

技术重塑管理
数据驱动的企业

用友
yonyou

yonyou

UAP 大型企业与组织计算平台

Computing Platform for Large-sized Enterprises and Organizations

用友 AE 文件存储 产品使用手册



版权

©2014用友集团版权所有。

未经用友集团的书面许可，本文档任何整体或部分的内容不得被复制、复印、翻译或缩减以用于任何目的。本文档的内容在未经通知的情形下可能会发生改变，敬请留意。请注意：本文档的内容并不代表用友软件所做的承诺。

目录

版权	2
目录	3
1. 概述	5
1.1. 产品概述	5
1.2. 产品架构	6
1.2.1. 产品优势	6
1.2.2. 模块特性	7
2. 安装配置	8
2.1. 配置过程	8
2.2. 配置说明	19
2.3. 注意事项	21
2.3.1. 服务器地址和端口配置	21
2.3.2. 内外网场景中服务器地址和端口的配置	21
2.3.3. 分开部署时存储方式选择	22
2.3.4. 数据源配置	22
2.4. 安装测试	23
3. 分布式安装配置	24
3.1. 安装 LIBEVENT	24
3.2. 安装分布式文件系统	26
3.3. 安装 NGINX 以及 UDFS-NGINX-MODULE	26

3.4.	配置及启动分布式文件系统	27
3.5.	文件服务器中配置分布式文件系统	31
4.	产品升级	33
4.1.	文档中心升级	34
4.2.	文档中心升级设置	35
5.	文件导入导出.....	38
5.1.	导出设置	38
5.2.	参数设置	38
5.3.	存储设置	39
5.4.	导入设置	40
5.5.	注意事项	40

1. 概述

文件服务器的基本功能是实现文件的上传、下载、查询、删除等基本文件存储操，AE 文件服务器在实现了基本的存储功能同时，关注如何高效、合理的管理文件。AE 文件服务器基于元数据信息管理文件，实际文件则按照应用场景的需要存储在不同的存储介质上。

1.1. 产品概述

随着业务的高速发展，企业的需要处理的文件或者数据越来越多，如何合理、高效的管理这些文件显得尤为重要。

目前文件系统虽然能满足基本的文件上传、下载、查询等功能，但是这些操作基本上都是以文件本身为中心的，不方便业务人员高效的管理数据。

AE 文件服务器通过对文件元数据的分析与处理，能高效的管理文件，满足业务部门自己的查询、分析要求。同时提供了多种文件存储介质，满足不同的存储要求。

AE 文件服务器的总体层次结构如下图所示：

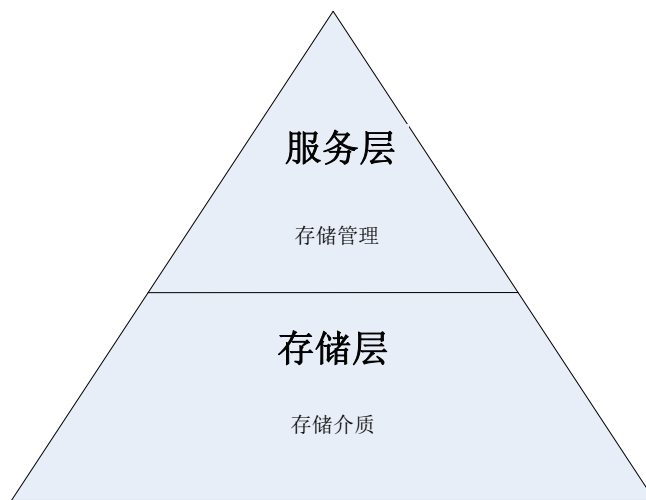


图 1 产品总体层次结构图

服务层：用于管理文件的元数据信息、文件的实际存储过程以及访问控制。文件的元数据信息目前保存在数据库，通过元数据可以方便的管理文件；存储管理主要通过不同的文件存储策略可以将文件存储在物理磁盘、分布式文件系统等存储介质上。访问控制防止用户绕过文件系统对文件进行访问；

存储层：文件的实际存储位置，支持多种存储介质。

AE 文件服务器总体模块结构如下图所示：

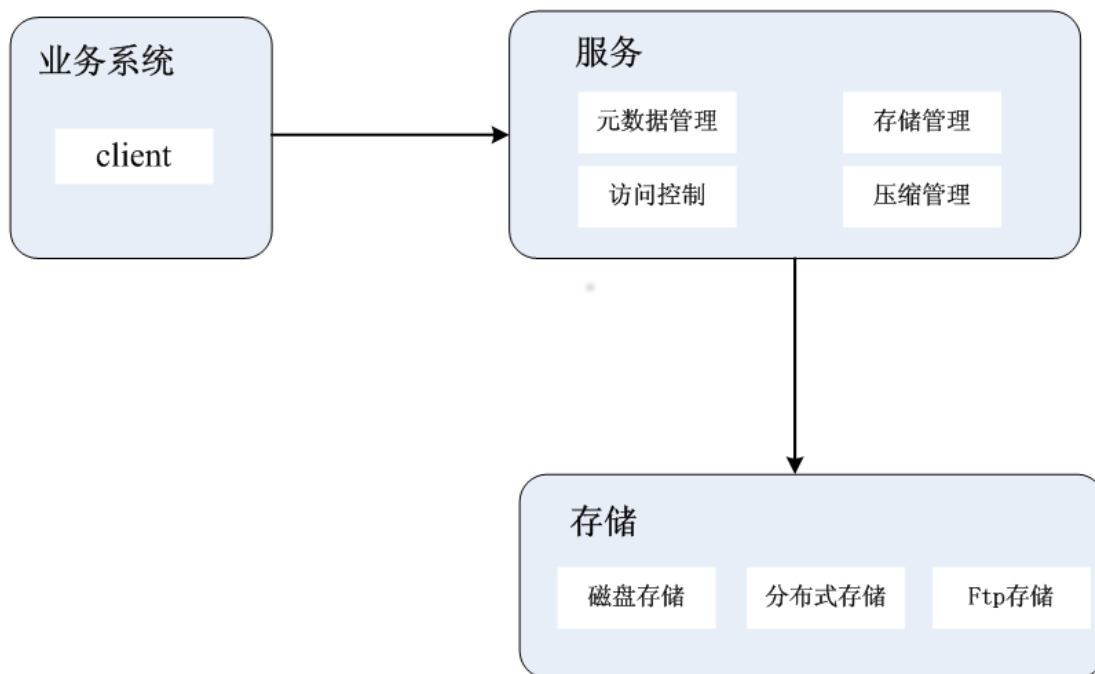


图 2 产品总体模块结构图

1.2. 产品架构

1.2.1. 产品优势

与其他版本的文件存储系统，AE 文件服务器有以下优势：

1. 扩展性好
 - 1) 基于元数据管理文件，提供扩展元数据信息，方便业务系统定制查询策略；
 - 2) 支持多种存储介质，满足不同的业务场景；
 - 3) 支持多种存储策略，方便管理文件；
2. 低成本
 - 1) 自主知识产权，无采购成本；
 - 2) 项目化研发团队，无二开成本；
 - 3) 界面化配置，无学习成本；
3. 技术优势

- 1) 基于元数据管理文件;
- 2) 支持多种存储策略;
- 3) 支持多种存储介质;

1.2.2. 模块特性

各模块特性如下文所述。

1. 元数据管理

- 1) 数据库存储;
- 2) 元数据包含文件基本信息、存储信息、业务系统定制信息
- 3) 支持上传、下载、删除、修改、查询等;

2. 存储管理

- 1) 存储策略
 - a) 支持多种存储策略 ;
 - b) 支持跨磁盘存储;
- 2) 存储介质
 - a) 支持多种存储介质存储 (普通磁盘、分布式存储、Ftp 存储等);

3. 访问控制

- 1) 防止用户绕过文件系统对文件进行访问;

4. 压缩控制

- 1) 支持文件压缩存储;

5. 磁盘存储

- 1) 支持磁盘存储, 允许跨磁盘存储;

6. 分布式存储

- 1) 支持文件的冗余、同步;
- 2) 支持请求负载均衡;
- 3) 支持动态扩展存储容量;
- 4) 容灾;

