源的 URL 格式信息，直接复制、粘贴后修改即可。
- 填写完成后，点击【测试连接】按钮测试资源是否可用。测试成功后方可保存资源，否则，请检查输入的资源 URI、用户名、密码等是否正确。

资源管理 X

资源管理 > 资源编辑

基本信息

资源命名: *

资源描述:

连接设置

资源URI: *

用户名:

密码:

测试连接

帮助信息

监测设置

轮询周期 (ms): *

指标值保存周期 (s): *

监测状态

监测开关: ☒ 开启 ☐ 关闭

监测状态: --

保存 返回

基本信息,用户自定义

必填项, 格式等详见帮助信息

WebSphere 控制台登录时的用户名和密码

帮助信息

必填

监测开关选择开启

图 3.4.5-2 资源编辑页面

【帮助信息】



图 3.4.5-3 URL 帮助信息

- 3) 复制【WebSphere JVM 堆内存】下的格式信息，进行修改。端口信息查看见[附录 D](#)
 WebSphere JVM 堆内存 URL 示例: **websphere/jvm/node01/server01/10.4.44.58:8881**
- 4) 点击【测试连接】按钮。
 测试成功后点击【保存】按钮保存资源。



图 3.4.5-4 测试连接成功

测试失败后请检查 URI、用户名、密码是否全部正确。

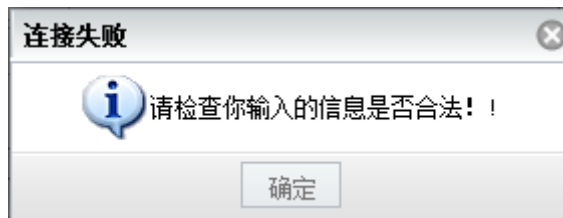


图 5-5 测试连接失败

- 5) 选择外部资源下的【WebSphere Jvm Heap 监控】子菜单。

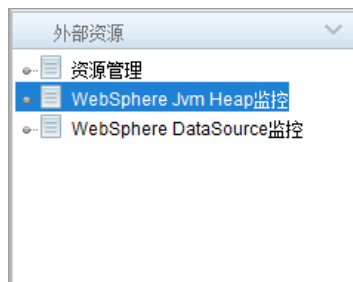


图 3.4.5-6 外部资源菜单

- 6) 选择刚才配置的资源。



图 3.4.5-7 WebSphere JVM Heap 监控选择页面

7) JVM 监控效果图

监控项：**JVM 最大内存容量**、**当前可用堆容量**、**当前已用堆容量**

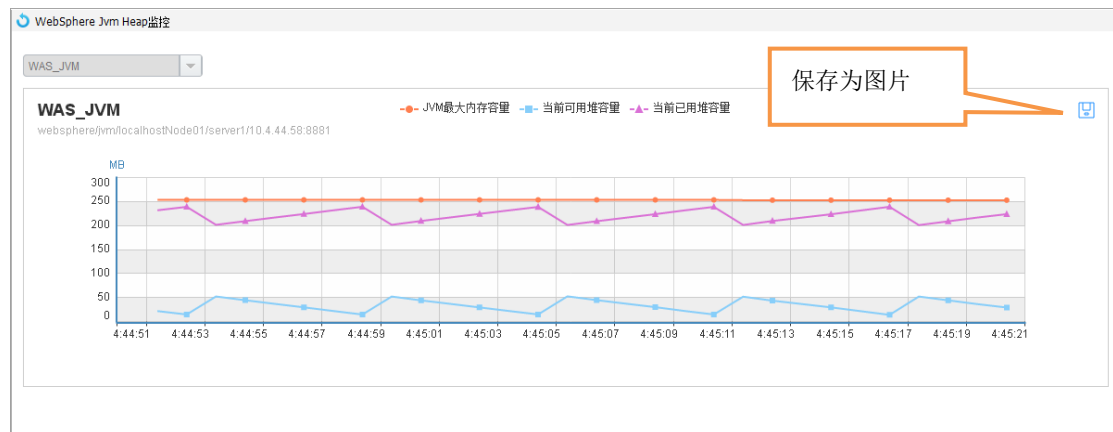


图 3.4.5-8 WebSphere JVM Heap 实时监控效果图

3.4.6 WebSphere 数据源实时监控

WebSphere 数据源实时监控基本步骤与 WebSphere 下的 JVM 的实时监控的操作步骤基本相同。二者最主要的区别是【资源编辑】页面下的【资源 URI】格式不同。

