文档编码	
密级	
文档版本	
拟制人	DNA
日期	

久其研发与应用平台
（DNA）
逻辑框架
概述

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

更新说明

日期	版本	更新说明	作者	备注

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

前言

1. 预期读者

该文档主要提供给使用 DNA 平台的开发人员。

2. 阅读指导

该文档包含了 DNA 逻辑框架的结构和运行机制。其目的是希望开发人员能够通过该文档了解 DNA 逻辑框架的功能特性和设计思想，并为深入学习各功能模块做铺垫。

文档主要包含的内容有：

第一章简要介绍 DNA 系统的整体结构，明确几个关键的概念。

第二章以应用的启动为线索，介绍应用启动的流程，机制及相关规范。

第三章以站点的启动为线索，介绍站点的结构，及相关细节。

以上三章属于框架基础知识的介绍，是理解元数据的注册与管理机制的基础。

第四章介绍框架的一些基本功能特性。

3. 术语

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

DNA 逻辑框架简介

1. 职责定位

DNA 平台由逻辑框架、界面框架和基础应用平台三部分构成。作为整个 DNA 应用的基础，逻辑框架的职责包括但不限于：

- ✓ 管理应用程序的生命周期
- ✓ 集成界面框架和基础应用平台
- ✓ 提供模块扩展机制
- ✓ 解决业务无关的技术性问题
- ✓ 为业务应用提供优化的处理模式
- ✓ 提供高效的基础功能接口

2. 功能特性

- ✓ 使用 OSGI 管理功能模块
- ✓ 以 Jetty7 为基础的 Web 容器，支持 Servlet 应用
- ✓ 以数据（消息）为接口的松耦合调用机制
- ✓ 完善的缓存管理机制，支持内存事务、多索引等特性
- ✓ 针对 Oracle、DB2、MSSQLServer、MySQL 和 Kingbase 数据库进行适配，提供统一的数据访问接口
- ✓ 自动化的事务管理机制，保障数据库和缓存的一致性
- ✓ 内置会话管理机制，结合界面框架实现对用户活动生命周期的管理
- ✓ 利用针对缓存的权限控制机制，业务层通过简单封装即可实现用户权限管理功能
- ✓ 完善的信息报告机制适用于业务日志、错误记录、本地化等多种应用场景
- ✓ 提供详细的性能指标，用于监控和分析系统运行状况
- ✓ 通过收集器机制，实现元数据和扩展模块的信息收集和注册
- ✓ 集成 WebService 发布框架，简化了 WebService 开发过程

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

第一章　框架体系结构概述

1. 整体结构

DNA 应用整个结构如图：

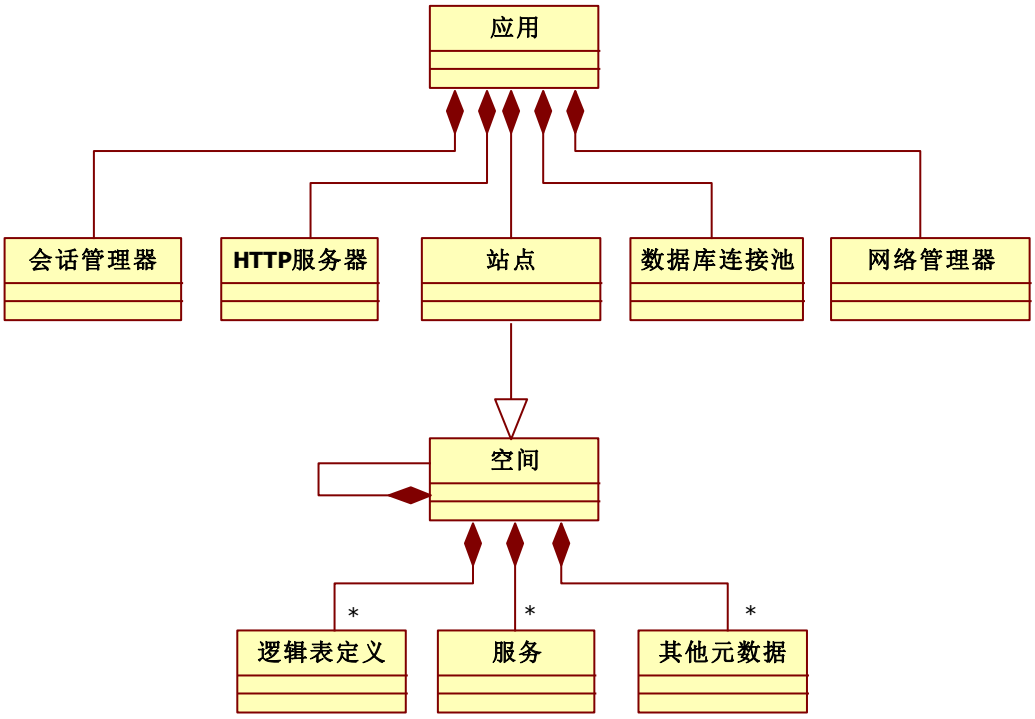


图 DNA 整体结构

2. 应用 Application

应用指 DNA 服务的运行实例，是功能模型与站点的容器，并提供了众多的基础服务功能。应用在 DNA 框架内，是处于最高层次的对象。每个 Java 虚拟机最多只支持启动一个应用实例。

每个应用都至少拥有一个站点。

应用的标识为 GUID 类型，我们称之为 appID。通常情况下 appID 是在系统运行时随机生成的，所以每个应用实例的 appID 都不同。在集群部署模式下，应用退化为一种逻辑概念。我们将集群中的 DNA 进程称为节点，不同节点上的应用实例在逻辑层面是同一个应用，这种情况下，我们需要将这些应用实例的 appID 设为相同的值，将 dna-server.xml 中的<cluster>标记的 id 属性设为相同值既可，系统会根据该值计算出 appID。

3. 空间 Space

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

空间是功能模块的容器，空间下可以包含子空间，从而形成树型结构。空间树只有一个根节点，称为站点(Site)。空间树的叶子节点称为服务(Service)。虽然功能模块在 Java 层面上平行的，但是在业务架构上我们利用空间将功能模块进行隔离，每个业务模块都从属于一个空间。可以将空间理解为文件夹，功能模块类似为文件，站点为根目录。将功能模块进行这样的逻辑划分的目的是实现业务功能的隔离。所以与 Java 语言的包类似，服务也具有可见性控制。

站点/空间/服务是框架提供功能的基本组织形式。

4. 站点 Site

站点向应用提供各种功能模块，从而应用才可以提供出完整的功能和服务。

站点还负责事务管理、缓存管理等职能。

举个形象的比喻：应用好比一家电影院，有各种放映设备——基础服务，站点好比电影的胶片——真正的内容所在，胶片拿到电影院才能放出一场完整的电影——完整业务服务。

5. 服务 Service

良好设计的业务模块其功能接口应当以服务的形式公开出来。服务能够提供多种类型的接口，包括：任务处理器(TaskHandler)、提供器(Provider)、监听器(Listener)、WebService 方法(WebMethod)、单元测试用例(CaseTester)，以后可能还会扩展更多接口。我们将前三种接口统称为处理器(Handler)。

我们怎样将业务功能封装为服务呢？

业务模块可以抽象为数据处理过程，业务模块之间相互调用实现了数据流转。那么我们可以将数据封装为任务(Task)，将数据处理过程封装为任务处理器。如果是以查询为主的应用就可以将查询逻辑封装为提供器。在绝大部分情况下，我们都可以使用服务提供的接口模式封装业务功能，从而实现功能的发布。服务的开发人员需要掌握服务接口的特点，以选择最合适的封装方式。

为什么使用服务？

首先，使用服务可以降低模块之间的耦合度。服务的设计类似消息系统，服务与服务或者其他模块是通过数据（任务、主键、事件参数等）进行交互的，类似消息传递过程。所以将功能封装为服务，只需要公开数据实体接口就可以了。

其次，处理器应当被设计成无状态的，可重入的，这样有利于模块并行的执行。

再次，以服务的形式来组织功能，易于对系统中的功能进行统一管理。

6. 处理器 Handler

应用内部，几乎所有功能逻辑的实现都是通过调用相应的处理器来完成的，包括同步或异步的执行任务(Task)，请求缓存资源(Resource)，请求注册元数据等。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

开发者可以利用上下文(context)上的接口与处理器进行交互。类似于方法调用，在一次交互过程中，处理器相当于被调用者。处理器应当根据传入参数执行适当的功能，如保存单据、执行查询、更新缓存等。处理器可以继续调用其他处理器，从而形成处理器的调用栈。

由于框架内部所有的调用都简化成了有限的几个接口，即方法的行为完全由参数决定，这样设计完全打破了系统内部各种接口上的依赖，使整个系统更易于维护和管理。关于处理器的开发请参考调用框架。

7. 上下文 Context

context 代表一次完整的请求，它提供了调用框架、缓存、数据访问等多种用途的接口，其的功能可以归结为以下几个方面：

- ✓ 调用处理器
  - ➢ 支持调用任务处理器、提供器（缓存和非缓存）、监听器
  - ➢ 允许同步和异步调用
- ✓ 访问数据库
  - ➢ 支持对象模型(DOM)
  - ➢ 支持 DNASQL
- ✓ 隐式管理事务
- ✓ 登录和权限验证
  - ➢ 获取登录信息
  - ➢ 验证登录用户/指定用户的权限
  - ➢ 切换登录用户
- ✓ 获取站点信息
- ✓ 发起远程调用
- ✓ 信息报告
  - ➢ 报告提示、警告、错误信息
  - ➢ 查询、报告工作进度
  - ➢ 获取本地化文本

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

第二章　应用启动

本章以应用的启动为线索，介绍应用启动的相关细节，并展示完成的应用启动流程。

1. 流程概述

标准 DNA 应用是基于 OSGi 平台的。从启动到 OSGi 环境到启动应用需要经过以下三个步骤：

图 DNA 启动顺序

启动 OSGi 平台即，从 java 环境启动 equinox 环境。其中 equinox 是一个 OSGi 规范的实现，并且作为 eclipse 的一个子项目用作 eclipse IDE 的底层运行时架构。

第二步，主要是安装应用需要的 Bundle 到 equinox 环境，包括 DNA 目录下的框架和业务相关的所有 jar 包，然后调用 core 包的 Activator 进入第三步。这部分工作由 dna 的 launcher 包负责完成。

第三步应用启动本身，包括了：读取应用配置参数，初始化各个基础服务器及管理器，读取各 Bundle 的 dna 配置文件，实例化站点，并根据收集的配置参数启动站点，从而完成应用启动的整个流程。