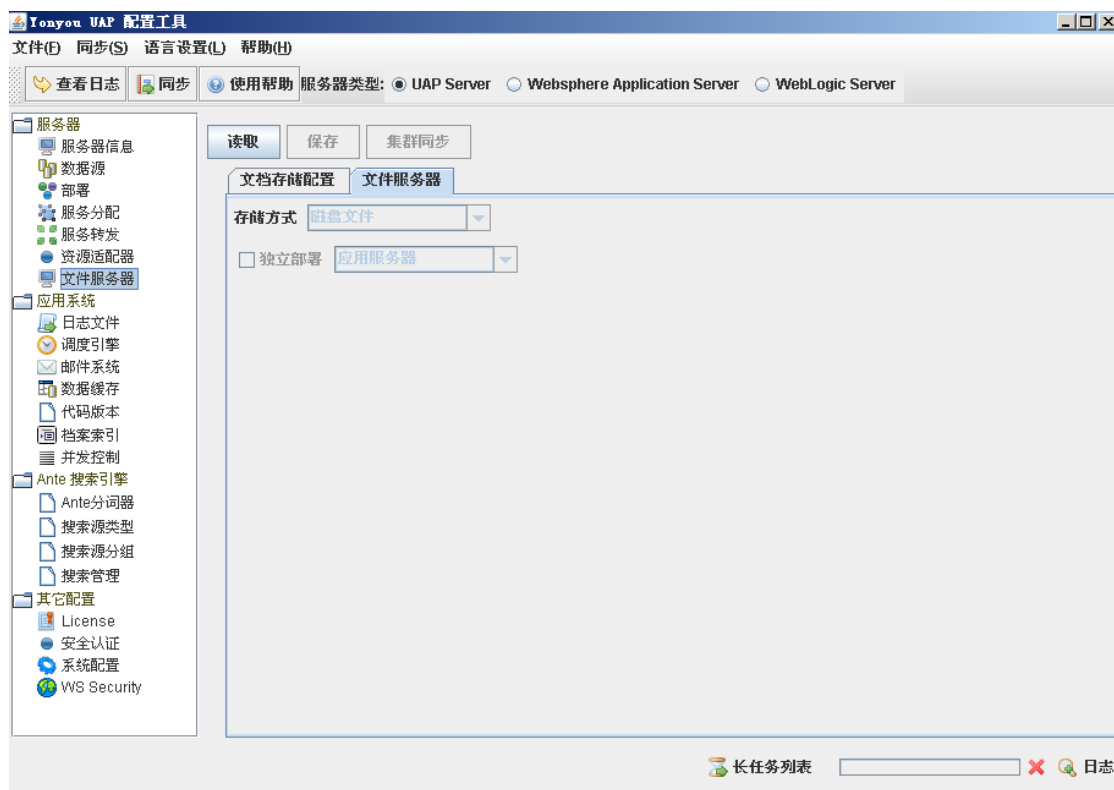
2. 安装配置

AE 件服务器支持独立部署。实施人员可以将文件服务器与应用服务器一起部署，也可以分开部署。两种部署方式的配置方式相同。

文件服务器支持文件上传、下载、查询、删除等一系列操作，为了能正常的使用文件服务，需要配置文件服务器的相关信息。本部分针对文件服务器相关配置进行简单的指导。

2.1. 配置过程

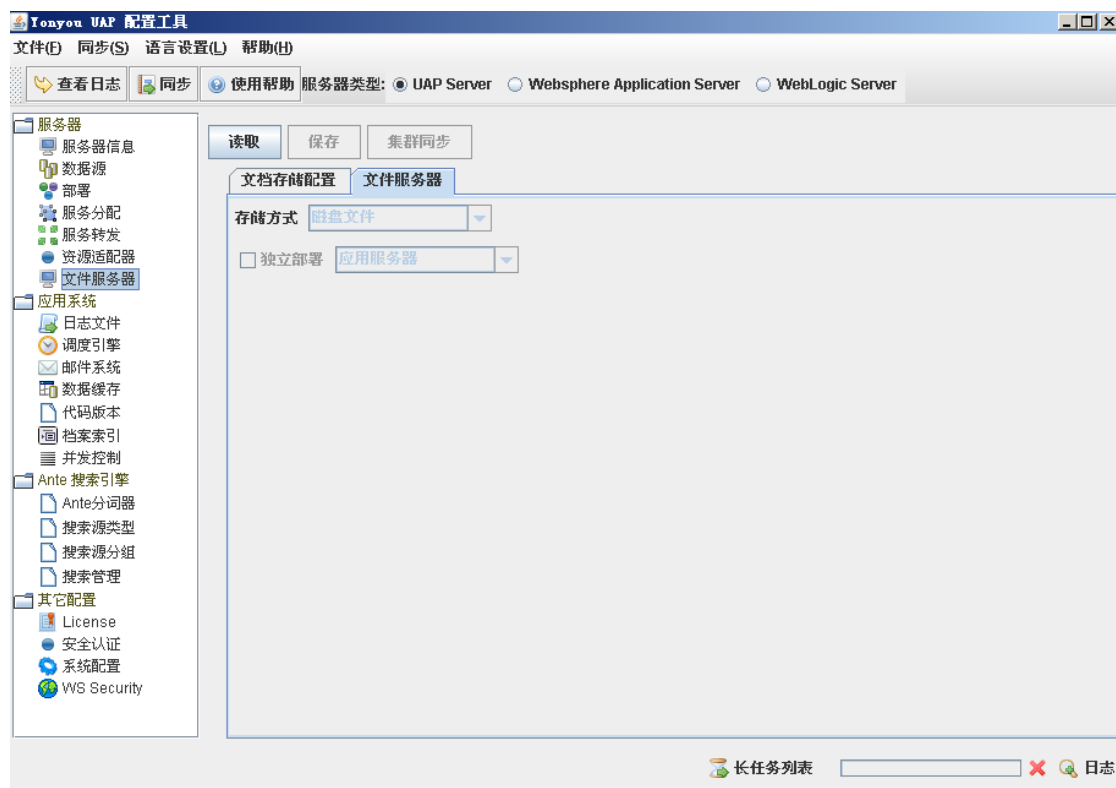
为了便于统一管理和方便用户使用，将 sysconfig 中原属于“系统设置”节点中的“文档存储设置”页签统一放到了“文件服务器”节点下。统一后的文件服务器的配置界面如下所示：



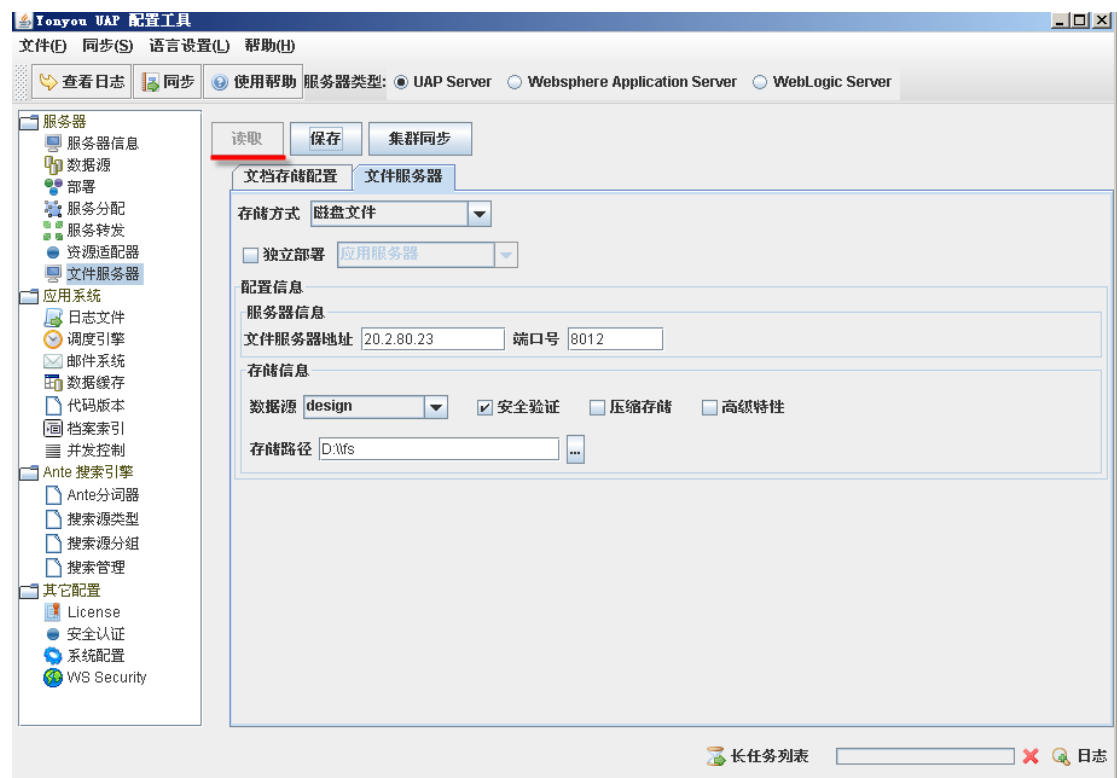
其中“文档存储配置”用于旧版文件服务器的配置，“文件服务器”页签用于本版本的文件服务器的配置。两套配置规则是相互独立的，不具有互斥性，可以同时配置。

在本版本的文件服务器中，支持两种存储类型——“磁盘文件存储”和“FTP 文件存储”。

运行\$UAP_HOME/bin/sysConfig.bat，选择“文件服务器”节点，可以看到如下的运行界面：

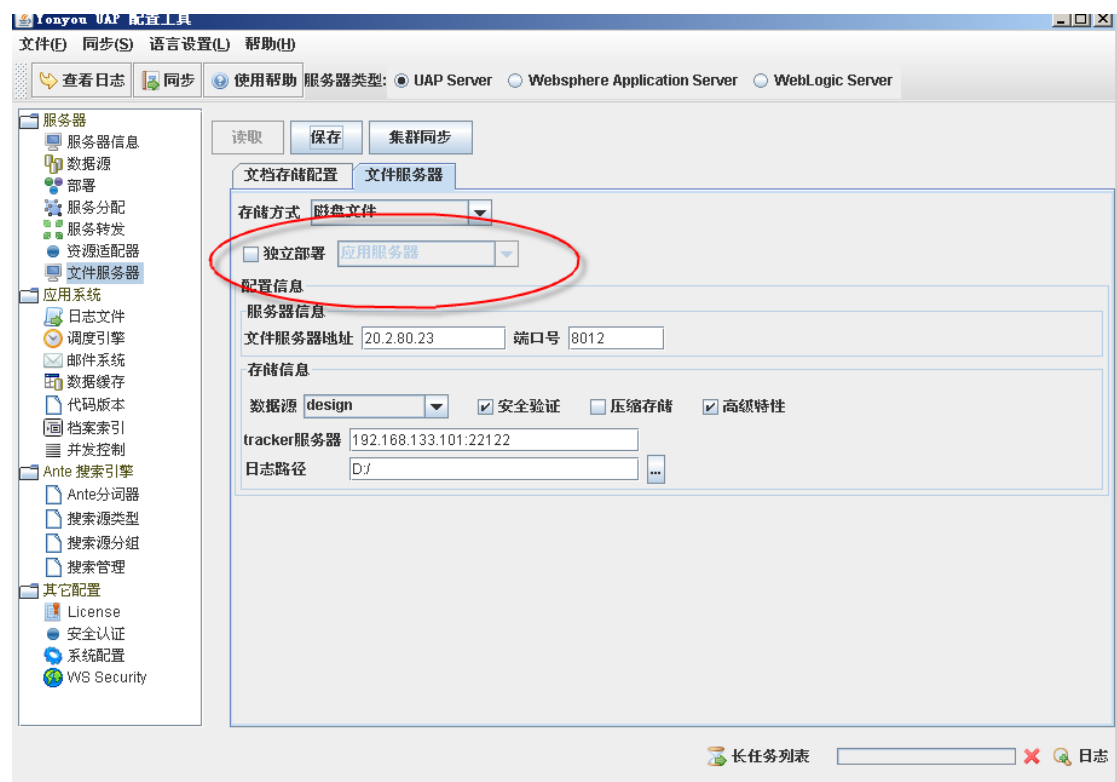


选择“文件服务器”页签，点击“读取”按钮，可以得到当前文件服务器的配置信息，如下所示：



本版本中，文件服务器的配置项涉及两个方面，第一个是文件服务器的**客户端配置项**，用于配置服务器的地址、端口号等配置信息。另一个是文件服务器的**服务端配置项**，主要设置文件的存储路径，数据源等配置信息。