- 1) 选择应用监控下的：外部资源→资源管理。
- 2) 点击左上角的【新增】按钮，进入【资源编辑】页面，填写资源的相关信息。
- 3) 复制【WebSphere JVM 堆内存】下的格式信息，进行修改。

WebSphere 数据源监控 URL 示例：

websphere/dataSource/localhostNode01/server1/Oracle JDBC Driver/10.4.44.58:8881

- 4) 点击【测试连接】按钮进行连接测试。测试成功后点击【保存】。
- 5) 选择外部资源下的【WebSphere DataSource 监控】子菜单。

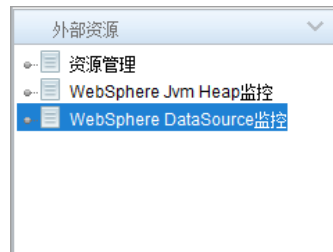


图 3.4.6-1 外部资源菜单

6) 选择刚才配置的资源。

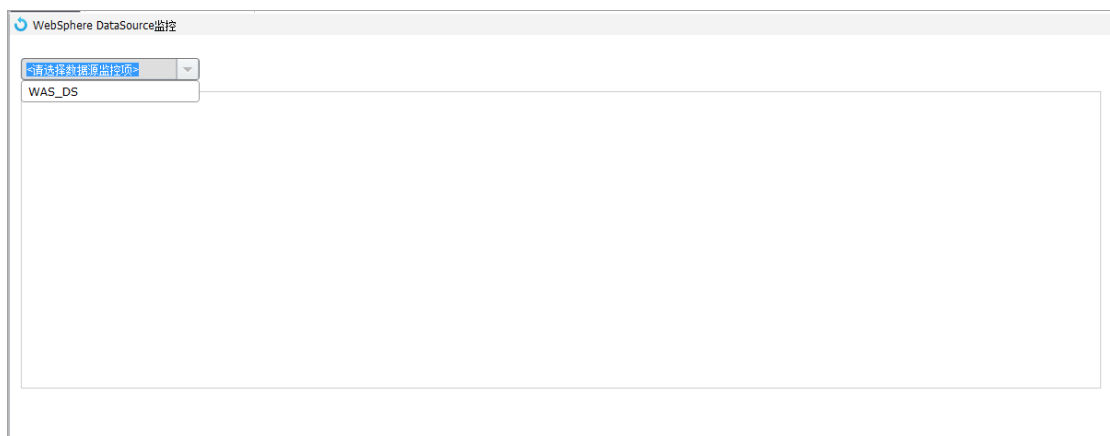


图 3.4.6-2 WebSphere DataSource 监控选择页面

7) DataSource 监控效果图

监控项：**创建的连接总数**、**连接池的大小**、**池中空闲的连接数**、**已关闭的连接总数**

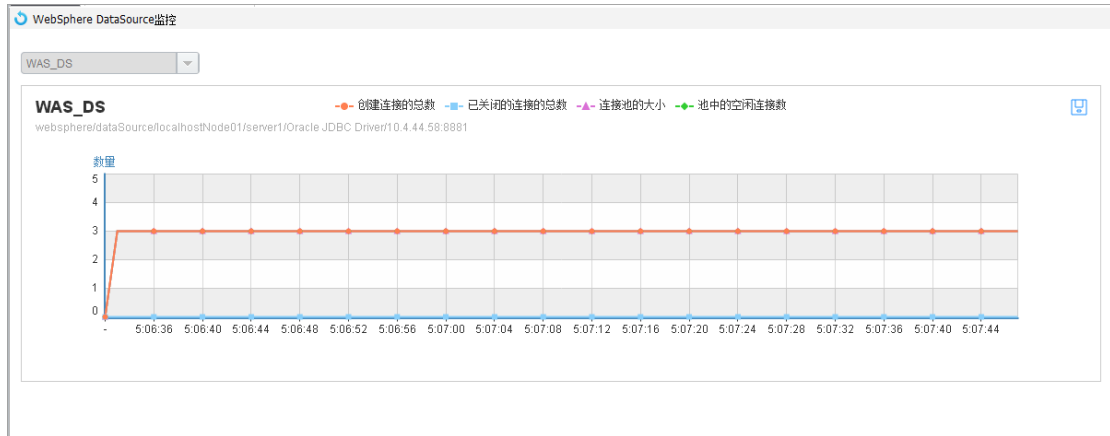


图 3.4.6-3 WebSphere DataSource 实时监控效果图

第4章 应用监控日常运维

本文第 2 章中介绍了不同的部署模式下，系统各组成部分的安装操作步骤。本章将从更宏观的视角，以各组成部分之间的联系为出发点，系统性地阐述关于启动、停机以及数据管理方面的内容。