应用启动大致包含了以下的步骤：

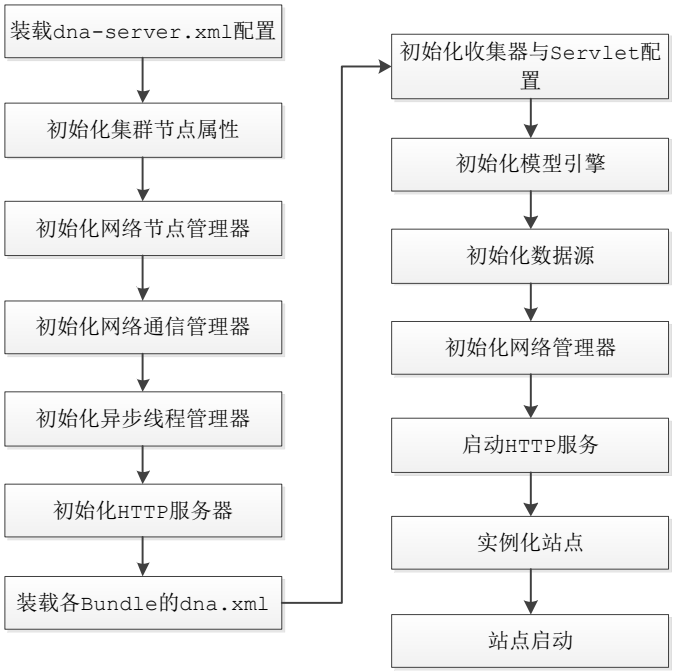


图 DNA 启动步骤

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

以上各种管理器或服务器都属于应用的基础服务，相关内容及细节对开发者来说是不可见的，所以不会详细介绍。以下主要介绍：

- ✓ 装载 dna-server.xml 配置，在装载应用配置文件一节中介绍。
- ✓ 装载各 Bundle 的 dna.xml，在装载 dna 配置文件一节介绍。
- ✓ 数据源相关，在初始化数据源及数据源支持两节中介绍。
- ✓ 收集器相关，在收集器一节中介绍。

此外，第三章会更加详细的介绍站点启动的流程，站点启动也包括了框架的开发者和使用者大部分最主要需要关心的问题。

2. 目录结构

一个标准的 DNA 应用的目录结构如下：

```
dna_root/                应用根目录
    app/                 业务模块
    dna/
        bap/             bap模块
        component/       组件库
        core/            逻辑框架
        ui/              界面框架
    lib/                 JQ库
    thr/                 第三方库
        com.jiuqi.dna.launcher_1.0.0.jar    dna启动模块
    work/
        dna-server.xml   配置文件
    jre/                 jre
```

3. 引导程序

DNA 应用启动是从 com.jiuqi.dna.launcher_1.0.0.jar(以下简称 launcher)开始的。这个程序首先启动 OSGi，然后尝试将 DNA 应用根路径及其子路径下的所有 jar 文件作为 Bundle 装载到 OSGi 平台中，接下来的启动过程由逻辑框架来完成。逻辑框架从参数 com.jiuqi.dna.rootpath 读取应用的根路径，然后读取 work 路径下的 dna-server.xml 文件。需要注意的是，在 eclipse 开发环境中，OSGi 的引导是由 eclipse 来完成的，所以不应当加载 launcher 包，即使加载了 launcher 包在调试时也不是由 launcher 程序启动的。launcher 程序仅用于引导发布的 DNA 应用启动。开发人员可以在 eclipse 环境下使用 com.jiuqi.dna.rootpath 参数切换配置文件路径，但是由于 launcher 会重写该参数，所以不能在生产环境中使用该参数。

4. 装载应用配置文件

dna-server.xml 文件，即**应用配置文件**，包含了应用的核心配置。该文件必须存放在 work 目录下。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

应用配置文件包括了应用数据源，HTTP 服务，远程调用，站点，集群定时重启策略等基础服务的配置。示例如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<dna>
    <!--数数据库连接信息-->
    <datasources>
        <datasource
            name="bap"
            location="jdbc:oracle:thin:@localhost:1521:bap"
            user="bap"
            password="bap"
            login-timeout-s="20"
            command-timeout-ms="0"
            max-connections="20"
            min-connections="5"
            idle-life-ms="120000"
        />
    </datasources>
    <!--Web容器信息-->
    <http
        context=""
        max-threads="100"
        http-sessions="true">
        <listen
            host=""
            port="9797"
            max-threads="500"
            accept-queue-size="50"
            acceptors="2"
            ajp13="false"
        />
    </http>
    <session timeout-m="0" heartbeat-s="300"/>
    <!--站点配置-->
    <sites>
        <site name="root">
            <datasource-refs>
                <datasource-ref space="dna/bap" datasource="bap"/>
            </datasource-refs>
        </site>
    </sites>
    <!--重启策略配置-->
    <reboot>
        <weekstrategy
```


公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

```
            weekday="0"
            reboot-time="00:00:00"
            delay-time-m="30"
            session-floor="10"
        />
        <daystrategy
            cycle="3"
            reboot-time="00:00:00"
            delay-time-m="30"
            session-floor="10"
        />
    </reboot>
    <!—集群配置-->
    <cluster id="dna1" index="1">
        <node url="http://localhost:9901"/>
        <node url="http://localhost:9902"/>
    </cluster>
</dna>
```

为方便开发者使用，我们为 dna-server.xml 文件定义了一个 schema：dna-server.xsd。如果要使用这个 schema 文件，需要在 dna-server.xml 的 dna 元素上增加三个属性，例如：

```
<?xml version="1.0" encoding="UTF-8"?>
<dna
    xmlns="http://www.jiuqi.com.cn/dna"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.jiuqi.com.cn/dna dna-server.xsd">
```

以上示例中仅给出了逻辑框架部分用到的配置，dna-server.xml 文件还被 DSM（DNA Server Manager）和其他模块使用，所以可能还包含其他配置信息。

1) 数据源配置

datasources 元素用于配置应用使用的数据源。每一 datasource 元素代表一个数据源配置。应用配置文件中应该至少包含一个有效的数据源配置，也可以配置多个数据源。其中第一个数据源默认作为站点的数据源。

属性名称	单位	默认值	说明
name	无	必填	数据源名称，存在多个数据源配置时，名称不能重复，必填属性。
location	无	必填	jdbc 的数据库连接地址。格式参见“数据源支持“一节。
user	无	必填	数据库连接的用户名。
password	无	必填	数据库连接的密码。
max-connections	无	20	最大连接数

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

min-connections	无	5	最小连接数
idle-lifetime-ms	毫秒	120000	闲置连接的保质期
login-timeout-s	秒	20	连接超时时间，秒单位
command-timeout-s	秒	0	数据库命令执行超时时间，0 表示即无限制。

2) HTTP 服务配置

http 元素用于配置网络服务的监听属性，http 元素本身的属性对所有监听有效。

属性名称	单位	默认值	说明
context	无	空	设置 Servlet 的 URL 根路径，默认的根路径即站点的根路径。例如，将应用部署在/dna/路径下，则可以设置 context="/dna"，**开头必须是“/”，并且结尾不能有“/”**
max-threads	无	100	设置共享线程池的最大线程数。系统提供一个共享线程池，如果 listen 元素上指定 max-threads，则为该监听分配一个独占的线程池，否则使用共享线程池。
http-sessions	无	true	是否启用 HttpSession，值为 true 或 false。

http 每个子元素 listen 都为一个端口监听的配置。

属性名称	单位	默认值	说明
host	无	空	监听 socket 绑定的地址。空表示监听所有地址，相当于 localhost。
port	无	9797	监听端口。
max-threads	无	0	独占线程池的最大线程数。如果 max-threads>0 则系统为该监听分别独占的线程池，否使用共享的线程池。
accept-queue-size	无	1000	等待处理的连接请求的最大数量。达到该数量后，再有新的请求，服务器会拒绝处理，客户端将得到 http 错误“503 Service Unavailable”。
acceptors	无	CPU 核数	监听执行接受请求操作的线程数量。该参数取值范围从 1 到 CPU 核数，如果指定了比 1 小的值则当作 1 处理，比 CPU 核数大的值当作 CPU 核数处理。
ajp13	无	false	是否用 ajp13 协议而不用 http 协议，默认用 http 协议。
ssl	无	false	是否启动 SSL 协议。该参数仅对 http 协议有效。默认不使用 SSL 协议。关于 SSL 的配置请参考如何配置 SSL 端口。
ssl-keystore	无	keystore.jks	密钥文件的名称，该文件应当放在 work 目录下。
ssl-key-password	无	必填	明文或密文的密码，密文密码是由明文密码混淆后产生的，以“OBF:”起始。
ssl-password	无	必填	明文或密文的密码。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

ssl-keystore-type	无	JKS	密钥文件的类型。

3) 会话配置

session 元素用于配置会话相关属性。

属性名称	单位	默认值	说明
timeout-m	分钟	0	定义会话过期时间，即客户端在指定时间没有向服务器发送心跳以外的请求，则判断会话过期。0 表示会话永不过期。
heartbeat-s	秒	300	连续两次心跳的时间间隔。

4) 站点配置

目前应用仅支持一个站点，如果 dna-server.xml 中存在 site 元素，则按照 site 元素的信息配置站点的名称和数据源，否则按照默认名称配置站点名称并且取 datasources 元素下第一个 datasource 元素作为数据源配置。

sites 元素配置应用的各个站点属性。

属性名称	单位	默认值	说明
name	无	default	站点名称。

datasource-refs，配置站点具体空间使用的数据源。

属性名称	单位	默认值	说明
space	无	空	空间的路径，空表示站点。
datasource	无	无	数据源的名称，空表示使用第一个数据源。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

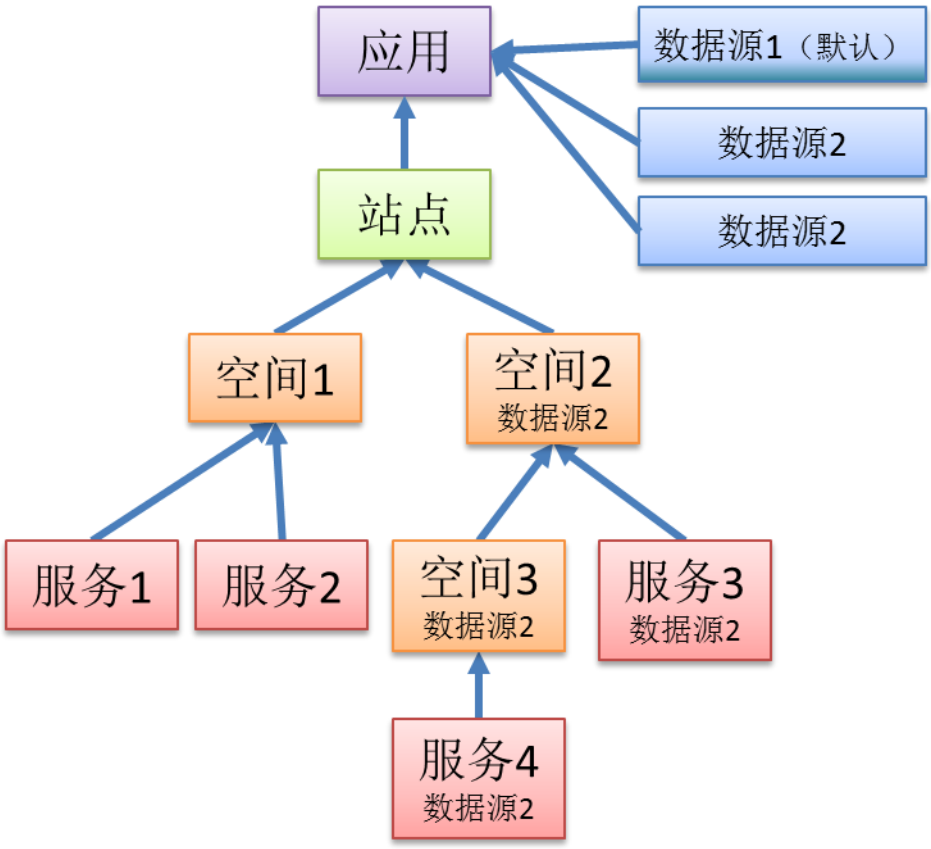


图 应用结构

如图所示，通过应用配置文件向应用注册数据源，其中第一个数据源作为默认数据源。应用的站点及所有空间，将自动使用默认数据源。即意味着，在所有空间下，context 执行的数据库操作自动使用默认数据源。

站点配置中，可以指定空间使用其他的数据源配置。并且，该配置会继续向下级空间传播。

例如上图，在不指定的情况下，各空间都自动使用数据源。当指定**空间 2** 使用**数据源 2** 后，其下所有服务及子空间都会继承该配置。

在 site 元素的 datasource-refs 元素中，使用 datasource-ref 为空间单独指定数据源。space 属性指定空间，datasource 元素指定使用的数据源名称。这样，该空间下的所有的服务的 context，都使用特别指定的数据源。

5) 重启策略配置

reboot 元素用于配置重启功能的策略。目前支持周策略和天策略两种配置，两种策略可以同时使用。集群定时重启功能的配置示例详见本章标题 4(装载应用配置文件)中的示例。

weekstrategy 周策略：

属性名称	单位	默认值	说明
weekday	无	0	设置在周几进行重启。 0 代表周日，1-6 分别代表周一至周六。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

reboot-time	无	00:00:00	设置重启时间。
delay-time-m	分钟	30	当时间到达设定的重启时间，而当前站点的会话数大于设定的 session-floor 的值时，站点将延时 delay-time-m 进行重启。
session-floor	个	10	设定站点会话数的下限，若小于该下限，则立即重启。否则，站点延时重启。

daystrategy 天策略：

属性名称	单位	默认值	说明
cycle	天	3	设置重启的周期，即 cycle 天重启一次。
reboot-time	无	00:00:00	设置重启时间。
delay-time-m	分钟	30	当时间到达设定的重启时间，而当前站点的会话数大于设定的 session-floor 的值时，站点将延时 delay-time-m 进行重启。
session-floor	个	10	设定站点会话数的下限，若小于该下限，则立即重启。否则，站点延时重启。

6) 集群配置

cluster 元素用于配置集群相关信息。如果是单独的应用，非集群环境，则可以将 cluster 段从 dna-server.xml 中删除。

属性名称	单位	默认值	说明
id	天	必填	集群的标识，可以由任意符合 xml 规范的字符组成。
index	无	必填	当前节点在集群中的索引号。要求同一个集群中的每个节点的索引号都是唯一的。

node 元素用于配置集群中节点的地址，它只有一个属性：

属性名称	单位	默认值	说明
url	无	必填	集群节点的 http 地址。即 Web 应用的 http 地址。注意，不是用户访问入口 url 地址。

注意，cluster 的 index 属性和 node 元素的顺序不相关。不要求 node 元素有序。index 属性不表示当前节点在 node 元素列表中的位置。node 元素列表中可以包含当前节点，也可以不包含当前节点。

7) LDAP 服务配置

ldap 元素用于配置 LDAP 服务的基本信息，各属性详细说明如下：

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

属性名称	单位	默认值	说明
ctx-factory	无	com.sun.jndi.ldap.LdapCtxFactory	构造 LDAP 上下文的工厂类。
url	无	必填	完整的 LDAP 服务地址。如：ldap://host:port
domain	无	空	域。如：xxx.com.cn

5. 装载 dna 配置文件

dna.xml 称之为 **dna 配置文件**，存放于每个 Bundle 的根目录下，用于声明声明该 Bundle 提供的收集器，Servlet 和其他发布元素包括：服务，元数据定义等。

一个 dna.xml 的示例如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<dna>
    <servlets>
        <servlet path="/sample" class="com.jiuqi.dna.SampleServlet" />
    </servlets>
    <gathering>
        <gatherer group="samples"
                  element="sample"
                  class="com.jiuqi.dna.SampleGatherer" />
    </gathering>
    <publish>
        <services>
            <service space="dna/sample"
                     class="com.jiuqi.dna.SampleService" />
        </services>
        <tables>
            <table space="dna/sample" class="com.jiuqi.dna.SampleTable" />
        </tables>
        <orms>
            <orm space="dna/sample" class="com.jiuqi.dna.core.SampleORM" />
        </orms>
        <querys>
            <query space="dna/sample"
                   visibility="PROTECTED"
                   class="com.jiuqi.dna.core.SampleQuery" />
        </querys>
        <commands>
            <command space="dna/sample"
                     visibility="PROTECTED"
                     class="com.jiuqi.dna.core.SampleCommand" />
```