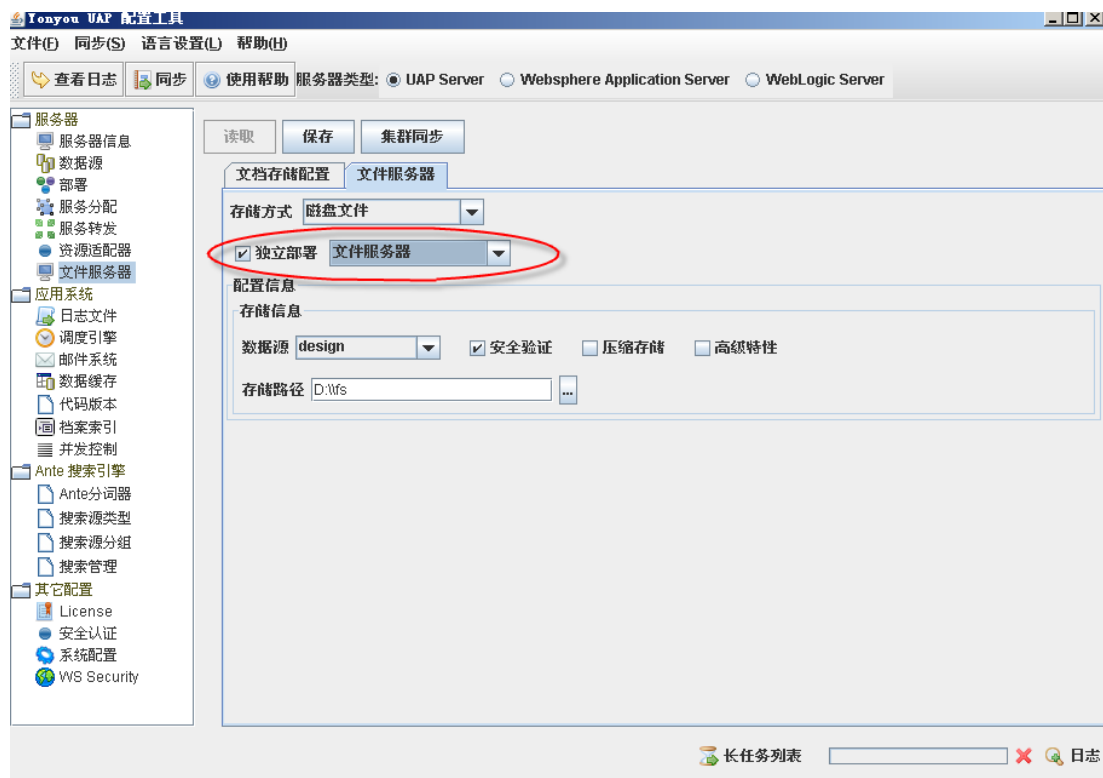
1. 如果文件服务器与应用服务器在同一台机器上，则在配置界面中则不勾选“独立部署”复选框。此时需要配置服务器信息和存储信息。如下图所示。



2. 如果文件服务器与应用服务器不再同一台机器上，即文件服务器的客户端（应用服务器）与文件服务器是分开独立部署，那么需勾选上“独立部署”复选框，从服务器类型中选择合适的服务器进行配置。应用服务器端配置如下：



文件服务器端配置如下：



下面我们分别详细的介绍“磁盘文件存储”和“FTP 文件存储”两种存储类型的配置过程。为了方便理解，我们介绍的是应用服务器和文件服务器分开部署的方式。这种方式与文件服务器和应用服务器一起部署的方式唯一不同在于后者不需要在配置界面勾选独立部署“。