4.1 应用监控启动

正如第 1 章中所介绍，整个监控体系中，逻辑上存在三种概念元素：监控代理、监控总线、监控控制台。这三种元素在形态上均是 JAVA Web 应用，故各孤立元素的启停操作，请参照所使用的 Web 应用服务器使用操作说明。

就三种元素间的相互影响关系来看，推荐的启动顺序是：

- 1、监控总线
- 2、监控代理
- 3、监控控制台

首先，应保证“监控总线”服务就绪，之后“监控代理”启动后，就可以在“监控总线”的管理界面上进行应用的注册审核操作。最后，启动“监控控制台”应用，在已审核通过的应用中，有选择性地信息进行订阅操作，订阅完成后，“监控总线”将会按照订阅关系，将监控数据转发到对应的“监控控制台”上。

若非按照上述说明进行，也不会带来太大影响。因为三种元素间没有强依赖关系，任何

一种元素出现不可用状况，不会牵连系统中的其它部分。对此，做出以下枚举说明：

1、当监控总线不可用时：

监控代理自动感知总线不可用状态，自动注销注册状态，停止监控数据收集，并且按照配置设定，周期性检查监控总线的状态，试图重新建立连接。

监控控制台不再收到监控信息，它记录的应用活跃时间一直停留在监控总线发生故障的时刻，该时刻供监控操作人员参考判断。

2、当监控代理不可用时：

监控总线不再收到代理的推送信息，它记录的代理活跃时间一直停留在代理发生故障的时刻，该时刻供总线操作人员参考判断。

监控控制台不再收到来自故障代理的推送信息，它记录的代理活跃时间一直停留在监控代理发生故障的时刻，该时刻供监控操作人员参考判断。

3、当监控控制台不可用时：

监控总线自动感知监控控制台不可用状态，降低对其消息转发优先级。总线管理员可以在管理界面上手动解除该订阅关系。

4.2 应用监控停机

关于应用监控停机，从系统健壮性角度来讲，由于之前所述原因（各元素间无强依赖），并无硬性规定。

从数据完整性角度来讲，推荐的优雅停机方案为：

- 1、关掉监控代理的所有监控项目。（低版本中未开放该功能接口，请联系服务支持人员了解具体情况）
- 2、待监控控制台中应用活跃时间停止更新后，解除订阅关系。
- 3、在监控总线管理界面中，解除应用注册。

如果想继续彻底停掉所有服务的话，请参照具体应用服务器的操作手册，彻底关闭所有应用。

4.3 监控数据管理

应用监控数据对于有着“数据驱动”高级需求的用户来讲，是一笔宝贵的资产。关于“监控数据管理”方面的详细介绍请联系服务支持人员获取《监控数据归档操作说明书》。

附录 A MongoDB 的安装部署

应用监控数据持久化方案采用了 MongoDB，更准确地说是采用了 TokuMX 发行版。对于该数据库产品，UniEAP 未做任何定制化工作，仅仅是单向依赖标准的 Mongo 服务。所以用户可以自由选取任何一种 Mongo 服务形式，比如说选择 MongoDB 官方发行版本而弃用 TokuMX，再比如说根据具体需求任意选择使用单实例服务或者集群服务。关于 MongoDB 与 TokuMX 更多信息，请访问官方网站获取最新、最详细的资料。