公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

```
        </commands>
    </publish>
</dna>
```

应用在启动阶段，会一次读取各 Bundle 的 dna 配置文件内容，但分两部分使用。第一部分，包括 servlest 和 gathering 元素。这部分配置会首先被解析，并在随后的应用启动流程中使用。包括 servlets 元素，和 gathering 收集器元素。收集器之后将作为站点启动的主要功能的完成者，负责进一步的解析 dna.xml 中声明的发布元素，逐步构建站点的空间，实例化服务和相关元数据。

第二部分为 Publish 元素。该部分用于声明各 Bundle 所有发布的元素。在站点启动的过程中，应用会使用已经注册的所有收集器收集所有 Bundle 的 Publish 元素。

Publish 元素包括两层 xml 标签，第一层称之为 group，第二层表示声明发布的元素。例如 services 标签为第一层，service 标签为第二层，用于声明发布的 service。

1) servlets 与 servlet 元素

用于注册 servlet。path 属性指示 servlet 的路径，class 指示 servlet 的具体类，必须是 javax.servlet.Servlet 的子类，init-order 指示初始化顺序。如果 init-order<0，servlet（fliter）在首次访问时初始化，否则在系统启动时按照 init-order 由小到大的顺序初始化。servlet 默认的 init-order 为 Integer.MAX_VALUE。

2) filters 与 filter 元素

用于注册 filter。path 属性指示 filter 的路径，class 指示 filter 的具体类，必须是 javax.servlet.Filter 的子类，init-order 指示初始化顺序（同 servlet）。filter 默认的 init-order 为 Integer.MAX_VALUE。

3) gathering 与 gather 元素

用于注册收集器。group 属性指示收集器收集元素的组名称，element 属性指示收集器收集元素的名称，class 指示收集器的类，必须是 com.jiuqi.dna.core.spi.publish.PublishedElementGather 的子类。

4) publish 元素

用于注册所有发布的元数据，其下元素有两级标签。第一次为组名，第二级为元素名。注册收集器时，即需声明其收集 publish 元素的组和元素名。

以下继续介绍 dna 框架默认支持的注册元素。

5) tables 与 table 元素

用于注册逻辑表。space 指示逻辑表存放的站点空间，visibility 属性指示元数据的可见性，class 指示逻辑表声明器的具体类，必须是 TableDeclarator 的子类。

6) querys 与 query 元素

用于注册查询语句。space 指示查询语句存放的站点空间，visibility 属性指示元数据的可见性，class 指示查询语句声明器的具体类，必须是 QueryStatementDeclarator 的子类。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

7) orms 与 orm 元素

用于注册绑定查询语句。space 指示绑定查询语句存放的站点空间，visibility 属性指示元数据的可见性，class 指示绑定查询语句声明器的具体类，必须是 MappingQueryStatementDeclarator 的子类。

8) commands 与 command 元素

用于注册更新语句。space 指示更新语句存放的站点空间，visibility 属性指示元数据的可见性，class 指示更新语句声明器的具体类，必须是 ModifyStatementDeclarator 的子类。

9) procedures 与 procedure 元素

用于注册存储过程。space 指示存储过程存放的站点空间，visibility 属性指示元数据的可见性，class 指示存储过程声明器的具体类，必须是 StoredProcedureDeclarator 的子类。

6. 数据源支持

详见“DNA 数据库访问层参考手册”。

7. 收集器

收集器的主要作用是解析 dna 配置文件中的发布元素，即 publish 元素，并应用到站点。

框架自带的默认收集器，可以收集 services，tables，commands，orms，querys 等元素。这些收集器负责实例化各元数据，关联引用，同步数据库结构，注册定义到站点等逻辑，这些过程在第三章站点启动中会详细阐述。

此外，框架也允许用户注册自定义的收集器。自定义收集器需要在 dna 配置文件的 gathering 元素中声明。

gather 标签指示注册收集器，属性 group 指定发布元素组的标签名，element 用于指定发布元素的标签名，class 用于指定收集器的实现类，必须继承自 com.jiuqi.dna.core.spi.publish.PublishedElementGatherer，并且提供不带参数的构造方法，或参数为 SXElement 的构造方法。

8. 启动参数

1) 调试参数

- ✓ com.jiuqi.dna.debug
  - ➢ 类型：boolean
  - ➢ 默认值：false
  - ➢ 说明：DNA 调试模式开关，这个“调试模式”不是指 JVM 的调试模式。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

- ✓ com.jiuqi.dna.debug.level
  - ➢ 类型：String
  - ➢ 默认值：all
  - ➢ 说明：由逗号“，”分隔的一组字符，指示使用 DNALogger 接口输出的调试信息的级别。
    默认为“all”表示所有级别的日志都记录。
    可选值为“debug,info,warn,error,fatal”。
    例如：-Dcom.jiuqi.dna.debug.level=debug,error 表示只记录级别为 DEBUG 和 ERROR 的日志，其他级别的日志不会记录。如果再加上-Dcom.jiuqi.dna.debug=true 参数，则记录日志的同时，还会将日志内容打印到标准输出上。