(1) 磁盘文件存储的配置过程，

1) 应用服务器端



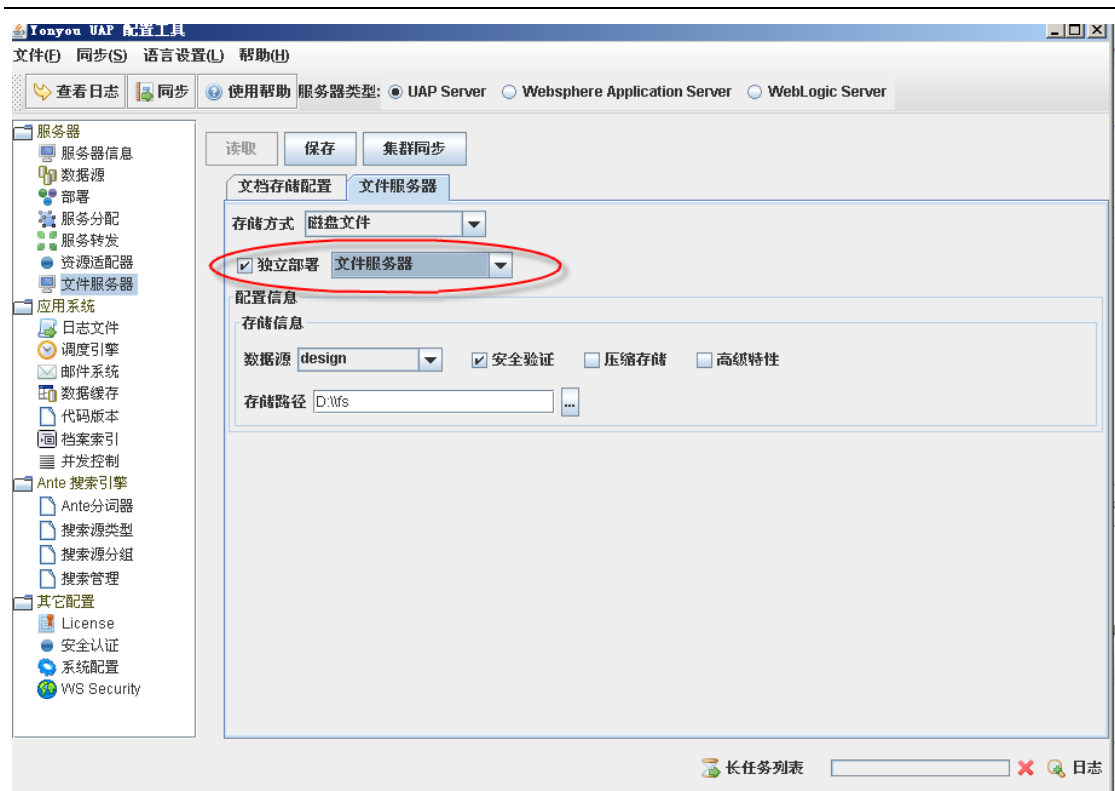
服务器地址：输入文件服务器的地址。

端口号：文件服务器启动文件服务的端口号

设置完毕后，点击“保存”按钮，保存配置信息。

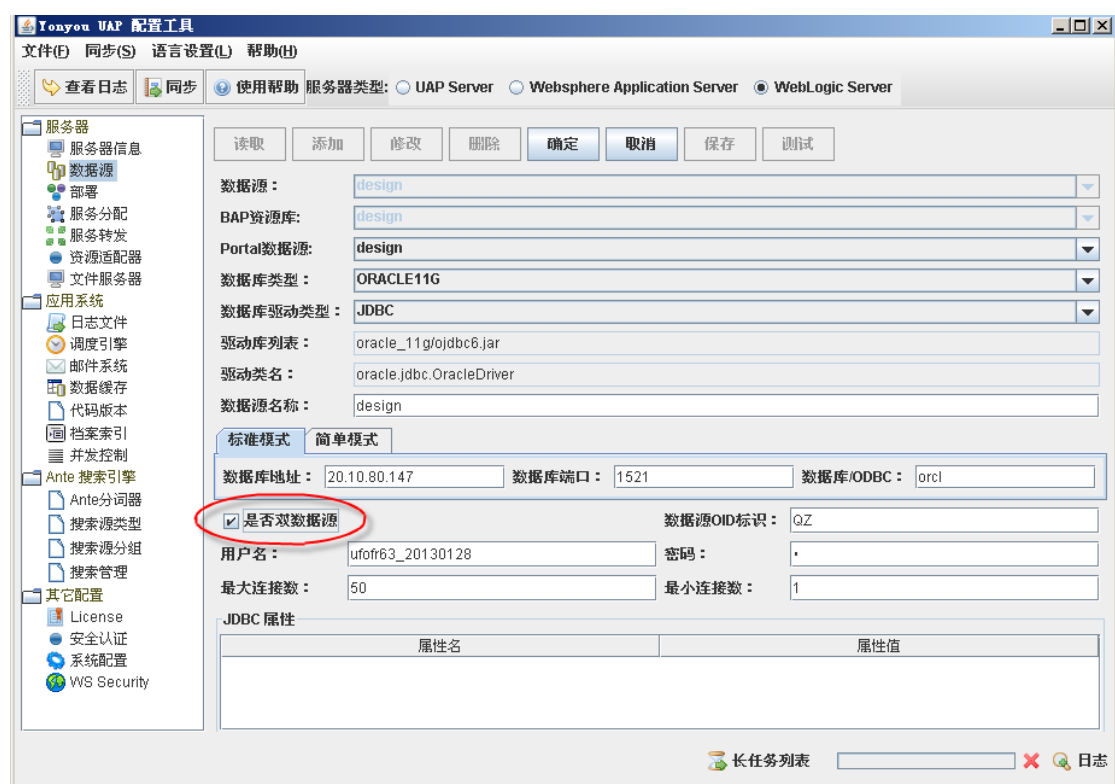
在集群环境下，点击集群同步按钮能将配置信息同步保存至集群各个节点中。

2) 文件服务器端



数据源：选择可用的数据源，用于保存文件的元数据信息。选择数据源时需保证数据源可用。

注意：如果登录系统时的系统数据源与文件服务器的数据源不是同一个，需要将文件服务器的数据源设置为双数据源。如下图所示：

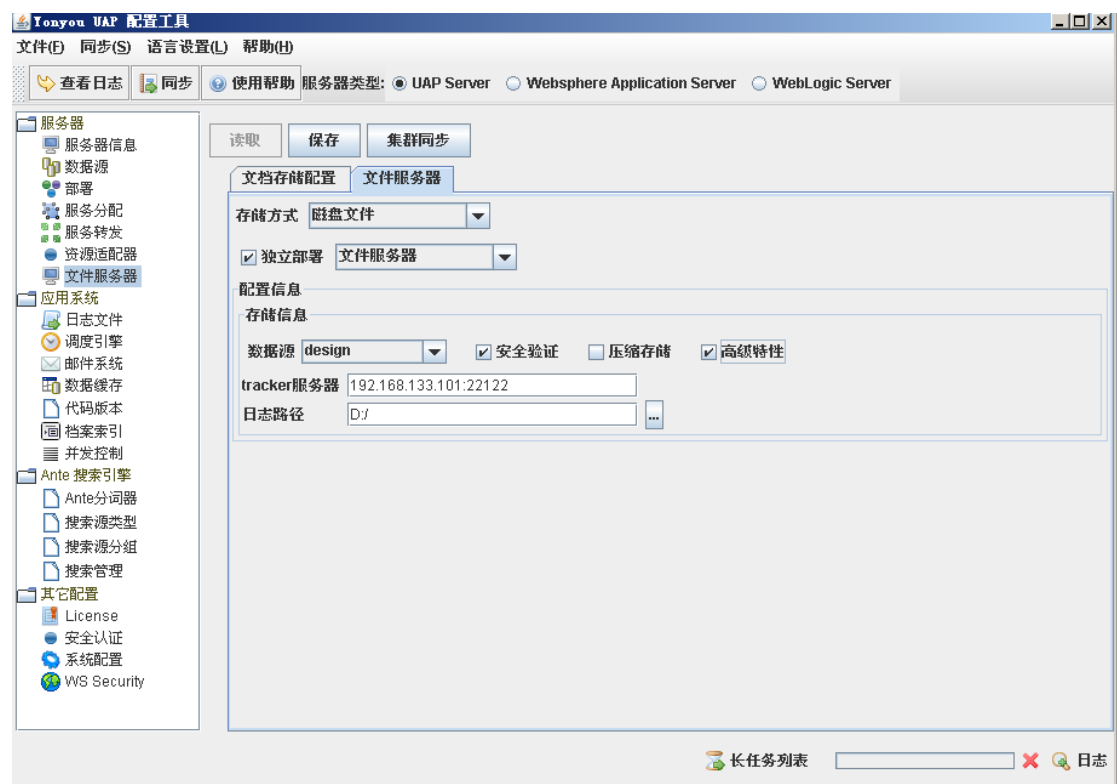


安全验证：选中此复选框，则启动安全验证，否则不进行安全验证。安全验证能防止其他用户绕过文件服务器，直接通过 url 方式下载文件。**注意：默认启动安全验证。**

压缩存储：选择此复选框，则在服务器端压缩存储上传后的文件，否则不压缩存储，按照原有类型存储。默认为非压缩存储。**注意：一旦设置了压缩存储，则不能改变，否则会导致文件下载、更新等操作失败。**

存储路径：指定了存储的根目录，所有的文件都存储在此目录下。**设置存储时建议不要手动输入，而是通过“文件选择框”完成。**

高级特性：磁盘文件存储包含两种存储类型 1) 普通磁盘文件存储 2) 分布式文件系统存储。选择此复选框文件支持分布式存储，否则支持本地磁盘存储，当为本地磁盘存储时，需要设置文件的存储路径，**默认为不显示高级特性**。下图是选择高级特性后的配置界面：



在本版本的文件服务器中，在使用高级特性前，需要搭建分布式文件系统的环境，并保证可用。否则文件服务器无法使用高级特性。

分布式文件系统的安装过程及详细配置说明见后续章节

(2) FTP 文件存储

1) 应用服务器端



服务器地址：输入文件服务器的地址。

端口号：文件服务器启动文件服务的端口号

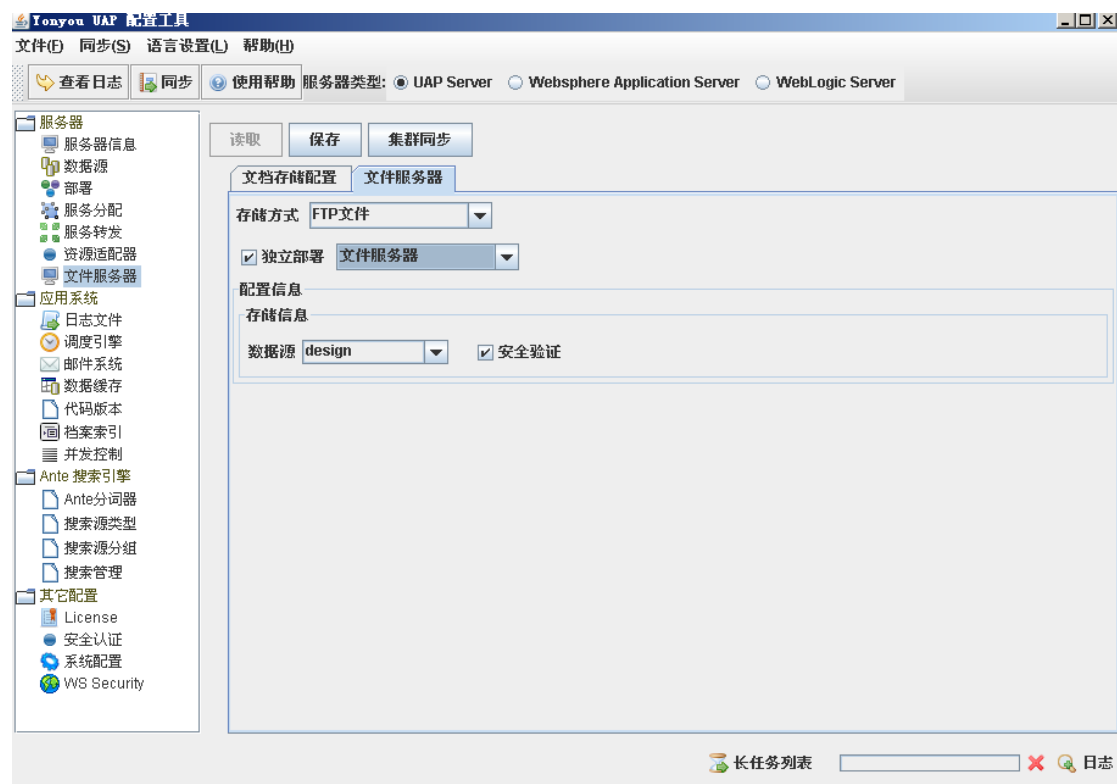
配置 FTP 相关配置项，如 FTP 服务器地址、端口号、登录的用户名、密码、文件的传输模式。这些配置项简单易懂，不再逐个介绍。

压缩存储：与磁盘存储方式不同，FTP 存储方式时，是否压缩存储在客户端设置。选择此压缩存储复选框，则在 Ftp 服务器上压缩存储文件，否则不压缩存储，按照原有类型存储。默认为非压缩存储。**注意：一旦设置了压缩存储，则不能改变，否则会导致文件下载、更新等操作失败。**

设置完毕后，点击“保存”按钮，保存配置信息。

在集群环境下，点击集群同步按钮能将配置信息同步保存至集群各个节点中。

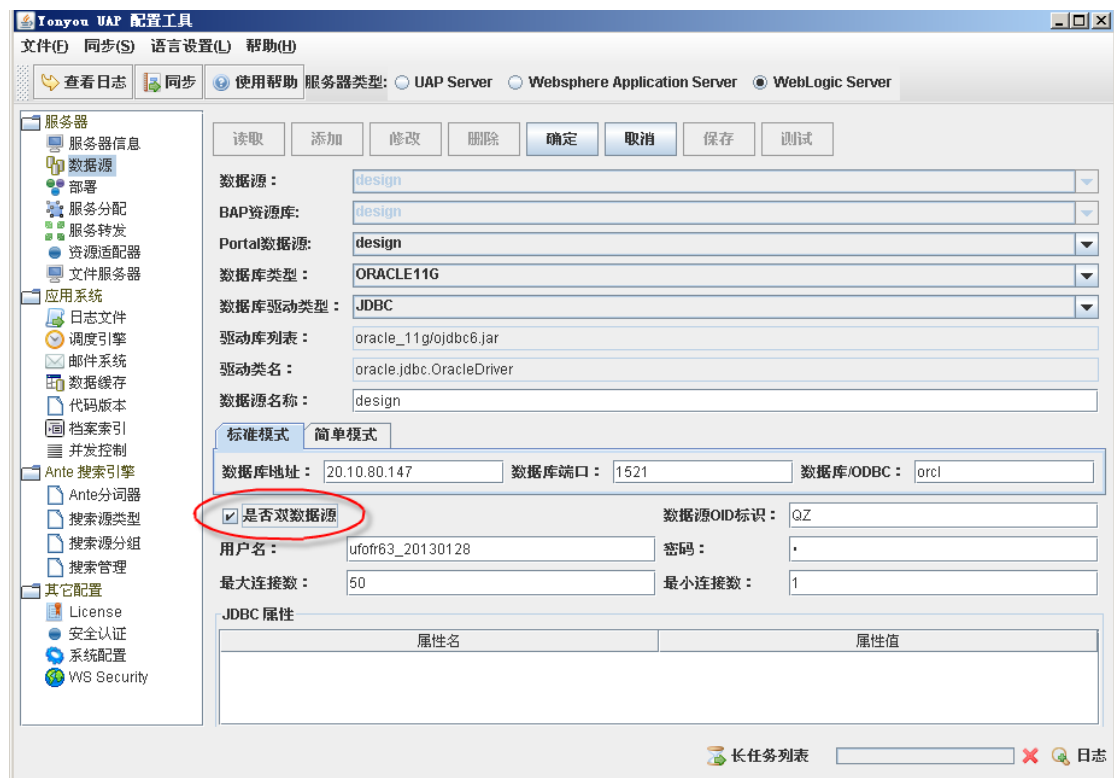
2) 文件服务器端



安全验证：选中此复选框，则启动安全验证，否则不进行安全验证。安全验证能防止其他用户绕过文件服务器，直接通过 url 方式下载文件。**默认启动安全验证。**

数据源：选择可用的数据源，用于保存上传文件的元数据信息。选择数据源时需保证数据源可用。

注意：如果登录系统时的系统数据源与文件服务器的数据源不是同一个，需要将文件服务器的数据源设置为双数据源。如下图所示：



设置完毕后，点击“保存”按钮，保存配置信息。

在集群环境下，点击集群同步按钮能将配置信息同步保存至集群各个节点中。

2.2. 配置说明

1) 文件服务器中配置文件位置

AE 文件服务中的涉及了四个配置文件，分别为 fileserver.properties、disk.properties、fdfs.properties、ftp.properties。这四个配置分别为记录了文件服务器通用配置项、磁盘存储配置项、分布式存储配置项、Ftp 存储配置项。这四个配置文件统一放在如下目录下：