为了方便用户快速部署，现提供 Tokumx 单实例相关安装文件包，以及安装过程指引说明以供参考。

前提准备：64 位版本 Linux 操作系统。

步骤 1：通过服务支持人员获取“TokuMX 部署安装包.zip”。该资料文件名称有可能会发生变动，以具体交付时为准，但资料数量与对应意义应该是一致的。

步骤 2：将安装包解压到系统文件目录：“/usr/local/tokumx”。解压完成后，目录结构内容如下图所示：

```
[root@BC-VM-f1a5eee481974b8b96d06149858c3b48 tokumx]# pwd
/usr/local/tokumx
[root@BC-VM-f1a5eee481974b8b96d06149858c3b48 tokumx]# ll
total 148
drwxr-xr-x. 2 root root 4096 Aug 14 16:01 bin
drwxr-xr-x. 2 root root 4096 Aug 14 16:20 conf
drwxr-xr-x. 2 root root 4096 Aug 15 09:20 db
-rw-r--r--. 1 root root 34520 Jun 17 05:31 GNU-AGPL-3.0
drwxr-xr-x. 2 root root 4096 Aug 14 16:01 lib64
drwxr-xr-x. 2 root root 4096 Aug 14 16:41 log
-rw-r--r--. 1 root root 58931 Jun 17 05:31 NEWS
-rw-r--r--. 1 root root 1279 Jun 17 05:31 README
-rw-r--r--. 1 root root 3714 Jun 17 05:31 README-TOKUKV
drwxr-xr-x. 2 root root 4096 Aug 14 16:01 scripts
-rw-r--r--. 1 root root 16787 Jun 17 05:31 THIRD-PARTY-NOTICES
[root@BC-VM-f1a5eee481974b8b96d06149858c3b48 tokumx]#
```

步骤 3：修改启动参数配置文件“/usr/local/tokumx/conf/tokumx.cfg”中的“bind_ip”属性值，将其改为实际值。

```
[root@BC-VM-f1a5eee481974b8b96d06149858c3b48 tokumx]# cat conf/tokumx.cfg
fork=true
bind_ip=172.16.0.242,127.0.0.1
dbpath=/usr/local/tokumx/db
logpath=/usr/local/tokumx/log/server1.log
logappend=true
journal=true
```

步骤 4：在“/usr/local/tokumx”目录下，运行命令“./bin/mongod -config ./conf/tokumx.cfg”启动 mongod 服务。

步骤 5：至此 Mongo 服务已经可用。

附录 B Solr 的安装部署

应用监控日志分析部分是通过 Solr 而非 MongoDB 来构造日志索引数据，由于 Solr4.7 以上的版本对 JDK 的要求是 1.7 及以上，故本文档中所讲到的 Solr 安装都是基于 4.7 版本的。

为方便用户快速部署，现提供 Solr 单实例相关安装文件包，以及安装过程指引说明以供参考。

前提准备：64 位版本 Linux 操作系统。

步骤 1：通过服务支持人员获取“Solr 安装包.zip”。该资料文件名称有可能会发生变动，以具体交付时为准，但资料数量与对应意义应该是一致的。

步骤 2：在 usr 目录下创建 solr 文件夹，将 solr 安装包解压到该文件夹下，解压后该文件夹下包含两个文件夹，tomcat 和 solr：

步骤 3：在“/usr/solr/tomcat/bin”目录下，运行命令“./startup.sh”。

步骤 4：至此 Solr 服务已经可用，访问 <http://ipaddress:8080/solr> 后就可以看到 solr 成功启动，如果 8080 端口被占用，请参照 tomcat 修改端口的方式进行修改即可。

附录 C 如何开启管理安全性

在未启用安全性的状态下，用户登录 WebSphere 控制台不需要用户名和密码，这意味着任何人都可以登录 WebSphere 的管理控制台修改配置或进行其他操作，并且多人同时修改配置并保存时，往往会发生操作丢失和冲突问题。这种状态下明显不利于开发和使用。所以需要开启安全性以阻止未经授权用户登录系统。下面介绍如何开启基本的【管理安全性】。

- 1) 登录 WebSphere 管理控制台，如果登录时不需要登录验证，则进行【下一步】，否则无需进行以下修改。
- 2) 进入 WebSphere 管理控制台，打开【安全性】菜单。