2) 数据访问相关参数

请参考《数据访问层参考手册》。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

第三章　站点启动

1. 流程概述

启动站点的流程如下：

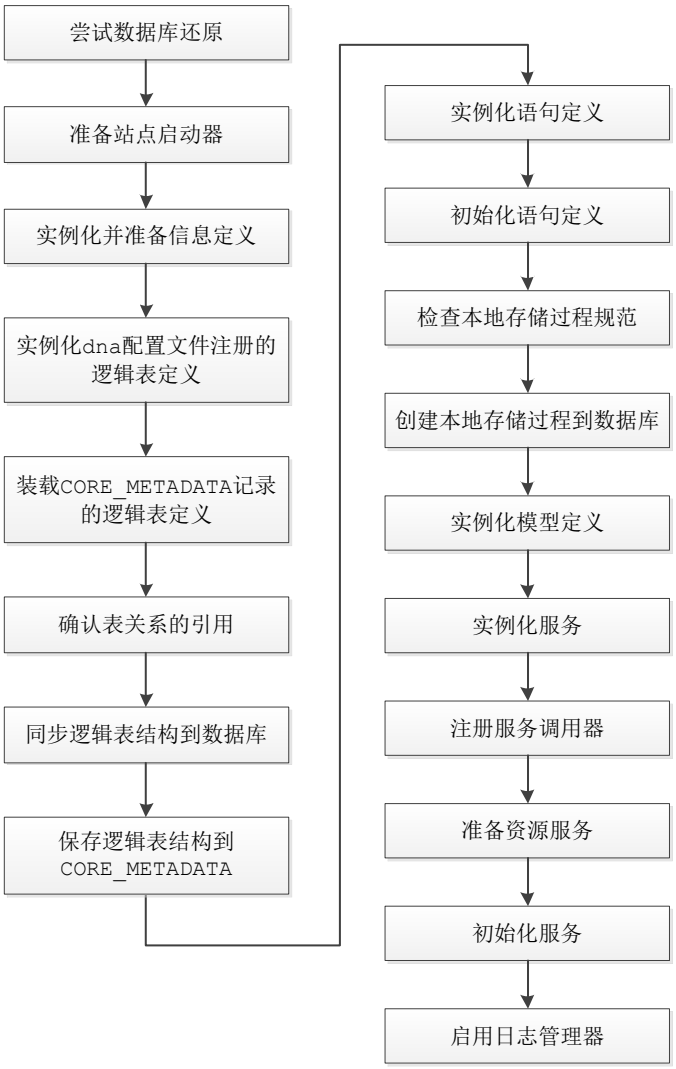


图 站点启动流程

2. 尝试数据库还原

应用会读取当前站点使用的数据源名称，并查找在应用的目录 work 下是否存在相同名称的且扩展名为 ddb 的数据库备份文件，如果存在，应用将还原备份文件中包含的表。还原操作将完全删除并重建备份文件中包含的表，所以务必小心操作。备份操作结束后，应用会移动备份文件到 work 的子目录中，从而下次应用启动不会再次触发数据库还原操作。

有关数据库备份还原功能参考相关文档。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

3. 实例化并准备信息定义

不详细介绍。

4. 实例化 dna 配置文件注册的逻辑表

该步骤中，应用尝试实例化所有 Bundle 的 dna 配置文件中声明的逻辑表定义，并将逻辑表注册到相应的站点空间中。

该步骤不会实例化表关系对象——因为表关系所引用的逻辑表在该步骤可能还未构造。

5. 装载 CORE_METADATA 记录的逻辑表

该步骤中，应用首先会读取数据库 CORE_METADATA 表中所有逻辑表的结构定义，如果指定名称的逻辑表在站点中已经注册，应用会将该表结构**合并**到站点注册的逻辑表中；如果指定名称的逻辑表在站点中不存在，应用会实例化该逻辑表定义，并注册到站点中。

具体的合并规则，将在逻辑表模型一章中介绍。

6. 确认表关系的引用

该步骤中，应用根据逻辑表定义的表关系定义的元数据，建立表关系的引用。

7. 同步逻辑表结构到数据库

该步骤中，应用会将所有在站点中注册的逻辑表结构同步到指定数据源中。具体的同步规则见数据库结构维护一章。

8. 保存逻辑表结构到 CORE_METADATA

装载 CORE_METADATA 过程中合并表定义，及同步表结构，都可能会导致逻辑表定义的属性或结构发生变化，此步骤会将最新的逻辑表结构保存到 CORE_METADATA 表中，以供下次应用启动使用。

9. 实例化语句定义

该步骤中，应用根据配置文件，实例化 Publish 元素中声明的各种语句。

由于语句是依赖于逻辑表的，当语句使用逻辑表在站点中不存在时，将进一步导致语句的实例化失败。

10. 初始化语句定义

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

该步骤中，应用准备实例化的语句对象，使其进入可被执行状态。

11. 检查原生存储过程规范

该步骤中，应用检查 dna 配置文件文件注册的原生存储过程是否符合框架要求的规范。

12. 创建原生存储过程到数据库

该步骤中，应用根据开发者提供的原生存储过程构建脚本，创建存储过程到数据库中。如果指定名称的存储过程已经存在，应用会重新创建存储过程，保证数据源中存储过程是最新的。

13. 实例化模型定义

不详细介绍。

14. 实例化服务

该步骤中，应用根据 dna 配置文件中声明的 service 定义，实现各服务对象。

15. 注册服务调用器

该步骤中，应用根据 service 类中，包含的各种 TaskHandler 及 ResultProvider 等声明，将服务调用器注册到应用的调用列表中，如果出现键值的调用器声明，将会抛出异常。

16. 准备资源服务

准备资源服务相关。

17. 初始化服务

即调用各服务的 init 方法，初始化各服务的状态。所有服务的初始化顺序按照各 service 重写的 getPriority()的值来决定。

18. 启用日志管理器

不详细介绍。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

第四章　框架特性

1. 元数据及使用

1) 元数据概念

元数据，Metadata，指用于描述数据的数据，主要描述数据的属性。

例如，数据库表可以存储数据，而用于描述表结构本身的数据，便可称之为元数据。

2) 框架定义的元数据

- ✓ 信息定义：信息定义用于向系统报告异常，日志及消息。
- ✓ 逻辑表定义：逻辑表是 DNA 框架定义及使用的抽象逻辑表结构，数据库访问层的基础。
- ✓ 语句定义：面向逻辑表的有种 SQL 语法的封装。在框架内，必须使用语句定义来访问数据库。
- ✓ 模型定义：DNA 框架提供的动态 java 实体模型。

定义各种框架定义的元数据，都使用名称标识，名称区分大小写，并且名称在同一空间下唯一。

3) 定义接口与声明接口

DNA 框架中，core 包大量使用面向接口的编程风格，所有的元数据定义都提供**两个方面**的接口，即定义接口 Define，和声明接口 Declare。例如逻辑表定义有 TableDefine，和 TableDeclare 两个接口。

Define 与 Declare 同为元数据定义的不同侧面。Define 为元数据的只读接口，即其所有方法不会修改对象的属性与结构，并且其 get 方法仍然返回 Define 接口；Declare 为可写接口，其方法可能修改对象属性或结构，并且其 get 方法返回 Declare 接口。

虽然在程序上，可以使用 java 的强制类型转换方法来改变 Define 接口到 Declare 接口，但**这种操作是严格禁止的**。因为当框架提供出 Define 接口的对象时，即隐含着该对象的属性和结构绝对不会发生变化，一旦使用强制转换修改对象的属性，即会很容易引发类似线程安全的错误，从而导致元数据的状态不一致。

4) 声明器

除了 Define 与 Declare 接口，框架还提供了各种声明器，即 Declarator。和 Define 及 Declare 接口不同的是——声明器不是对象本身的某个方面，而是一个用于构造某具体对象的容器，也是该具体对象构造逻辑的封装，并且还可以提供对象各元素的强引用声明。这部分在逻辑表声明器的相关部分有更为详细的展示。声明器的 getDefine()方法可以返回其定义对象的 Define 接口。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

需要特别注意一点：所有 Declarator 对象都是单例的，其实例化过程只能由框架本身完成。使用者调用任何 Declarator 的构造方法，都会抛出异常。使用者只能通过 context 的 get 方法，获取指定类型的 Declarator 的实例。例如：

```
TableA tbA = context.get(TableA.class);
```

相关更详细的使用范例参考调用器一节。

框架目前提供的涉及到涉及元数据的接口，大部分都过载了 Declarator 和 Define 类型的参数，本质上是一致的。

声明器必须在 dna 配置文件中注册。

5) 可见性

元数据在注册时有可见性属性，框架通过可见性控制业务访问的数据和功能调用的范围，实现业务层的封装。可见性是可配置的，在 dna 配置文件中元数据的注册信息上指明了可见性，属性名为“visibility”。

可见性属性的有效值包括：

名称	xml 属性值	java 枚举值	说明
DEFAULT	默认值	Publish.Mode.DEFAULT	默认，与父元素访问性相同
SITE_PUBLIC	site_public	Publish.Mode.SITE_PUBLIC	对所有站点的所有模块可见，远程调用亦可见
~~SITE_PROTECTED~~	~~site_protected~~	~~Publish.Mode.SITE_PROTECTED~~	~~对本站点以及子站点的所有模块可见~~
PUBLIC	public	Publish.Mode.PUBLIC	对本站点的所有模块可见
~~PROTECTED~~	~~protected~~	~~Publish.Mode.PROTECTED~~	~~对本空间下模块以及子模块可见~~

由于目前的版本中没有实现子站点的设计，也没有启用 PROCTECTED 可见性检查，所以 SITE_PROTECTED 和 PROTECTED 可见性是无效的。所有 PROTECTED 的元数据均按照 PUBLIC 来进行访问。

鉴于以上原因，可见性的主要应用是在远程调用方面。为远程调用服务的元数据（或者处理器）需要声明为 SITE_PUBLIC，否则不能访问。

除了元数据，框架还对处理器进行了可见性控制。处理器的可见性不能通过 dna 配置文件进行配置，而是用 Publish 标注（annotation）来指定的。

2. 调用器

1) 调用器概念

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

Context（上下文）接口中提供了各种 get 和 handle 方法：前者表示请求某种类型的数据，并强调该处理过程不会修改应用的状态；后者表示处理某种类型的数据，该处理过程可能会修改应用的状态。

框架定义用于返回数据的逻辑单元为“提供器（Provider）”，处理数据的逻辑单元为“处理器（Handler）”。两者本质上是一致的，即可以统称为“调用器”——根据某种类型的数据，调用一段逻辑代码单元。区分出提供器和处理器两种分类，是为了更好的规范接口的含义并使用接口使用更为方便。

2) 方法、方法依赖、方法行为

在一般编程上，通过方法来封装一段逻辑处理块，所谓的调用，通常都是指对方法的调用。

一个方法的签名，可以认为包括了两个要素：方法名称；参数类型。

针对方法的调用，即依赖了以上的两个要素。其中，对方法名的依赖，可以进一步的描述为：对称方法的提供者的依赖，以至于对方法提供者的构造的依赖。相比而言，对参数类型的依赖就显得更为轻量级。

在声明一个方法时，也可以通过以上两个要素来描述方法的行为，提供使用者方便理解方法。那么，**如何分摊方法名和参数类型对方法行为影响力**就成了一种编程的风格。换句话说：到底是方法名决定行为，还是参数决定行为，抑或是两者共同决定行为。

例如，某类有两个典型方法：

```
void methodA(boolean b);
void methodB();
```

完全以方法名来决定行为，方法声明可以改写为如下：

```
void methodA0();
void methodA1();
void methodB();
```