\$UAP_HOME\ierp\s\ fileserverconfig

2) 文件服务器配置项说明

除了上文所述的配信息，在文件服务器的配置文件中还包含其他配置信息，这些配置信息可以通过手动修改配置文件来设置。下面就详细的介绍这些没有在界面中显示的配置信息

1. 服务器通用配置项

1) override

override 代表出现同名文件时，是否覆盖旧文件。

override=true;表示覆盖旧文件

override=false;表示不覆盖旧文件，后台存储时新建一个文件。

默认为 false。

注：当使用“磁盘文件存储”时，在服务端设置此配置项。当使用“FTP 文件存储”时在客户端配置此配置项

2) displayByName

上传、更新文件操作后，文件服务会给返回给调用者一个标识信息，

displayByName 用来设置此标识信息的格式。

displayByName=true;返回的标识为文件名称

displayByName=false; 返回的标识为一串 PK 值。

默认为 false。

注：由于中文在传输的过程中存在乱码问题，因此强烈建议此处设为 false。

“磁盘文件存储”和“FTP 文件存储”方式都在服务器配置此配置项

2. 磁盘存储配置项

1) defaultFileRepo

是 filePath 的冗余项，如果未设置 filePath，则使用此配置项作为文件存储的根目录

注：“磁盘文件存储”和“FTP 文件存储”方式都在服务器配置此配置项

3. 分布式存储配置项

分布式存储的相关配置说明见后续章节。

2.3. 注意事项

2.3.1. 服务器地址和端口配置

如果文件服务器与应用服务器在一台机器上，即本地应用服务器提供文件服务，那么输入应用服务器的 ip 地址，端口号为应用服务器启动服务的端口号。如果文件服务在远程服务器上，即应用服务器与文件服务器分开部署，那么文件服务器地址为远程文件服务器服务器的地址，端口号为文件服务器启动服务的端口号。

在配置 ip 地址和端口号时需要注意以下几点：

1. 服务器地址项不能配置成 127.0.0.1 或者 localhost，必须配置成真实的 ip 地址。
2. 目前文件服务器地址只支持 ip，不支持机器名。
3. 在集群环境下，不要使用 webserver、master 的 ip 地址和端口号，而应指定一个固定的 ncMem 的 IP 地址和端口号作为文件服务器服务的地址和端口，或者使用一台独立的服务器作为文件服务器（此服务器不在集群中，独立部署）。（如果是垂直集群，则可以使用 master 地址，但是端口号不能使用主端口号，需要指定一个 ncMem 的端口号作为文件服务器的端口）

2.3.2. 内外网场景中服务器地址和端口的配置

机器 A 做了内外网隔离，文件服务部署在内网上，如 192.100.1.155:8080，外部无法直接访问到内网的服务。使用环境在外网，地址为 20.2.64.57:8080，内外网通过 ip 端口映射来完成。当外部访问环境 20.2.64.57:8080 时，会自动映射到内网 192.100.1.155:8080 上，从而获取文件服务。这种场景下，文件服务器的配置需要注意以下几点：

- 1) 服务器的 ip 地址和端口号应配置为外网的 ip 地址和端口号。

在上述场景中，文件服务器地址为 20.2.64.57 端口为 8080

- 2) 检查内外网的网络设置，保证内外网能相互连接。

在上述场景中，保证在机器 A 的外网环境能连接到内网的 8080 端口。同时内网环境也能连接到外网的 8080 端口

- 3) 检查防火墙设置
- 4) 检查用户能否连接文件服务器

在上述场景中，保证用户所在的网段（用户本地地址）能连接上文件服务器所在机器的 ip 地址（20.2.64.57）。

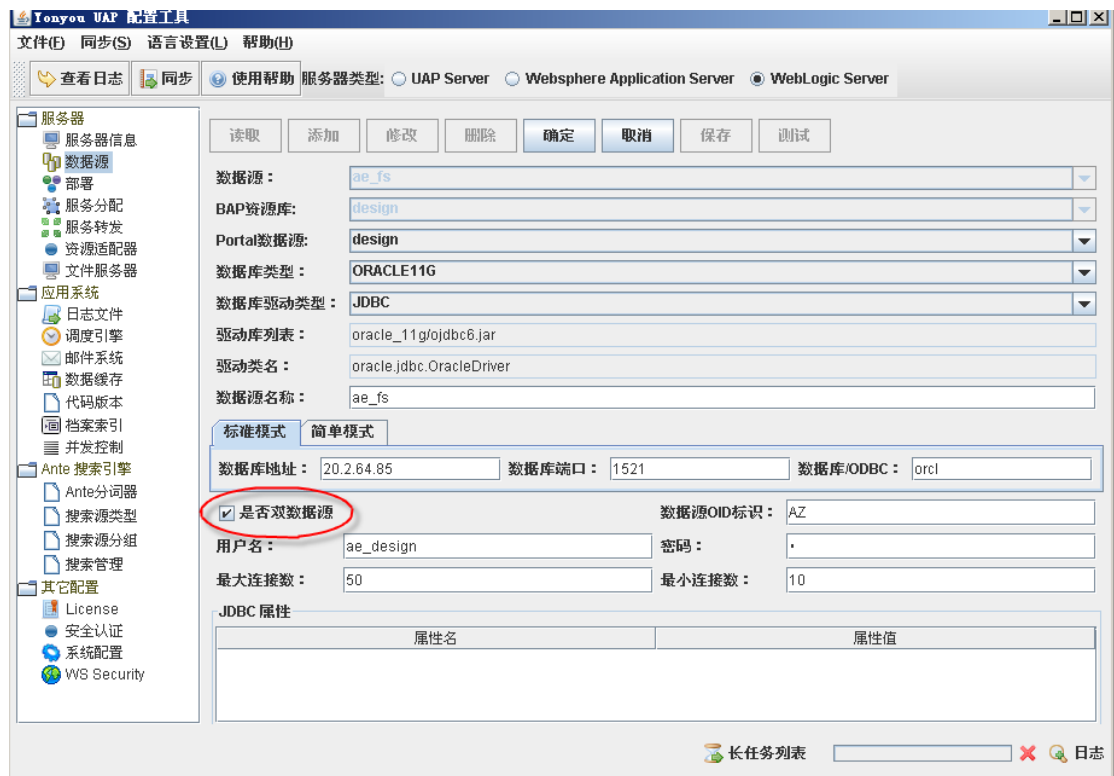
2.3.3. 分开部署时存储方式选择

客户端和服务端选择的存储类型必须保持一致，如果在客户端选择了某种文件存储方式，那么在服务器端也必须选择这种存储方式。如，在客户端选择了“磁盘文件存储”，那么在服务器端也必须选择“磁盘文件存储”，否则会导致文件服务器无法正常使用。

2.3.4. 数据源配置

文件服务器允许用户自己配置数据源，如果用户配置的文件服务器的数据源与系统的用户数据源不是同一个，需要将文件服务器的数据源配置成双数据源。

如下图所示：

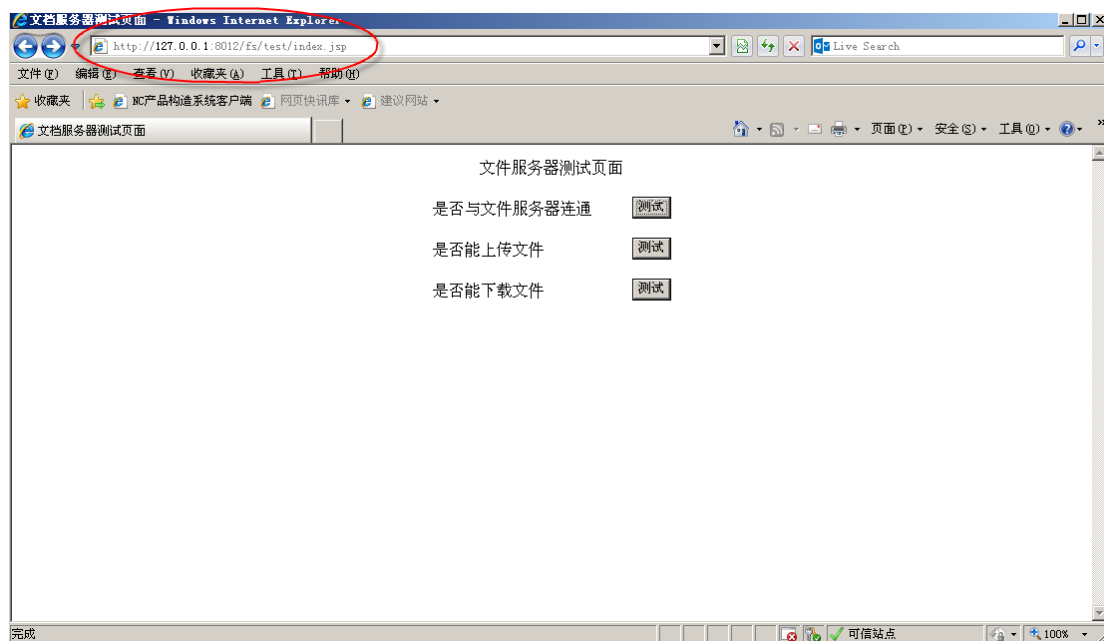


2.4. 安装测试

本版本的文件服务器中增加了简易的用于测试文件服务器是否可用的页面，在浏览器中输入如下形式的 url 地址即可跳转至文件服务器测试页面

<http://ip:port/fs/test/index.jsp>

在上面的 url 中 ip 和 port 指的是应用服务器的 ip 地址和端口号，而非文件服务器的 ip 地址和端口号。测试页面如下所示：



在上述的测试页面中，应用服务器在本地，服务端口号为 8012。文件服务器地址通过配置界面配置为远程地址。

测试页面的地址为 <http://127.0.0.1:8012/fs/test/index.jsp>。

3. 分布式安装配置

本章详细的介绍了分布式系统的最小集群部署过程。集群环境如下：

tracker: 20.2.64.133 。用于调度工作，在访问上起负载均衡的作用。

group1:

storage1: 20.10.129.150。存储节点存储文件，完成文件管理的所有功能。

storage2: 20.12.1.73。

group2:

storage2:20.12.1.74

以下将详细的介绍此分布式的安装过程，针对不同的系统，安装过程中部分细节（存储路径设置等）可能稍有不同，但基本步骤相同。

3.1. 安装 libevent

在搭建分布式环境之前，需要确保已经安装了 libevent，安装分布式系统时对 libevent 的版本有一定要求。注需要卸载重装系统自带的 libevent。使用最新的 **stable 版本**。本示例中使用的版本为 libevent-2.0.21-stable。注意，此步骤需要在所有机器上都执行，否则会导致安装失败安装失败。