图 C -1 WebSphere 安全性菜单

- 3) 点击【全局安全性】子菜单,勾选【启用管理安全性】，点击【应用】后保存修改，结果如下：



图 C-2 全局安全性管理页面

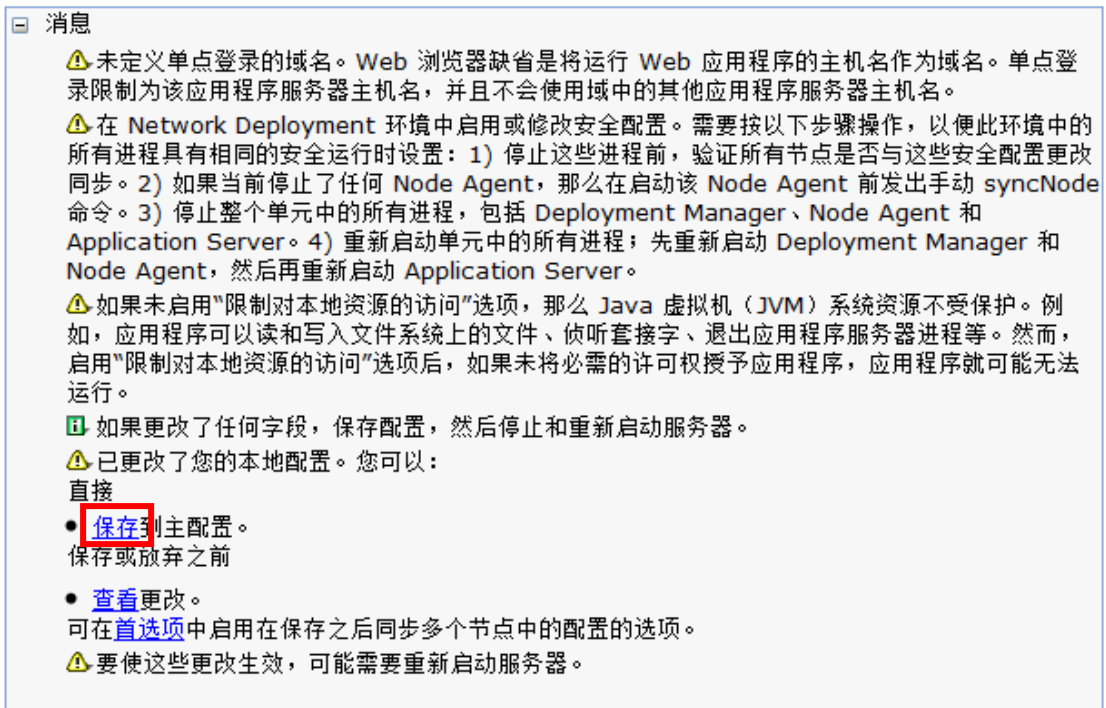


图 C-3 保存修改

- 4) 重启 WebSphere，登录控制台，此时用户登录时需要进行登录验证，管理安全性开启成功。



图 C-6 安全登录

附录 D 如何查看服务器的 SOAP 端口号

- 1) 打开管理控制台，选择【服务器】→【WebSphere Application Server】

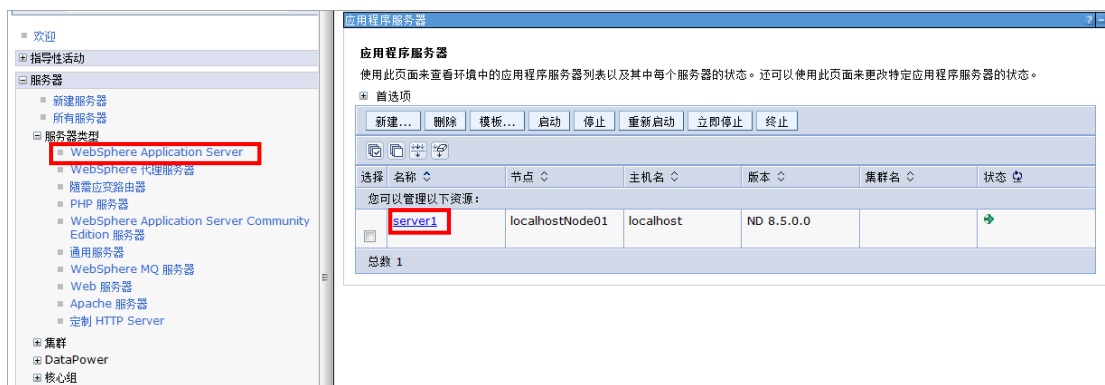


图 D-1 应用服务器

- 2) 点击【server1】，进入到应用服务器配置页面。

应用程序服务器 > server1

使用此页面来配置应用程序服务器。应用程序服务器是提供运行企业应用程序所需服务的服务器。

运行时

配置

报告

操作

常规属性

名称

server1

节点名

localhostNode01

☐ 以开发方式运行

☒ 并行启动

☐ 在需要时启动组件

访问内部服务器类

允许

特定于服务器的应用程序设置

类装入策略

多个

类装入方式

类已装入并且是先使用父类装入器

应用

确定

复位

取消

容器设置

会话管理

SIP 容器设置

Web 容器设置

Portlet 容器设置

EJB 容器设置

容器服务

业务流程服务

应用程序

已安装的应用程序

服务器消息传递

消息传递引擎

消息传递引擎入站传输

WebSphere MQ 链路入站传输

SIB 服务

服务器基础结构

Java 和进程管理

管理

Java SDK

通信

端口

图 D-2 应用服务器配置页面

- 3) 点击展开右下角【通信】栏下的【端口】，此时会列出所有的端口信息。

通信

端口

SOAP 端口为: 8881

| 端口名 | 端口 | 详细信息 |
|---------------------------------------|-------|------|
| BOOTSTRAP_ADDRESS | 2810 | |
| SOAP_CONNECTOR_ADDRESS | 8881 | |
| ORB_LISTENER_ADDRESS | 9102 | |
| SAS_SSL_SERVERAUTH_LISTENER_ADDRESS | 9409 | |
| CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS | 9408 | |
| CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS | 9407 | |
| WC_adminhost | 9062 | |
| WC_defaulthost | 9081 | |
| DCS_UNICAST_ADDRESS | 9355 | |
| WC_adminhost_secure | 9045 | |
| WC_defaulthost_secure | 9444 | |
| SIP_DEFAULTHOST | 5063 | |
| SIP_DEFAULTHOST_SECURE | 5062 | |
| SIB_ENDPOINT_ADDRESS | 7278 | |
| SIB_ENDPOINT_SECURE_ADDRESS | 7287 | |
| SIB_MQ_ENDPOINT_ADDRESS | 5559 | |
| SIB_MQ_ENDPOINT_SECURE_ADDRESS | 5579 | |
| IPC_CONNECTOR_ADDRESS | 9634 | |
| OVERLAY_UDP_LISTENER_ADDRESS | 11009 | |
| OVERLAY_TCP_LISTENER_ADDRESS | 11010 | |

图 D-3 端口列表

附录 E 启用性能监控基础结构（PMI）

- 1) 进入 WebSphere 管理控制台，打开【监视和调整】菜单。



图 E-1 监视和调整菜单

- 2) 选择【性能监控基础结构（PMI）】



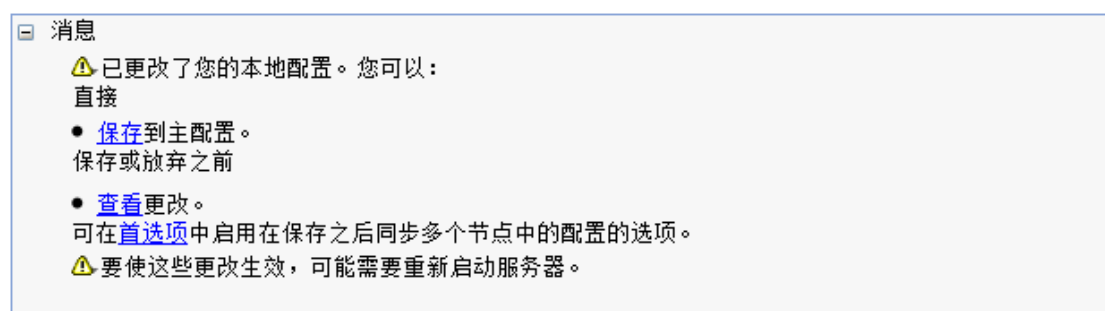
图 E-2 性能监控基础结构资源管理

- 3) 点击资源【server1】，进入性能监控基础结构（PMI）配置页面。
勾选【启用性能监控基础结构】，选择【当前监视的统计信息集】为【基本】，点击【应用】按钮应用配置。



图 E-3 性能监控基础结构 (PMI) 配置页面

4) 点击【保存】，保存所有配置。



附录 F SQL 监控支持 JNDI 数据源配置

用服务器提供的 JNDI 数据源后，原来的数据库配置文件：Eapconfig.properties 不再使用。不同服务器的数据源配置方式不同，其中 Tomcat 服务器的 JNDI 数据源与 WebSphere 以及 WebLogic 服务器有很大区别，请注意区分。

下面介绍如何修改配置文件以使得 Sql 监控支持 JNDI 数据源：

步骤1、 建立数据源，设置数据源的 JNDI 名称，此名称是区别于其他 JNDI 的唯一标识。本例中，数据源的后台数据库是 Oracle，数据源的 JNDI 名称是：**jdbc/oracle**。