完全以参数来决定行为，方法声明可以改写为如下：

```
void method(int methodId);
```

以上三种风格的代码，都能实现相同的逻辑，但给使用者以完全不同的体验。一般需要根据具体的场景来选择相应的风格。

3) 参数决定行为

鉴于以上论述，对方法名的依赖会导致对方法提供类的依赖，进一步的导致对方法提供类的构造的依赖。在大型系统中，这种依赖关系就显得难于维护。因此，DNA 框架在封装各个逻辑处理单元时，便采用了完全的参数决定行为的风格，以此降低各模块调用的依赖，更便于系统的维护和管理。可以看到，在框架内部，各逻辑单元之间的相互调用，基本上可以简化为两类方法：get 和 handle，即提供接口和处理接口，或更统一的称之为调用接口。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

3. Web 容器

框架使用 Jetty 7 作为 web 容器。借助界面框架，开发人员可以方便的实现大部分前台业务逻辑，而不需要直接与 web 容器打交道。但是框架仍然支持 filter 和 servlet 应用。

通过 dna 配置文件注册 servlet/filter：

```
<?xml version="1.0" encoding="UTF-8"?>
<dna>
   <servlets>
      <servlet path="/dna_core/obf"
               class="com.jiuqi.dna.core.impl.ObfuscateServlet"/>
      <servlet path="/*" init-order="0"
         class="org.eclipse.equinox.http.servlet.HttpServiceServlet"/>
   </servlets>
</dna>
```

框架通过收集 dna 配置文件的 servlets/servlet 元素来注册 servlet 类。

开发人员在注册 servlet 时应当注意避免 url 模式冲突。逻辑框架已经占用/dna_core/和/dna_ws/路径，开发人员应当避免注册这两个路径。逻辑框架提供了一些实用工具，地址如下：

- ✓ /dna_core/obf 文明密码混淆工具
- ✓ /dna_core/unobf 密文密码反混淆工具
- ✓ /dna_core/db 数据库备份工具

4. 调用框架

1) 调用框架的目标

调用框架的首要目的是解除调用者和被调用者之间的耦合关系。

其次，调用框架封装了调用过程，将本地调用、远程调用、异步调用等多种调用方式以统一的接口呈现给上层开发人员，便于使用。

另一方面，调用框架通过 SPI 进行扩展的方式为功能模块的结构提供了合理的划分，适合作为编码规范。

2) 基本思想

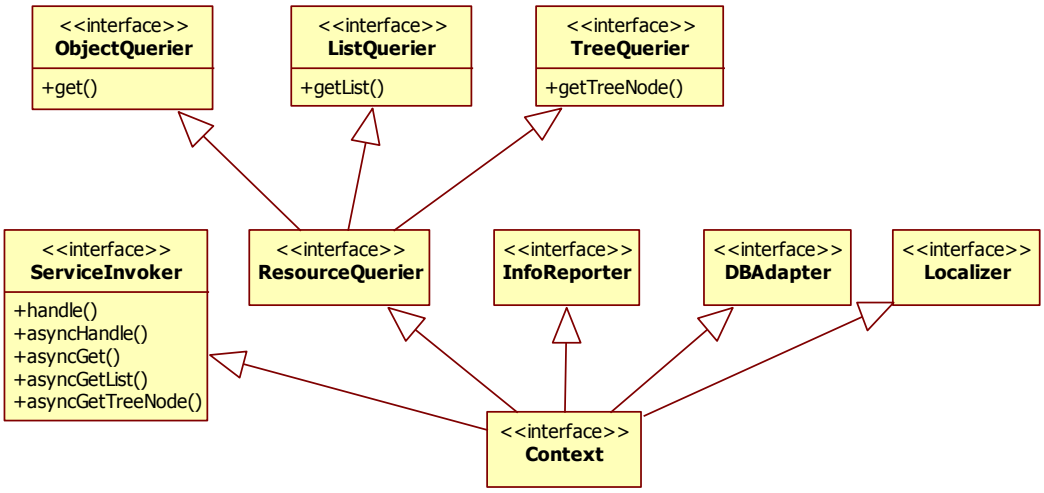
解耦的基本思路是使用数据对象作为功能模块间的接口进行交互，避免模块间的强引用。

统一接口的思路是将常用的调用模式抽象为几种固化的接口（任务处理器、提供器、事件监听器等），而将参数类型弱化。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

框架将被调用者封装为处理器。在系统启动阶段处理器就被注册到框架中，当执行调用时，框架根据调用者传入的参数（数据对象）类型信息查找到已注册的处理器，然后调用处理器的方法来实现参数的处理过程，最后将处理过的参数返回给调用者。

3) 执行调用



框架将各种模式的调用抽象成有限的几个方法，集中为 Context 接口。调用可能由框架来处理，也可能是框架外部模块注册的处理器来处理的。Context 接口提供的调用方法主要包括：

✓ get

用于获取数据对象。数据对象可以是框架内部对象，如表定义，也可以由结果提供器(ResultProvider)产生。框架支持的对象包括：

- 类型(java.lang.Class)
- 表定义(TableDefine)
- 新增语句定义(InsertStatementDefine)
- 删除语句定义(DeleteStatementDefine)
- 更新语句定义(UpdateStatementDefine)
- 查询语句定义(QueryStatementDefine)
- ORM 定义(MappingQueryStatementDefine)
- 函数定义(UserFunctionDefine)
- 存储过程定义(StoredProcedureDefine)
- 信息报告组定义(InfoGroupDefine)

✓ getList

用于获取一组数据对象，支持从缓存中获取或者从提供器获取。开发人员可以注册相应的结果列表提供器(ResultListProvider)来处理此类调用。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

✓ getTreeNode

用于获取树节点，支持从缓存中获取或者从提供器获取。开发人员可以注册相应的树提供器(TreeNodeProvider)来处理此类调用。

✓ handle

用于执行任务。该接口将任务(Task)作为参数调用相应的任务处理器(TaskMehtodHandler)。

✓ dispatch

触发事件。开发人员可以注册事件监听器(EventListener)来处理此类调用。

我们将以上几个方法相归为同步调用，与之相对，Context 接口还提供了一组异步调用的方法，例如 asyncGet、asyncHandle、occur 等。使用这些方法执行调用时，调用过程不是在当前线程中执行的而是在其他线程中执行的，使用的处理器和同步调用的处理器完全相同。

4) 提供器的键

我们将数据对象划分为无键、单键、双键和三键的，这里的键指数据的主键，通常存储于数据对象的属性中。相对的，提供器也分为无键、单键、双键和三键的。框架为不同键的提供器提供了不同的基类，如单键提供器的基类为 OneKeyResultProvider，双键的为 TwoKeyResultProvider。无键提供器通常只提供单例的对象，其基类为 ResultProvider。根据键的数目不同，调用提供器的方法也不相同。get 方法有多个签名，分别对应不同键的提供器，例如 get(facade)调用 ResultProvider，get(façade, key1, key2)调用 TwoKeyResultProvider，超过三个键的方法如 get(façade, key1, key2, key3, others)都没有实现，应避免使用。

上面介绍的几个提供器只能提供单个数据对象，框架还支持提供列表和树型结构的提供器。列表提供器的基类为 ResultListProvider、OneKeyResultListProvider 等，调用列表提供器的方法为 getList。树提供器的基类为 TreeNodeProvider、OneKeyTreeNodeProvider 等，调用树提供器的方法为 getTreeNode。

除了 get 方法，框架还提供了 find 方法来获取单个数据对象。get 与 find 的区别为，在找不到数据对象时 get 方法会抛出异常，find 方法不抛出异常而是返回 null。

5) 编写服务

get、handle、dispatch 等调用必须有相应的处理器来处理，否则执行调用时框架会抛出异常。框架提供了 Service 基类让开发人员进行扩展。下面我们以单键提供器为例说明如何扩展功能模块。

首先，我们创建一个 Service 的子类，编写处理器的实现。

```
class SampleService extends Service {
    public SampleService() {
        super("演示服务");
    }

    class SampleProvider extends OneKeyResultProvider<Sample, String> {
        @Override
```


公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

```
        protected Sample provide(Context context, String key)
                                    throws Throwable {
            // 执行查询并返回Sample对象
            String value = (String) context
                    .executeScalar(
                            context.parseStatement(
                                    "define query(@key string) begin select
s.value as value from Sample as s where s.key = @key end",
                                    QueryStatementDeclare.class), key);
            if (value != null) {
                return new SampleImpl(key, value);
            }
            return null;
        }
    }
}
```

然后，在 dna.xml 中注册这个服务：

```
<?xml version="1.0" encoding="UTF-8"?>
<dna>
    <publish>
        <services>
            <service space="sample" class="sample.SampleService" />
        </services>
    </publish>
</dna>
```

最后，用 ObjectQuerier 接口来访问提供器：

```
Sample s = context.get(Sample.class, "abc");
```

任务处理器和事件监听器也采用同样的方法进行扩展。

5. 缓存框架

请参考《缓存开发参考手册》。

6. 数据访问

请参考《数据访问层参考手册》。

7. 事务管理

框架提供自动化的事务管理机制。通常情况下，事务和线程是绑定在一起的，即一个线程一个事务。开发人员可以通过异步调用发起新事务。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

开发人员需要小心的处理异常情况，因为功能模块抛出的任何异常都会导致事务回滚。当且仅当整个处理过程都没有抛出为处理的异常时，框架才会提交事务。

还应当注意的是，并非框架提供的所有操作都是支持事务的。例如 DDL 操作在 SQLServer 上是支持事务的，在 Oracle 上不支持事务。所以开发时应当考虑到操作失败对数据的影响，并进行相应的处理。

框架对事务的管理不仅局限于数据库事务，框架还对缓存进行内存事务管理，并且总是保证数据库事务和内存事务一同提交或一同回滚。关于内存事务请参考《缓存开发参考手册》。

8. 会话管理

会话用于表示一系列操作的生命周期。框架提供多种不同类型的会话以适应不同的应用场景。

1) 系统会话

在系统启动时，框架使用系统会话进行系统的初始化，启动完成后，系统会话就销毁了。

2) 标准会话

标准会话是由用户发起的会话，通常是由界面框架创建的。标准会话表示了一个用户从访问站点开始直到离开站点的整个生命周期。标准会话主要用于保存用户交互过程中的登录信息、权限、界面控件等状态。

一个典型的会话管理流程如：

- ✓ 客户端访问站点：系统创建一个标准会话，用匿名用户 BuildInUser.anonym 作为当前用户。系统将会话的标识发送给客户端，客户端的后续操作都会带着该标识。
- ✓ 用户登录：系统为其分配一个新会话，并记录用户名、密码、权限等用户相关信息，将旧会话销毁。
- ✓ 用户发出请求（如点击按钮）：系统根据会话标识找到先前分配的会话，然后从会话中获取界面状态等信息，以此为基础处理用户请求。处理过程中可能对会话状态产生影响。
- ✓ 用户注销：系统将会话销毁。当用户再次访问系统时，又由第一步开始重复会话管理流程。

用户操作可能不按照我们预期的顺序进行，例如没有注销或者直接关闭浏览器等。为了在这些情况下能正确的处理会话状态，我们为标准会话设计了超时特性。通过 dna-server.xml 可以配置会话超时时间和心跳时间。