1: 判断是否安装了 libevent

```
>> rpm -qa|grep libevent
```

如果有输出，则说明已经安装了 libevent

本分布式系统对 libevent 版本有要求，所以建议先删除系统自带的 libevent, 然后安装最新稳定版本

在本示例中，输出为 libevent-1.4.13-1.el6.x86_64

2: 卸载已有的 libevent

```
>> rpm -e libevent -nodeps
```

3: 安装最新的 libevent

本示例的 libevent 使用版本为 libevent-2.0.21-stable.tar.gz

安装 libevent

```
>> tar zxvf libevent-2.0.21-stable.tar.gz
```

```
>> cd libevent-2.0.21-stable
```

```
>> ./configure --prefix=/usr
```

```
>> make
```

```
>> make install
```

4: 检查是否安装成功

```
>> ls -al /usr/lib | grep libevent(或者>> ls -al /usr/local/lib | grep libevent)
```

若有结果，则说明安装正确

注意:

1) 如果是在 32 为系统上安装，且 libevent 的安装目录在 /usr/local/lib 下，则还需要建立 libevent-2.0.so.5 到 /usr/lib 的软链接，这样其他程序运行时才可以找到 libevent 库

```
>> ln -s /usr/local/lib/libevent-2.0.so.5 /usr/lib/libevent-2.0.so.5
```

2) 如果是在 64 位系统，需要创建一个 libevent-2.0.so.5 到 /usr/lib64 的软链接

```
>> ln -s /usr/lib/libevent-2.0.so.5 /usr/lib64/libevent-2.0.so.5 (假设 libevent 的安装目录在 /usr/lib 下)
```

到此一步，安装 libevent 工作完毕。

3.2. 安装分布式文件系统

```
>> tar zxvf UDFS_v4.06.tar.gz
```

```
>> cd UDFS
```

```
>> ./make.sh
```

```
>> ./make.sh install
```

注意上面两步，检查是否出错，如果出错，则说明上面的 libevent 没有安装好

安装成功后，分布式文件系统安装在/usr/local/bin 中。配置文件在/etc/fdfs 中

至此，安装完毕

3.3. 安装 nginx 以及 udfs-nginx-module

分布式系统通过 HTTP 服务器来提供 HTTP 服务。为了支持高并发的访问，以及提供负载均衡等高性能的服务，本示例使用 nginx 作为 HTTP 服务器，此分布式系统提供了支持 Nginx 的模块。此步操作只需要在 **storage** 上执行。**tracker** 上可不执行。

1: udfs-nginx-module 安装:

```
>>tar -zxvf udfs-nginx-module_v1.11.tar.gz    (解压后的路径为  
/home/chenfeic/fdfs/udfs-nginx-module)
```

nginx 需要依赖第三方的库，nginx 的 gzip 模块需要 zlib 库，rewrite 模块需要 pcre 库，因此在安装 nginx 之前需保证这些第三方库安装完成。

2: zlib 库安装:

```
>> tar -vxzf zlib-1.2.8.tar.gz    (本示例中解压位置为/home/chenfeic/udfs/zlib-1.2.8)
```

```
>> ./configure --prefix=/usr/local/zlib    (设置安装路径)
```

```
>> make
```

```
>> make install
```

3: pcre 库安装:

```
>> tar -vxzf pcre-8.33.tar.gz （本示例中解压位置为--with-pcre= /home/chenfeic/ fdfs  
/ pcre-8.33 ）
```

```
>> cd pcre-8.33
```

```
>> ./configure --prefix=/usr/local/pcre
```

```
--libdir=/usr/local/lib/pcre --includedir=/usr/local/include/pcre
```

```
>> make
```

```
>> make install
```

4: nginx 安装:

```
>> tar -zxvf nginx-1.4.4.tar.gz
```

```
>> cd nginx-1.4.4
```

```
>> ./configure --prefix=/usr/local/nginx --with-zlib=/home/chenfeic/fdfs/zlib-1.2.8
```

```
--with-pcre=/home/chenfeic/fdfs/pcre-8.33 --sbin-path=/usr/local/nginx
```

```
--add-module=/home/chenfeic/fdfs/udfs-nginx-module/src
```

```
>> make
```

```
>> make install
```

注意: --with-pcre --with-zlib 后面的路径都是源码路径, 而不是编译后的包。在实际的过程中, 此路径根据具体情况填写。

至此, 我们的 nginx 安装完成, nginx 默认是 80 端口, 为了防止与其他服务冲突, 建议换成其他端口 (后续章节有详细介绍)。

3.4. 配置及启动分布式文件系统

1: tracker 配置 (只需在跟踪器节点配置)

修改 tracker 的配置文件 tracker.conf。该文件位置在 /etc/fdfs/tracker.conf。主要修改以

下两处:

```
>> /etc/fdfs/tracker.conf
```

base_path=/home/chenfeic/fdfs/tracker ====> 放置 data 和 log 的目录。这个路径必须存在。

##include http.conf 修改为#include http.conf ====> 这里一定要注意! 是 #include, 不是 include!!! 如果没有此配置项, 则不需要处理

```
rotate_error_log=true
```

启动 tracker

```
>> /usr/local/bin/fdfs_trackerd /etc/fdfs/tracker.conf
```

检查是否正常启动

```
>> netstat -an|grep -w fdfs
```

看是否有输出, 且 22122 端口已启动

2: storage 配置 (只需在存储节点配置)

1) 修改 nginx 端口 (可选)

nginx 默认端口号为 80, 为了防止冲突, 可以讲设置为其他端口, 本示例中将其修改为 8080。

修改/usr/local/nginx/conf/nginx.conf

```
>> vim /usr/local/nginx/conf/nginx.conf
```

```
Server {
    listen 8080(从 80 改为 8080)
}
```

2) nginx 支持分布式系统

在 nginx 的 **server 配置段中**增加 /group{i}/M00 的 location 声明

```
storage $> vim /usr/local/nginx/conf/nginx.conf
```

```
location /group1 /M00 {
```

```
root /home/chenfeic/ fdfs /storage/data;

ngx_udfs_module;

}
```

注意: 此处如果是在 group1 就写成 group1/M00。如果是 group2 节点就写 group2/M00。依次类推。

/home/chenfeic/ fdfs /storage/ 为 storage 中配置的 store_path0 路径, data 目录一定要加上, 且必须保证路径存在, 若不存在就创建

注意这两行都有分号

给 storage 的存储目录做一个软连接

```
>>ln -s /home/chenfeic/ fdfs /storage/data /home/chenfeic/ fdfs /storage
/data /M00
```

3) 修改 storage 配置

修改 storage 的配置文件 storage.conf。该文件位置在 /etc/fdfs/storage.conf。主要修改以下几如有其他调整, 其他的可以使用默认的配置。

```
>> vim /etc/fdfs/storage.conf
```

group_name=group1 ====> 此台 storage server 所属的服务器组名。**此处如果是在 group1 就写成 group1。如果是 group2 节点就写 group2。依次类推。**

base_path=/home/chenfeic/ fdfs /storage ====> 放置 data 和 log 的目录。**此路径必须保证存在。**

store_path0=/home/chenfeic/ fdfs /storage ====> 放置文件的目录, **此路径必须保证存在, 建议跟 base_path 保持一致 (/home/chenfeic/ fdfs /storage 与 nginx server 段中的设置保持一致,)**

```
rotate_error_log=true
```

tracker_server=20.2.64.133:22122 ====> tracker server 的 ip 和端口,

此处可以写多个 tracker server，每行一个，因为本示例中只有一个 tracker，所以只需写一个。

http.disabled=true ====> 关闭内置的 web server，如果有此配置项就设置，否则就不用设置。

http.server_port=8080 ====> web server 的端口改成 8080（与 nginx 端口一致）

4) 修改 udfs-nginx-module 配置

拷贝 mod_udfs.conf 到/etc/fdfs 目录下

```
>>cp /home/chenfeic/ fdfs /udfs-nginx-module/src/mod_udfs.conf /etc/fdfs/
```

5) 修改 mod_udfs.conf 配置文件

```
>> vim /etc/fdfs/mod_udfs.conf
```

base_path=/home/chenfeic/ fdfs /storage ====> 放置 log 的目录, 与 storage 中的配置保持一致。

tracker_server=20.2.64.133:22122 ====> tracker server 的 ip 和端口，此处可以写多个 tracker server，每行一个，因为本示例中只有一个 tracker，所以只需写一个。

group_name=group1 ====> 此台 storage server 所属的服务器组名。**此处如果是在 group1 就写成 group1。如果是 group2 节点就写 group2。依次类推。**

url_have_group_name = true ====> 在 URL 中包含 group 名称

store_path0=/home/chenfeic/ fdfs /storage ====> 放置文件的目录，建议与 base_path 保持一致

group_count=2 ====>group 的个数，在本示例中有两个 group, 所以设为 2

到此，配置已经完成，启动 nginx 和相关 storage

```
>>usr/local/nginx/nginx
```

```
>>/usr/local/bin/fdfs_storage /etc/fdfs/ storage.conf
```

检查是否正常启动

```
>>netstat -anp|grep -w 8080 看 8080 端口是否启动
```

3: 测试是否正常安装

此步骤在 tracker 或者 storage 都可以测试, 本示例中在 storage 上测试, 修改 client.conf 配置文件。

```
>>vim /etc/fdfs/client.conf
```

```
base_path=/home/chenfeic/fdfs
```

```
tracker_server=20.2.64.133:22122
```

测试:

```
>>vim hello.txt (创建一个测试的文件)
```

```
>>/usr/local/bin/fdfs_test /etc/fdfs/client.conf upload hello.txt
```