注：若应用部署在 **WebSphere** 的集群上时，在创建数据源时，须选择数据源的作用域为整个【集群】。

步骤2、 打开工程的数据源配置文件：applicationContext-datasource.xml

文件位置：**platform/core/content/conf/ applicationContext-datasource.xml**

a) 将以下部分注释掉或者删除：

```
<!-- 定义系统公共配置 -->
<bean id="propertyConfigurer"
      class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
  <property name="locations">
    <list>
      <value>WEB-INF/conf/platform/core/EAPConfig.properties</value>
    </list>
  </property>
</bean>

<!-- 定义系统数据源 -->

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"
      destroy-method="close">
  <property name="driverClassName">
    <value>${unieap.datasource.driverClass}</value>
  </property>
  <property name="url">
    <value>${unieap.datasource.jdbcUrl}</value>
  </property>
  <property name="username">
    <value>${unieap.datasource.user}</value>
  </property>
  <property name="password">
    <value>${unieap.datasource.password}</value>
  </property>
  <property name="initialSize">
    <value>${unieap.datasource.initialSize}</value>
  </property>
  <property name="maxActive">
    <value>${unieap.datasource.maxActive}</value>
  </property>
  <property name="maxIdle">
    <value>${unieap.datasource.maxIdle}</value>
  </property>
  <property name="maxWait">
    <value>${unieap.datasource.maxWait}</value>
  </property>
</bean>
```

- b) 在文件中添加以下内容，红色部分替换为在**步骤 2**中配置的数据源的 JNDI 名称。
- 当应用部署在 WebLogic 或 WebSphere 服务器上时，增加以下代码：

```
<!--JNDI数据源配置(红色部分为需要注意的地方)-->
<beanid="dataSourceTarget"class="org.springframework.jndi.JndiObjectFactoryBean"
">
    <!--配置外部数据源的JNDI名称（在WebLogic和WebSphere服务器下） -->
    <propertyname="jndiName"value="jdbc/oracle"/>
    <propertyname="lookupOnStartup"value="false"/>
    <propertyname="cache"value="true"/>
    <propertyname="proxyInterface"value="javax.sql.DataSource"/>
</bean>

<!-- JDBC Driver代理 -->
<beanid="dataSource"class="com.neusoft.unieap.monitor.jdbc.monitor4jdbc.datasou
rce.UniDataSource">
    <constructor-arg>
<reflocal="dataSourceTarget"/>
</constructor-arg>
</bean>
```

- 当应用部署在 Tomcat 服务器上时，增加以下代码：

```
<!--JNDI数据源配置(红色部分为需要注意的地方)-->
<beanid="dataSourceTarget"class="org.springframework.jndi.JndiObjectFactoryBean"
">
    <!--配置外部数据源的JNDI名称（在Tomcat服务器下） -->
    <propertyname="jndiName"value="java:comp/env/jdbc/oracle"/>
    <propertyname="lookupOnStartup"value="false"/>
    <propertyname="cache"value="true"/>
    <propertyname="proxyInterface"value="javax.sql.DataSource"/>
</bean>
```

```
<!-- JDBC Driver代理 -->
<beanid="dataSource"class="com.neusoft.unieap.monitor.jdbc.monitor4jdbc.datasou
rce.UniDataSource">
    <constructor-arg>
<reflocal="dataSourceTarget"/>
</constructor-arg>
</bean>
```

步骤3、 修改方言包,将方言包替换为当前数据库对应的 Hibernate 方言包的路径(Hibernate 方言包在 hibernate-xxxx.jar 的 org.hibernate.dialect 包下)。

```
<beanid="sessionFactory"
class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
    <propertyname="dataSource"ref="dataSource"/>
    <propertyname="mappingDirectoryLocations">
        <list>
            <value>/WEB-INF/conf/**/hbm</value>
        </list>
    </property>
    <propertyname="hibernateProperties">
        <props>
            <!-- 注意此处方言包配置要与数据库类型相对应 ,本例中使用的是oracle的方言包-->

            <propkey="hibernate.dialect">org.hibernate.dialect.Oracle9Dialect</prop>
            <propkey="hibernate.show_sql">>false</prop>
            <propkey="hibernate.format_sql">>true</prop>
        </props>
    </property>
    <propertyname="lobHandler">
        <reflocal="oracleLobHandler"/>
    </property>
</bean>
```

步骤4、 重启应用。