超时时间指当用户不进行任何操作并超过一定时间的情况下，系统会认为用户不再使用系统，并且将会话销毁。

心跳时间用于控制系统和客户端之间的心跳时间间隔。无论用户是否进行操作，客户端总是会定期向服务器端发送心跳信号以通知系统客户端还在线。当该心跳信号中断时，即连续两次心跳时间间隔大于设定值，系统会认为客户端已经离线，所以将会话销毁。

利用超时的特性，在用户不注销或者意外离线的情况下，系统也能够对会话进行管理。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

3) 临时会话

临时会话是由功能模块发起的异步调用产生的会话。默认情况下，异步调用是在当前会话中执行的，但是开发人员也可以指定在新的会话中执行异步调用。对于后一种情况，系统会为异步调用分配临时会话。异步调用完成，会话就被销毁。

4) 远程调用会话

远程调用会话是用于处理远程调用请求的会话。该会话是由系统分配、销毁的。开发人员不能发起远程调用会话。但是发起远程调用总会导致接收远程调用的系统分配远程调用会话。

9. 权限控制

请参考《缓存开发参考手册》。

10. 信息报告

信息报告机制主要有以下几个方面的应用：

1) 反馈任务执行的进度

2) 记录日志

3) 本地化

11. 性能监控

12. 收集器

收集器用于收集 dna 配置文件中的注册信息。逻辑框架使用收集器来收集服务、表定义、DML 语句等注册信息。开发人员也可以利用逻辑框架提供的接口来扩展自定义的收集器。

首先，自定义收集器继承自 com.jiuqi.dna.core.spi.publish.PublishedElementGatherer。下面的代码演示了一个最简单的收集器的结构。

```
public class SampleElement extends PublishedElement {
    public SampleElement(SXElement element) {
        // 读取xml属性
    }
}

public class SampleGatherer
                extends PublishedElementGatherer<SampleElement> {
    @Override
    protected SampleElement parseElement(SXElement element,
```


公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

```
                                        BundleToken bundle)
                                             throws Throwable {
        return new SampleElement(element);
    }
}
```

然后，在 dna.xml 中注册收集器。

```
<?xml version="1.0" encoding="UTF-8"?>
<dna>
    <gathering>
        <gatherer group="samples"
                  element="sample"
                  class="sample.SampleGatherer"/>
    </gathering>
    <!-- 使用sample收集器
    <samples>
        <sample/>
    </samples>
    -->
</dna>
```

系统启动时会自动从 dna.xml 中收集<dna>/<samples>/<sample>元素，并逐一调用 SampleGatherer.parseElement 方法进行处理。

13. WebService

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

附录

1. 如何配置 dna-server.xml

1) 重点参数

首先给出重点参数：

```
<?xml version="1.0" encoding="UTF-8"?>
<dna>
    <datasources>
        <datasource max-connections="35" min-connections="35"/>
    </datasources>
    <http>
        <listen max-threads="25" accept-queue-size="1000"/>
    </http>
    <session heartbeat-s="300"/>
</dna>
```

关键配置项已经用蓝色字体标出，下面逐一进行说明：

2) max-connections 和 min-connections

dna 框架使用了池方式对数据库连接进行缓存和复用。以上两个参数用于设置该池的初始连接数和最大连接数。由于数据库管理系统通常有最大连接数/会话数限制，在配置 max-connections 参数时要考虑到数据库的限制，并且要考虑到在集群中所有节点的连接数总和才是实际的数据库连接数，所以集群所有节点的 max-connections 参数总和应当小于数据库最大连接数限制。另外，oracle 的最大会话数和最大连接数是相等的，一条 sql 语句在执行时只需要一个数据库连接，但是需要一个到多个数据库会话，所以还要考虑会话数的限制。oracle 最大会话数的默认值为“1.1*最大连接数+5”。

对于性能测试，建议将 min-connections 和 max-connections 设为相同值，这样系统一启动就尝试分配数据库连接，如果超出数据库最大连接数限制程序就会抛出异常。

3) max-threads 和 accept-queue-size

max-threads 参数决定了 dna 应用程序允许并发处理的 http 请求的上限。accept-queue-size 参数决定了待处理的 http 请求的最大数量。

这两个参数仅对一个 listener（即监听）有效，每个 listener 都有自己的 max-threads 和 accept-queue-size。

以上配置文件样例配置了两个监听，第一个允许同时处理 25 个 http 请求，并且只接受 1000 个排队中的请求。服务器会拒绝处理队列排满后再接收到的 http 请求，这种情况下客户端会得到

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

http 错误（如 503）。实际业务中一个操作可能需要多个数据库连接，所以建议设置 max-threads 参数比 max-connections 参数值稍小，如 40/50。对于集群的部署方式，我们通常要配置两个 http 监听，一个用于处理用户请求，另一个专用于集群通信。集群在稳定情况下，每个节点与其他节点保持两个连接，所以对于 n 个节点的集群，我们通常要将监听的 max-threads 设置大于 (n-1)*2（应当预留若干连接），并且 accept-queue-size 较小（如 10/20）。

4) heartbeat-s

dna 的客户端总是向服务器发出心跳信号以表示客户端在线。如果两次心跳间隔大于 heartbeat-s 参数，在后一次心跳收到之前服务器会认为该客户端掉线而释放掉相关的会话。

使用测试工具（如 loadrunner）模拟用户操作时应当注意将 heartbeat-s 适当调大。通常测试工具不能模拟 dna 客户端的心跳信号，所以较小的 heartbeat-s 参数更容易引起超时错误。

2. 如何配置 SSL 端口

1) 生成 keystore 文件

使用 JDK 提供的 keytool 生 keystore 文件，例如：

```
D:\dna>keytool -genkey -alias dna -keyalg RSA -keystore key.jps
输入keystore密码:  abcdefg
您的名字与姓氏是什么?
  [Unknown]:
您的组织单位名称是什么?
  [Unknown]:  jq
您的组织名称是什么?
  [Unknown]:  jq
您所在的城市或区域名称是什么?
  [Unknown]:  beijing
您所在的州或省份名称是什么?
  [Unknown]:  beijing
该单位的两字母国家代码是什么
  [Unknown]:  cn
CN=Unknown, OU=jq, O=jq, L=beijing, ST=beijing, C=cn 正确吗?
  [否]:  y

输入<dna>的主密码
        (如果和 keystore 密码相同, 按回车):
```

参数说明：

- ✓ -genkey 表示生成 keystore 文件
- ✓ -alias 表示这个 keystore 的别名，别名要求是独一无二的。
- ✓ -keyalg 表示加密的算法
- ✓ -keystore 表示 keystore 文件的名称

2) 安装 keystore 文件

将 keystore 文件复制到 dna 的 work 目录下即可。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

3) 修改 dna-server.xml

在 dna/http 段里添加一个 ssl 的监听端口，例如：

```
<listen port="9903"
        ssl="true"
        ssl-keystore="key.jps"
        ssl-key-password="abcdefg"
        ssl-password="abcdefg"
        ssl-keystore-type="JKS"/>
```

其中，

- ✓ ssl="true"表示这个端口启用 ssl 协议。
- ✓ ssl-keystore="key.jps"表示 keystore 文件的名称是 key.jps，这个文件必须放在 work 目录下。
- ✓ ssl-key-password 和 ssl-password 是使用 keytool 工具生成 keystore 文件时输入的密码。
- ✓ ssl-keystore-type 是指 keystore 文件的类型，通常是 JKS。更多类型请参考 keytool 相关文档。

3. 如何根据 RECID 计算数据年龄

DNA 应用程序生成的数据表都包含名为 RECID 的 GUID 型字段。该字段是默认的主键字段。我们通常调用 DBAdapter.newRECID()方法来生成 RECID。RECID 是一个 128bit 的数据，它的格式如下：

位置(bit)	长度(bit)	说明
0~63	64	随机序列
64~68	5	簇：0 表示该 RECID 由数据库生成，1~31 表示生成该 RECID 的集群节点索引号。
69~92	24	递增序列：确保 32 毫秒内（相同时间序列）可以产生最多 16M 不冲突的 RECID。
93~127	35	时间序列：由 64bit 时间戳的 5~39bit 组成，由于舍去了低 5 和高 24 位，所以该时间序列以 32 毫秒为单位，每 2^{40} 毫秒≈35 年循环一次。

我们可以截取 RECID 的高 35 位作为数据录入的时间。如果要比较两条数据的年龄，则比较 RECID 的高 59 位即可。计算时需要考虑 35 年循环一次的情况。逻辑框架提供了由 RECID 提取时间的方法。

示例代码：

```
GUID recid;
// 获取时间
long time = TimeRelatedSequence.helper.timeOf(recid.getMostSigBits());
```

4. 文件命名规范

由于受到 OSGI 框架(Equinox)的限制，DNA 不能从包含特殊字符的路径启动，所以开发和部署 DNA 应用程序时应当注意文件和目录的命名。DNA 应用程序仅支持从本地路径启动，不支持从 SMB、

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

HTTP 等远程路径启动。本地路径应当由**英文字母**、**数字**和**下划线**组成，并且所有字符都是**半角**字符。

不符合以上要求的路径命名可能导致 DNA 应用程序无法启动，或者启动过程中发生异常，系统不能提供服务。

目前已知可能导致该问题的字符包括：!#%

不排除其他字符也可能导致该问题的可能性。

5. 如何获取 Java 进程的 PID

Process Identifier 简称 PID 是操作系统为应用程序进程分配的标识符，表示为一个整数。不同操作系统下获取 PID 的方法有所不同：

✓ 使用 Sun 的 JDK 工具获取 PID：

- 进入 JDK 的 bin 目录。
- 在命令行上执行（Windows）：

```
jps
```

- 或者（Linux）：

```
./jps
```

- 结果如下：

```
C:\Program Files\Java\jdk1.7.0_04\bin>jps
2056 org.eclipse.equinox.launcher_1.2.0.v20110502.jar
4056 Jps
8536 Main
```

✓ 在 Windows 下使用任务管理器获取进程的 PID：

- 登录到 Windows 桌面。
- 打开任务管理器，选择菜单“View->Select Columns”（中文版是“查看->选择列...”）
- 勾选 PID 列。点击 OK 按钮。
- 返回任务管理器主界面，可以看到 PID 列已经显示出来。

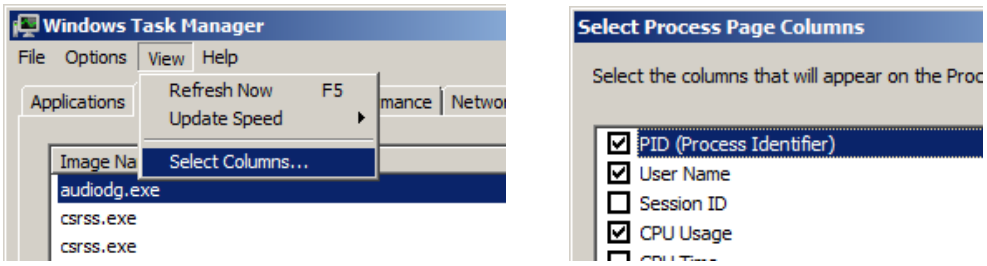


图 开启任务管理器的 PID 列

✓ 在 Windows 下用命令行获取进程 PID：

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

- 登录到 Windows 桌面并打开 cmd 窗口，或者用 telnet 连接到远程 Windows 主机。
- 执行命令：

```
tasklist /fi "imagename eq java.exe"
```