查看结果, 看文件是否正确上传

可以在任何 storage 机器上执行如下命令, 查看当前集群状态, 如果各个 storage 状态都为 ACTIVE, 则搭建成功。

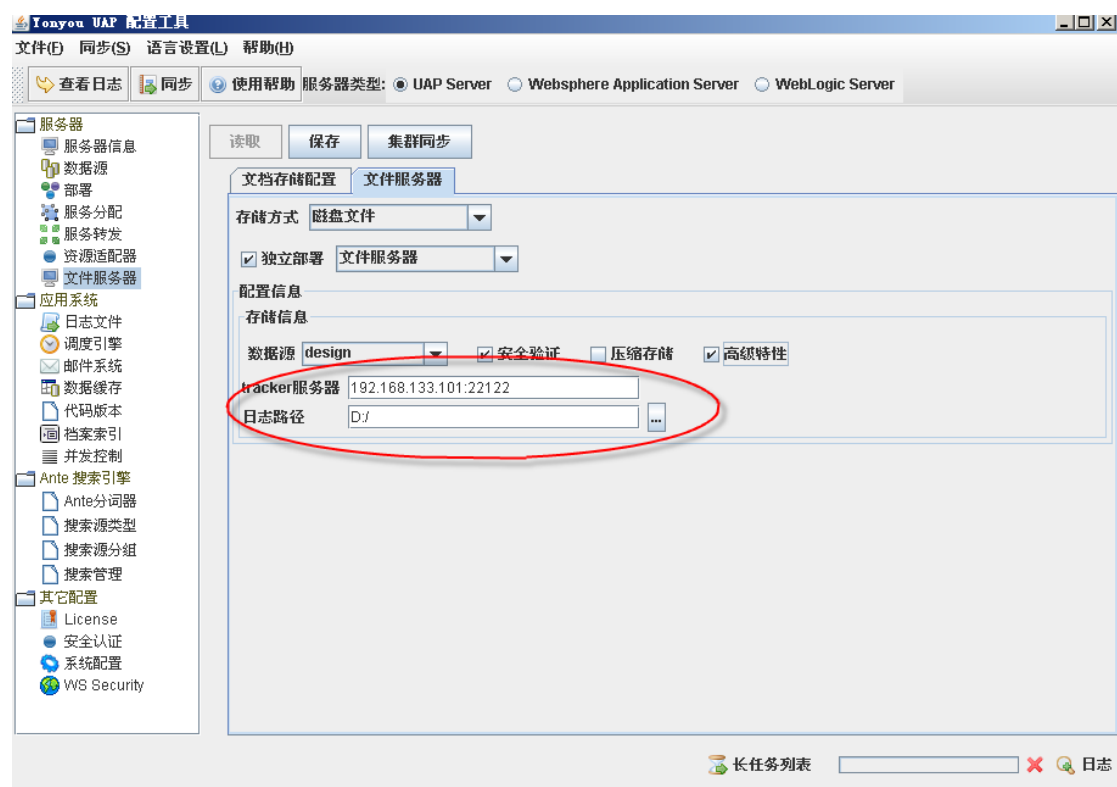
```
/usr/local/bin/fdfs_monitor /etc/fdfs/storage.conf
```

注意: 在配置过程, 要注意防火墙的设置。关闭防火墙, 或者设置对相关端口例外。

3.5. 文件服务器中配置分布式文件系统

上述操作中, 详细的介绍了分布式文件系统的搭建和启动过程, 为了使文件服务器能连接上分布式系统, 文件服务器需要设置分布式系统的地址等信息, 下面将详细的介绍这些配置项。

在文件服务器选择磁盘文件存储中的高级选项, 即选择使用分布式文件系统存储文件。如下图所示:



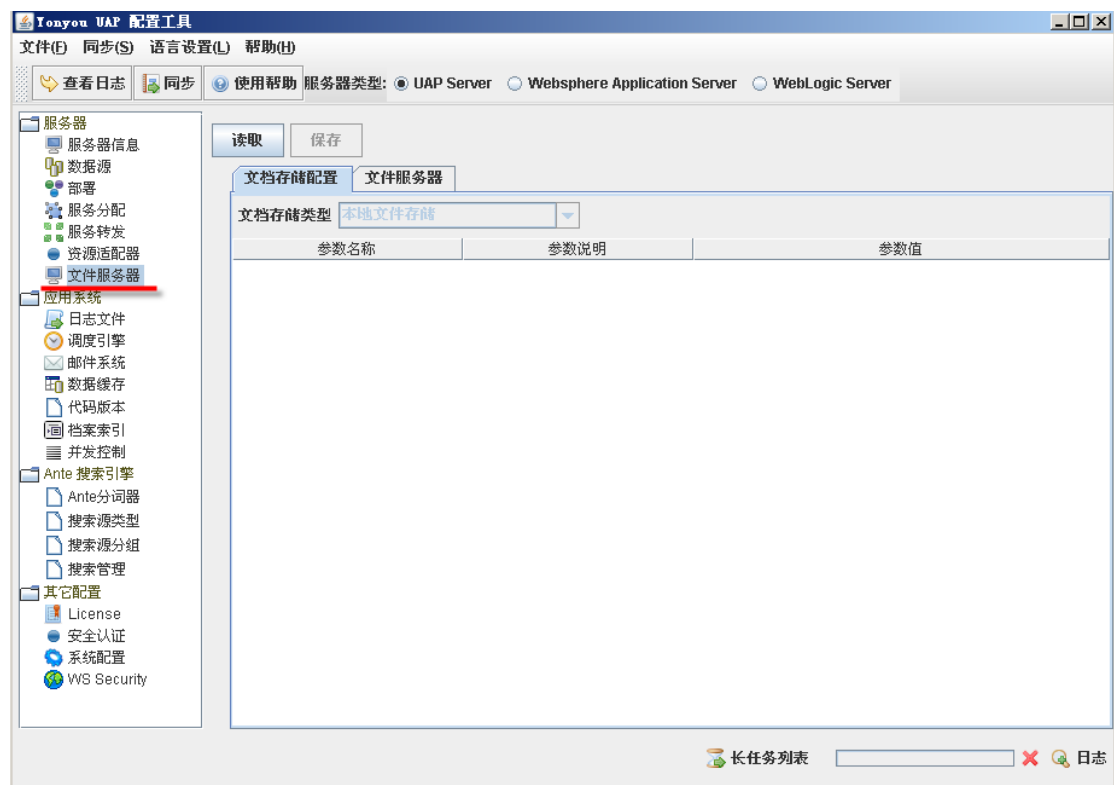
tracker 服务器：即分布式文件系统中 tracker 服务器的地址和端口，本示例中因为 20.2.64.133: 22122。

日志路径：存放文件上传下载过程的日志记录。在设置的时候需保证路径存在。

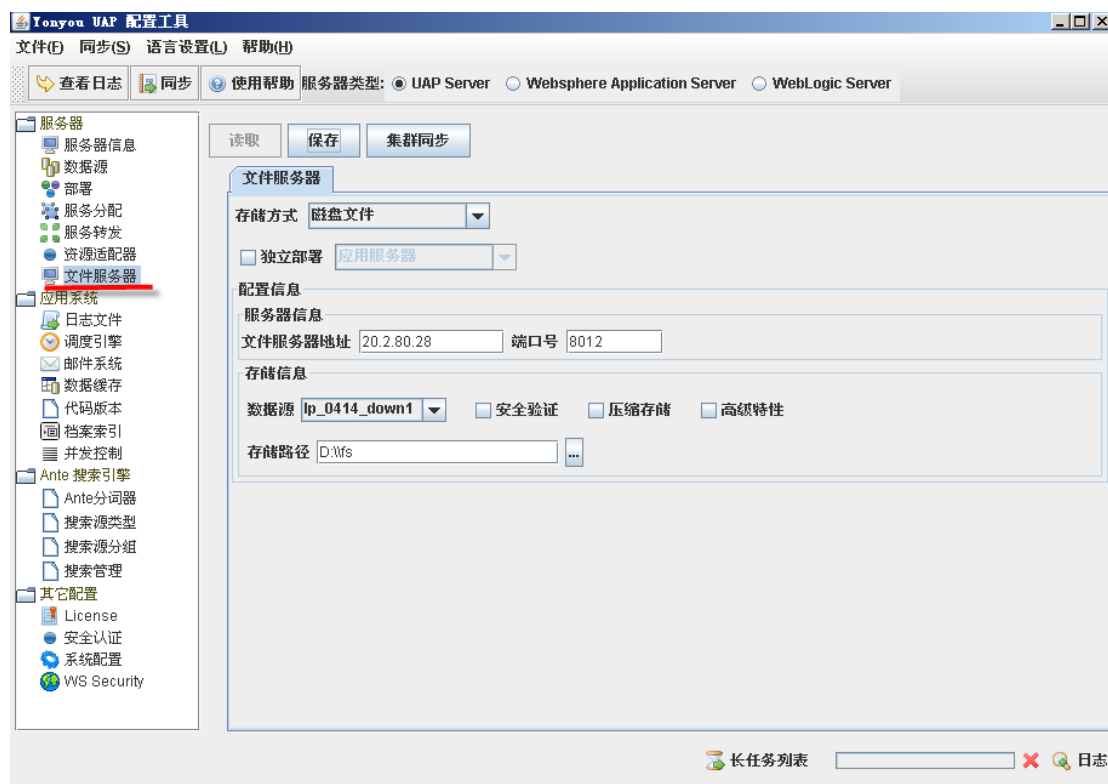
4. 产品升级

在 UAP 6.3 版本中，有两套文件存储系统，这两套存储系统相互独立。在实际的过程容易让人产生疑惑，而且也极大的增加了维护的成本。因此在 UAP 6.5 统一了文件存储方式，废弃了文档存储配置，只保持文件服务器。

在 UAP6.3 中配置界面为：



UAP6.5 中废弃文档存储配置后的配置界面为：



本文档主要演示废弃文档存储后，将旧版本（UAP 65 版本之前）中通过文档存储方式存储的附件升级到文件服务中的过程以及注意事项。升级后的文件在文件服务器中能正常使用。