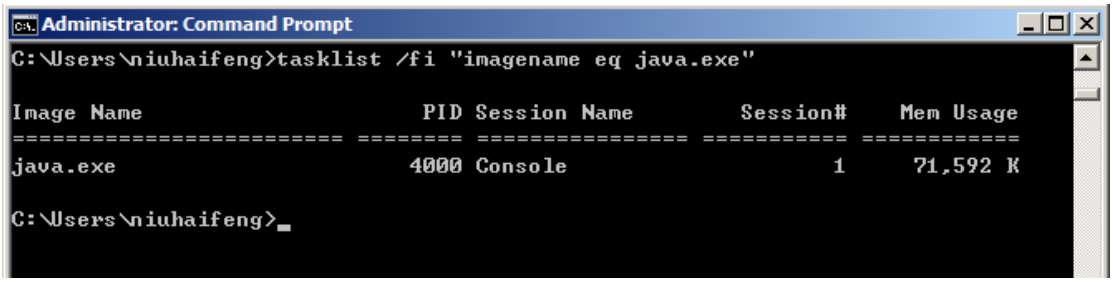


图 在 Windows 下用命令行获取进程 PID

✓ 在 Linux 下获取进程 PID：

- 登录到 Linux 桌面并打开 terminal，或者用 telnet/ssh 登录到远程 Linux 主机。
- 执行命令：

```
ps -ef|grep java
```

- 启动命令中带有 Java 字样的进程将会被打印出来。第二列就是 PID。

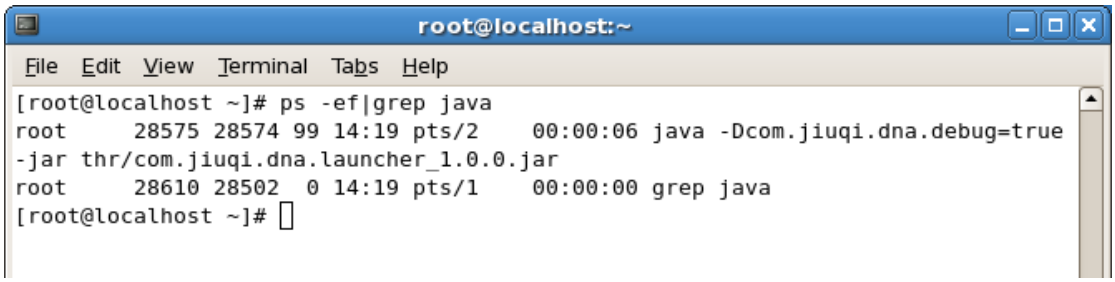


图 在 Linux 下获取进程 PID

6. 如何杀死 Java 进程

✓ 在 Linux 下，用命令行命令：

```
kill -9 <pid>
```

- <pid>指进程的 PID

✓ 在 Windows 下，用任务管理器，或者命令行命令：

```
taskkill /pid <pid> /f
```

- <pid>指进程的 PID
- /f 是可选参数，表示强制结束进程。

7. 如何使用 dump 文件

1) dump 的类别

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

JVM 的 dump 文件对分析程序运行情况有重要帮助。dump 文件包含程序在某个时间点的运行情况，分为两种类型：

- ✓ heap dump：包含程序内存中的全部 Java 对象的镜像。
- ✓ thread dump：包含程序全部线程信息。

2) 使用 JDK 工具生成 dump

用 jdk 自带的工具生产 dump 文件，有几种方式：

- ✓ 用 visualvm 工具：visualvm 是性能分析工具，支持依附到本地进程或者远程进程。
  - ➢ 启动 jdk 的 bin 目录下的 jvisualvm.exe。
  - ➢ 在左侧进程树中选取 dna 进程，点击鼠标右键。
  - ➢ 选择上下文菜单“堆 dump”生成 heap dump。
  - ➢ 选择上下文菜单“线程 dump”生成 thread dump。

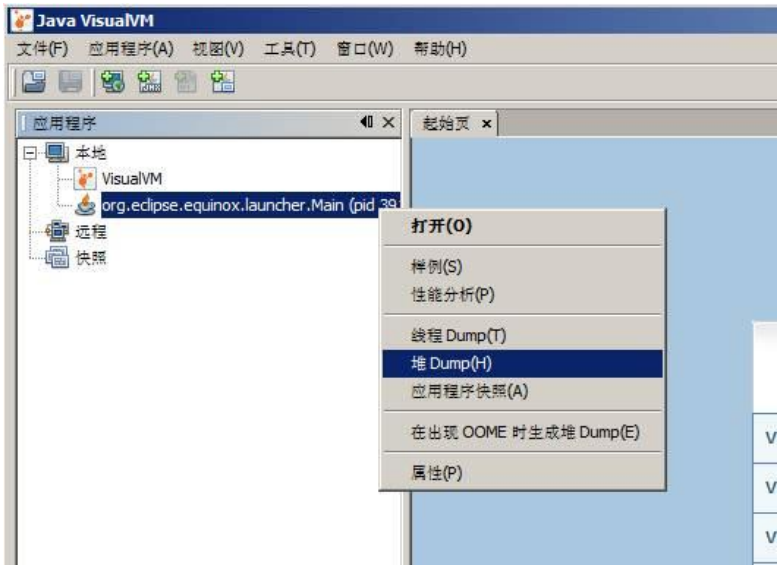


图 用 VisualVM 获取 dump

- ✓ 用 jmap 制作 heap dump：jmap 是命令行工具，只能生成 heap dump。不同版本和平台的 jmap 参数格式可能不同。
  - ➢ windows 平台 sun jdk 1.6 的 jmap 命令格式如下：

```
jmap -dump:file=<dump file> <pid>
```

  - ➢ <dump file>指生成的 heap dump 文件的路径。
  - ➢ <pid>指 Java 程序的 PID。
- ✓ 用 jstack 制作 thread dump：jstack 是命令行工具，格式如下：

```
jstack -l <pid
```

  - ➢ <pid>指 Java 程序的 PID。
- ✓ 在程序发生 OutOfMemoryException 时自动产生 heap dump 的方法：

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

- 在 Java 程序上增加启动参数

```
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=<path to dump file>
```

- <path to dump file>指 heap dump 文件的路径。

✓ 在 linux 环境中获取 thread dump 的方法：

- 在命令行上执行：

```
kill -3 <pid>
```

- <pid>指 Java 程序的 PID。
- thread dump 会输出到 Java 进程的标准输出上，即如果 Java 程序启动时没有重定向就输出到 terminal 上，否则输出到重定向文件中。

3) 使用第三方工具生成 dump

除了 jdk 自带的工具，一些第三方工具也可以生成 dump 文件：

✓ MAT (Memory Analyzer)：Eclipse 推出的 heap dump 分析工具，开源软件，推荐使用。该工具比 visualvm 多查询对象的速度更快，并且对大尺寸的 dump 文件支持较好。

- 官方网站：http://www.eclipse.org/mat/
- 该工具不支持获取远程进程的 dump。
- 选择菜单“File->Acquire Heap Dump…”

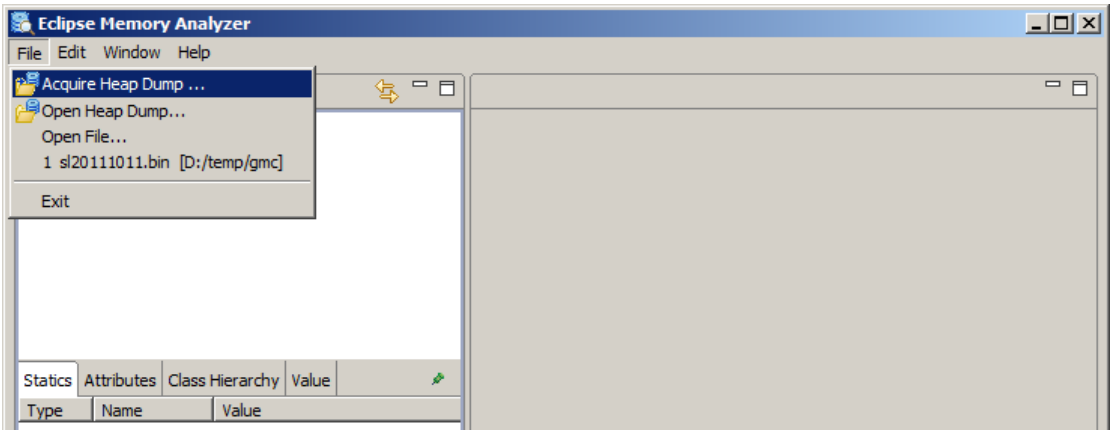


图 用 MAT 获取 dump

- 选择一个进程，选择 dump 文件路径，点击 Finish 按钮。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

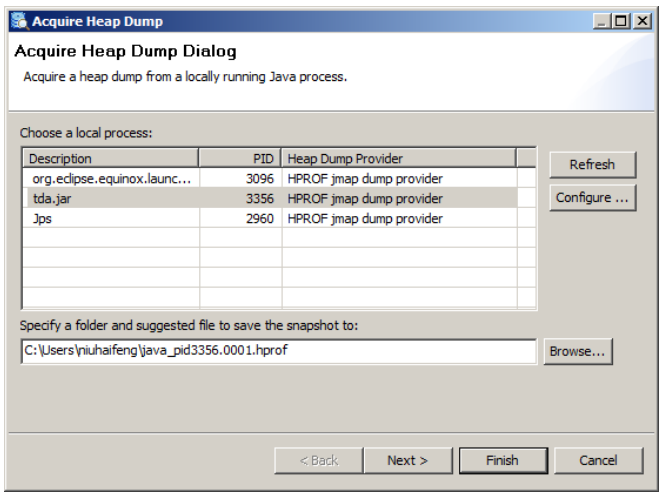


图 MAT 选择进程对话框

✓ Jprofiler：Java 性能分析工具，商业软件，支持 heap dump 文件生成和查看。

- 选择菜单“Session -> New Session”。
- 在“Session Settings”对话框里选择“Session Type”为“Attach”，这样可以依附到现有的进程。

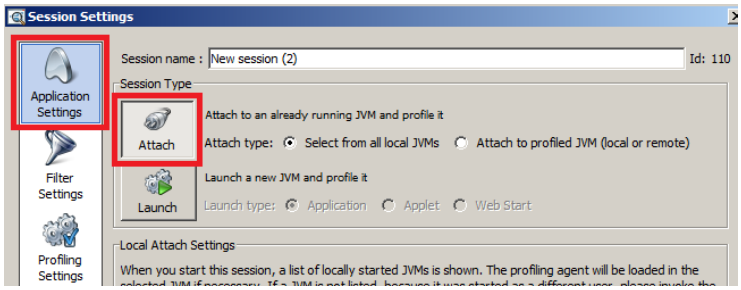


图 Jprofiler “New Session”对话框设置

- 在“select a local jvm”对话框中选择目标进程，并点击“ok”按钮

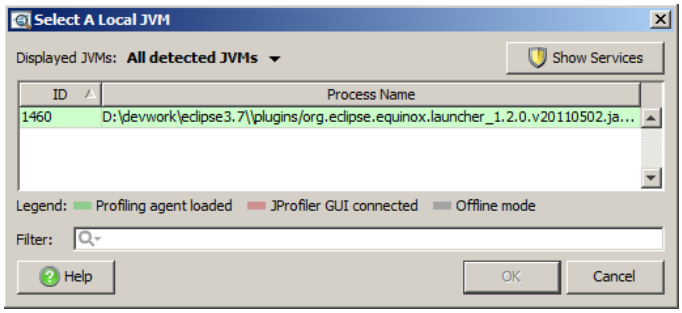


图 Jprofiler 选择进程

- 点击“Take Snapshot”按钮生成 heap dump。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

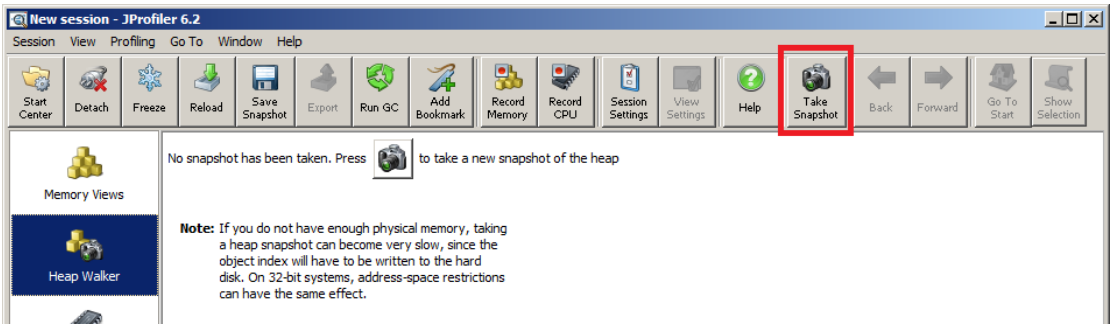


图 用 Jprofiler 获取 dump

4) 分析 thread dump

thread dump 文件是文本文件，可以用任何文本编辑器打开查看。

5) 分析 heap dump

heap dump 文件可以用上面介绍的 visualvm、MAT 和 Jprofiler 工具打开查看。

6) 注意事项

虽然很多 heap dump 工具支持远程获取 dump 文件，但是会受到 Java 安全机制的限制，配置起来非常繁琐，建议尽量将工具安装到目标计算机上再执行。

heap dump 文件的大小通常和程序的内存使用情况有关。使用的内存越多，生成的 heap dump 文件越大。**所以在生成 heap dump 前最好执行以下垃圾回收，以尽量减少 dump 文件的大小。**在 OSGI console 上执行 gc 命令即可执行垃圾回收，上面提到的几个图形化工具也支持垃圾回收的功能。

请注意，前面介绍的方法仅针对 sun 的 Java 虚拟机有效，没有测试过其他的虚拟机实现。

7) 关于 IBM Java Dump 的说明

IBM Java 虚拟机的 dump 与 sun 的不同。在此不作介绍，请参考 IBM 官方网站。

MAT 配合 IBM 提供的 Diagnostic Tool Framework for Java（简称 DTFJ）插件，可以实现为 IBM JDK 生成 dump，并且支持查看 IBM 的 dump 文件。

以下方法没有测试过，仅供参考：

- ✓ 首先确认 IBM JDK 版本，生成 dump 对 JDK 版本要求为：IBM JDK 1.4.2 SR12, 5.0 SR8a and 6.0 SR2。
- ✓ 下载并安装 MAT（版本号 >= 0.8）。
- ✓ 下载并安装 DTFJ 插件，请参考：http://www.ibm.com/developerworks/java/jdk/tools/dtfj.html
- ✓ 安装完成后启动 MAT。选择菜单“Help->About…”，然后点击“Installation Details”，可以看到插件列表中显示 DTFJ 安装好。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

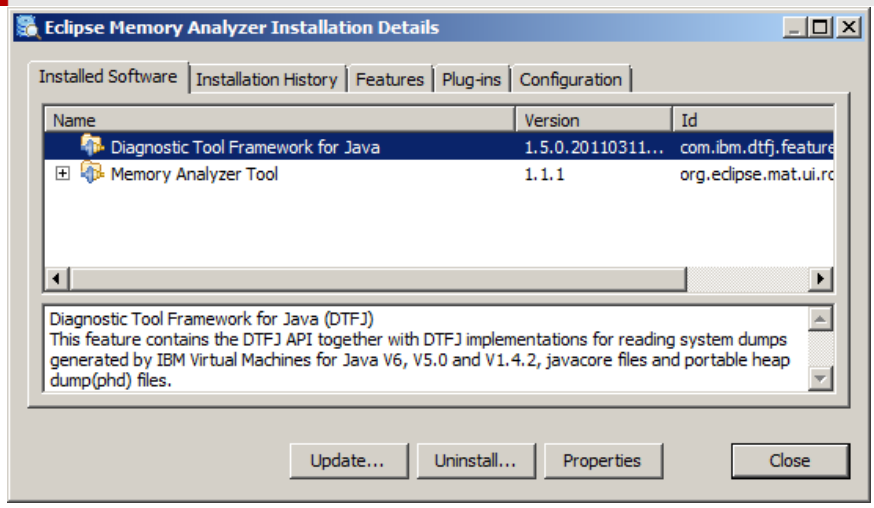


图 安装 DTFJ 插件

✓ 生成 dump 文件的方法与前文介绍的相同。

✓ 打开 dump 文件时，注意选择正确的文件类型。

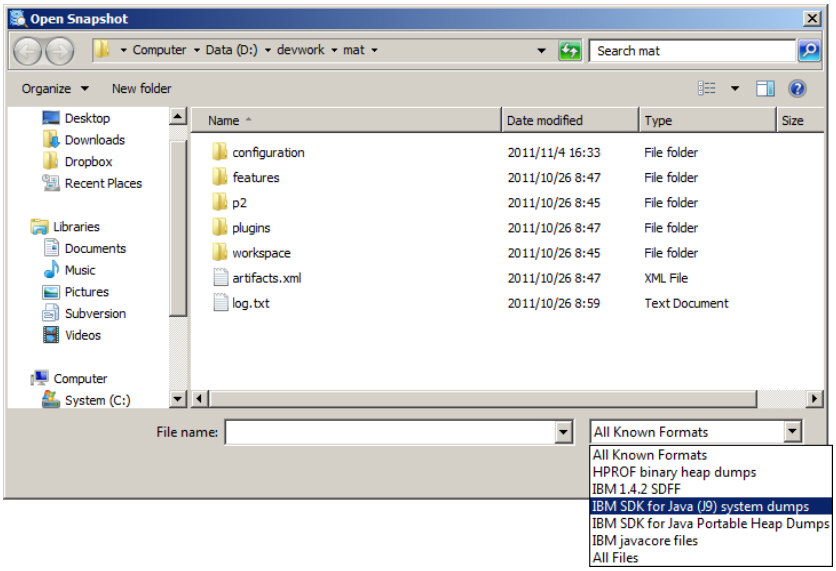


图 打开 dump 文件

8. 如何配置 Apache 代理服务器

理论上 DNA 可以配合任何支持 http1.1 协议的代理服务器来使用。但是实际上由于代理服务器实现存在的差异，可能会对 DNA 产生影响，所以如果使用了代理服务器，请进行全面测试。

修改 apache 的配置文件 httpd.conf 以实现以下几种模式：

1) 反向代理

```
#启用代理模块
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
```


公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199 传真：010-58561196
网址：http://www.jiuqi.com.cn 电子信箱：jiuqi@jiuqi.com.cn

```
<IfModule proxy_http_module>
  ProxyRequests Off
  ProxyVia Off
  ProxyPreserveHost On
  #配置前端和后端的url对应关系
  ProxyPass / https://localhost:9903/
  ProxyPassReverse / https://localhost:9903/
</IfModule>
```

2) 负载均衡

```
#启用代理模块
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
#启用负载均衡模块
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
#启用header模块
LoadModule headers_module modules/mod_headers.so

<IfModule proxy_http_module>
  ProxyRequests Off
  ProxyVia Off
  ProxyPreserveHost On
  #将路由信息保存到cookie中，用于将http会话路由到特定的服务器
  Header add Set-Cookie
"DNA_CLUSTER_NODE_INDEX=.%{BALANCER_WORKER_ROUTE}e; path=/"
env=BALANCER_ROUTE_CHANGED
  #配置前端和后端的url对应关系
  ProxyPass / balancer://web/
  ProxyPassReverse / balancer://web/
  #配置后端服务器组，"web"是自定义的名称
  <Proxy balancer://web>
    BalancerMember http://localhost:9701 route=1
    BalancerMember http://localhost:9702 route=2
    #保存路由信息的cookie名称
    ProxySet stickysession=DNA_CLUSTER_NODE_INDEX
  </Proxy>
</IfModule>
```

9. 如何启用 Apache 的 SSL 协议

1) 下载 openssl 工具

http://openssl.org/related/binaries.html

仅 Windows 需要下载此工具。Linux 平台自带。

公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

2) 生成证书文件

在命令行执行（Windows）：

```
openssl req -new -x509 -days 365 -sha1 -newkey rsa:1024 -nodes -keyout
server.key -out server.crt -subj
"/O=Company/OU=Department/CN=www.example.com"
```

如果是 Linux，请将命令中的双引号替换成单引号。

参数说明：

- ✓ -x509 指示 openssl 生成一个证书
- ✓ -days 指示证书的过期时间（单位为天）
- ✓ -sha1 指示使用 SHA1 算法对证书进行加密
- ✓ rsa:1024 指示生成 1024 位（bit）的密钥
- ✓ -nodes
- ✓ -keyout 指示 key 文件的文件名
- ✓ -out 指示证书的名称
- ✓ -subj 指示证书相关的信息。

该命令将生成两个文件：

- ✓ server.key 存储了密钥的文件
- ✓ server.crt 存储了证书的文件

3) 修改 Apache 配置

```
#启用SSL模块
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
#启用代理模块
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
#配置SSL，其中的路径指出了证书文件和密钥文件的位置
SSLEngine On
SSLCertificateFile "D:/dev/Apache2.2/conf/server.crt"
SSLCertificateKeyFile "D:/dev/Apache2.2/conf/server.key"
#配置代理
<IfModule proxy_http_module>
  ProxyRequests Off
  ProxyVia Off
  ProxyPreserveHost On
  #配置前端和后端的url对应关系
  ProxyPass / http://localhost:9902/
  ProxyPassReverse / http://localhost:9902/
```


公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn

```
</IfModule>
```


公司总部：北京经济技术开发区西环中路 6 号（100176）
技服中心：北京市西城区西直门内南小街国英一号 13 层（100035）
电话：010-88551199　　传真：010-58561196
网址：http://www.jiuqi.com.cn　电子信箱：jiuqi@jiuqi.com.cn