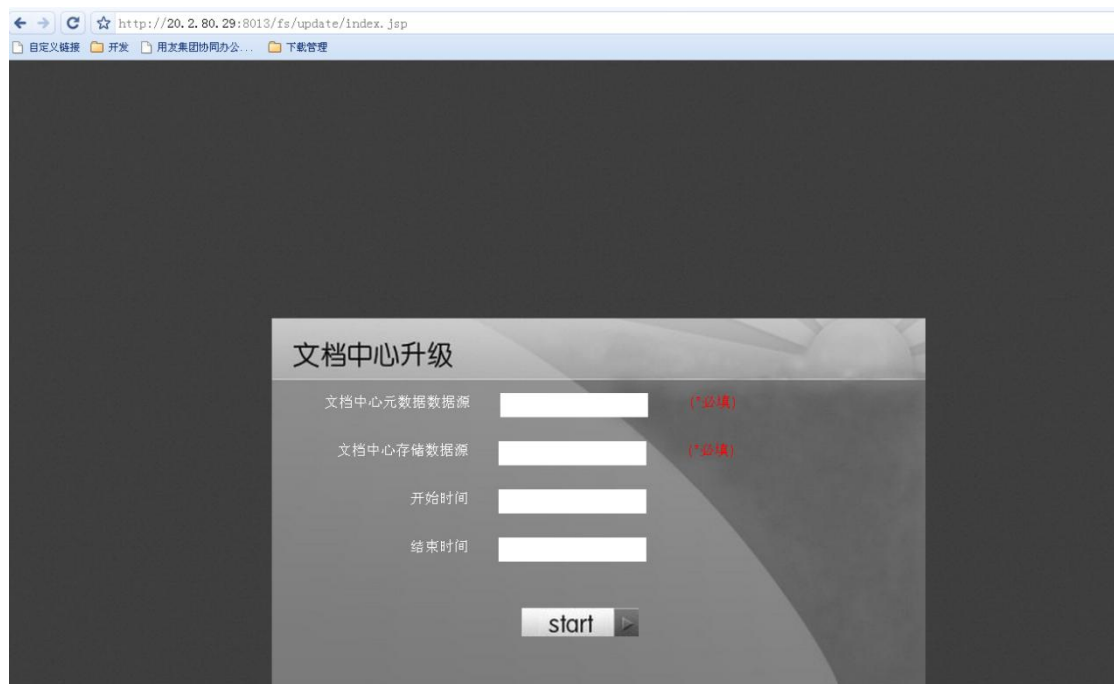
4.1. 文档中心升级

升级环境必须同旧版本中应用服务器部署在同一台机器上。

部署成功后，打开升级页面 <http://{server-host}:{port}/fs/update/index.jsp>，配置升级所需参数。

其中 server-host 为应用服务器的 ip 地址，port 为应用服务器的服务端口号。

升级页面如下：



4.2. 文档中心升级设置

1) 文档中心元数据源

文档存储中，文件的基本信息如文件的标识，文件创建者，修改者等都是保存在数据中的。这个数据源就是存储这些基本信息的数据源。

值：应是当前应用服务器的帐套数据源。

2) 文档中心存储数据源

除了上述的存放文件元数据的数据源，还有一个跟文件真实存储有关的数据源，如在数据库存储方式中，用于存储真实文件流的数据源。

值：这个数据源的值应该取自文档服务器（注：文档服务器跟应用服务器有可能不在一台机器上，所以升级之前请先确认文档服务器的位置）。

查看文档服务器下 {nchome}/ierp/sf/ documentConfig.xml 文件。

数据库存储时：此文件内容如下：

```
<?xml version="1.0" encoding="GB2312" standalone="no"?>

<fileStoreConfs>
```

```
<fileStoreConf code="code" name="name">
```

```
<fsvFactoryClsName>nc.file.fsv.db.DBFileStoreViewFactory</fsvFactoryClsName>
```

```
<param key="dbfile.dsname" value="{currThreadDsname}"/>
```

```
<param key="dbfile.tablename" value="sm_filestoreview"/>
```

```
</fileStoreConf>
```

```
</fileStoreConfs>
```

查看<param key="dbfile.dsname" value="{currThreadDsname}"/>。如果 value 值为 {currThreadDsname} 则此时存储数据源设置为当前帐套数据源，否则就设置 value 的值

本地文件存储时：

```
<?xml version="1.0" encoding="GB2312" standalone="no"?>
```

```
<fileStoreConfs>
```

```
<fileStoreConf code="code" name="name">
```

```
<fsvFactoryClsName>nc.file.fsv.local.LocalFileStoreViewFactory</fsvFactoryClsName>
```

```
<param key="root.dir.path"
```

```
value="{nc.home}/documentroot/{currThreadDsname}"/>
```

```
</fileStoreConf>
```

```
</fileStoreConfs>
```

查看<param key="root.dir.path" value="{nc.home}/documentroot/{currThreadDsname}"/>。如果 value 的中包含的就是 {currThreadDsname}，那么此时存储数据源设置为当前帐套数据源。否则就设置为 {currThreadDsname} 对应的值。如

```
<param key="root.dir.path" value="{nc.home}/documentroot/ae_fs"/>
```

此时存储数据源就设置为 ae_fs。

Ftp 文件存储

Ftp 文件存储时直接设置当前帐套数据源即可。

3) 开始时间、结束时间

由于旧版本文件存在文件数目很大，升级的数据量很大，因此在升级的过程中，可以指定需要升级的文件范围。

每一个上传的文件都会有相应的元数据存放在数据库中（如文件大小，文件标识符，文件创建时间等）。在数据库中相应的有一个 ts 字段，用于记录数据库记录的最新变更时间。插入，更新记录，ts 字段都会记录插入或者更新的时间。本工具的开始时间、结束时间指的就是 ts 字段对应的值。

开始时间、结束时间可以不设置，此时一次对所有的文件进行升级

开始时间、结束时间可以只设置一个。如只设置开始时间为 2014-01-01（**必须按照此格式**）。则只更新数据库中 ts>' 2014-01-01 00:00:00' 的记录对应的文件。若只设置结束时间为 2014-02-01（**必须按照此格式**）则只更新数据库中 ts<=' 2014-02-01 00:00:00' 的记录对应的文件。

开始时间、结束时间可以同时设置。如设置开始时间为 2014-01-01（**必须按照此格式**），设置结束时间为 2014-02-01（**必须按照此格式**），则更新数据库中 ts>' 2014-01-01 00:00:00' and ts<=' 2014-02-01 00:00:00' 的记录对应的文件。

5. 文件导入导出

文件服务器目前支持的存储方式有本地磁盘存储、分布式存储、Ftp 文件存储。为了防止因为物理存储介质故障，导致文件数据丢失或者不可用，文件服务器提供了文件导入导出的功能。本手册主要说明导入导出过程注意的事项以及相应的参数设置。

文件导出导入操作都是在文件服务器的服务端进行的。

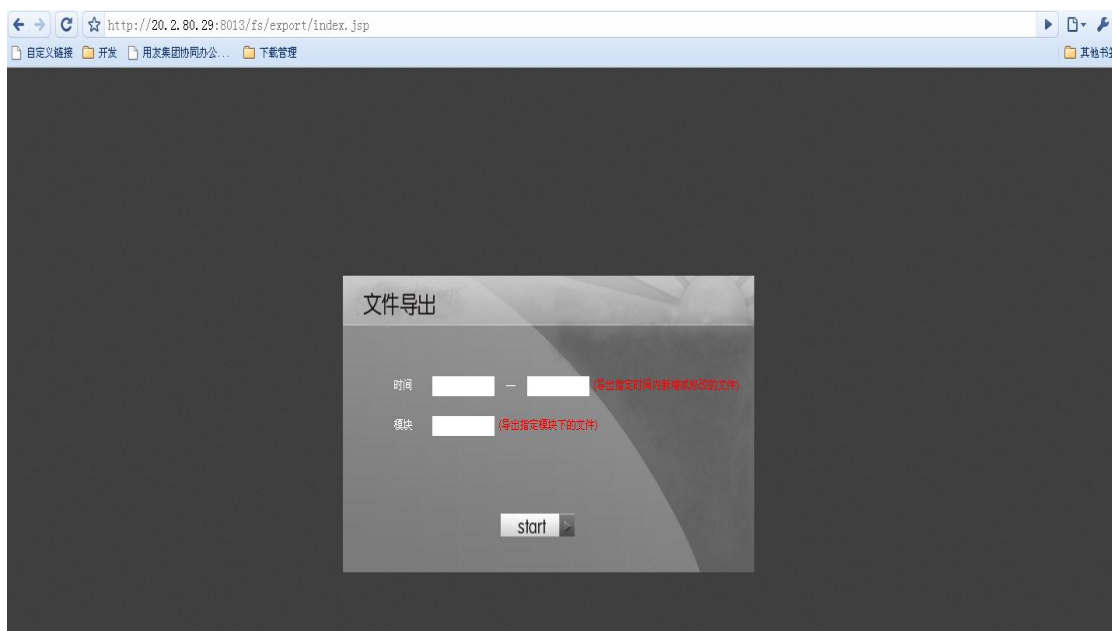
5.1. 导出设置

输入如下地址即可打开导出文件的设置页面。

http://{fs_server_host}:{fs_server_port}/fs/export/index.jsp

其中，fs_server_host 为文件服务器的 ip 地址；fs_server_port 为文件服务器的端口号。

导出页面如下：



5.2. 参数设置

文件服务器存在的文件数目比较大，总的数量也比较大。如果一次性全部导出，会对

整个系统的性能以及稳定性产生影响，因此，在导出操作时，支持按照时间段，模块的方式进行导出。

时间：

代表的是文件的最新修改时间。

在配置页面中有两个输入框，用于导出指定时间区间内的文件。我们将前一个输入框对应的时间称为开始时间，把后一个输入框对应的时间成为结束时间。

开始时间和结束时间可以都不设置，此时说明则不指定时间，导出所有文件。

可以只设置一个。如只设置开始时间为 2014-01-01 (**必须按照此格式**)。则只导出修改时间在 '2014-01-01 00:00:00' 之后的文件 (包括 2014-01-01 00:00:00)。若只设置结束时间为 2014-02-01 (**必须按照此格式**) 则导出修改时间在 '2014-02-01 00:00:00' 之前的文件 (包括 2014-02-01 00:00:00)。

开始时间、结束时间可以同时设置。如设置开始时间为 2014-01-01 (**必须按照此格式**)，设置结束时间为 2014-02-01 (**必须按照此格式**)，导出修改时间在 '2014-01-01 00:00:00' 之后 (包括 2014-01-01 00:00:00) 在 '2014-02-01 00:00:00' 之前的文件 (包括 2014-02-01 00:00:00)。

模块：

指定需要导出的模块。如果没设置则导出所有文件，否则只导出指定模块下的文件。

5.3. 存储设置

在 {nchome}/ierp/sf/filesserverconfig/imexconfig.properties 中设置导出后的文件存储路径。

exportLoc: 设置导出后的文件存储路径

logfileLoc: 设置导出过程中日志路径

5.4. 导入设置

输入如下地址即可打开导入文件的设置页面。

http://{fs_server_host}:{fs_server_port}/fs/import/index.jsp

其中, fs_server_host 为文件服务器的 ip 地址; fs_server_port 为文件服务器的端口号

相对于导出文件时需要设置参数, 导入操作不需要设置任何参数, 只需要在文件服务器后台设置需要导入的文件目录即可。

即在 {nchome}/ierp/sf/fileserverconfig/imexconfig.properties 中设置即可

importLoc: 设置需要导入的文件路径

logfileLoc: 设置导入过程中日志路径

5.5. 注意事项

导入导出操作都是文件服务器的服务端完成的, 但是 Ftp 方式中, Ftp 服务器的信息是在客户端设置的, 因此, 若想对 Ftp 方式上传的文件进行导入导出操作, 需要在服务器端设置 Ftp 服务器信息。