

## Research and Application of Java Auto-Configuration and Deployment Tool Based on XML Metadata

Wu Hailli\*, Gong Renbin, Wang Congbin

Research Institute of Petroleum Exploration & Development-Northwest, PetroChina,  
No.535 Yanerwan Road, LanZhou, GanSu 730020, China, Ph./Fax: +010-51551230-8823/51551390

\*Corresponding author, e-mail: wuhl\_xb@petrochina.com.cn

### Abstract

With the on-line and acceptance of A11's production management system, the operational work is becoming increasingly critical to managing distributed system. How to ensure system version update work fast, stable and efficient, become an urgent problem to be solved. We design and implement ConfigTool which is a auto-configuration and deployment tool based on XML metadata implemented by Java. ConfigTool realizes the automated processes of decompression, modification, compression and deployment of packages. In addition, we also study and optimize the file compression algorithm which is used in ConfigTool. Through ConfigTool, operational work can become simple and improving work efficiency.

**Keywords:** distributed system, XML, auto-configuration and deployment, compression algorithm

Copyright © 2015 Universitas Ahmad Dahlan. All rights reserved.

### 1. Introduction

With the on-line and acceptance of A11's production management system, the operational work is becoming increasingly critical to managing distributed system. How to ensure system version update work fast, stable and efficient, become an urgent problem to be solved. Because of the following reasons, we developed ConfigTool.

Firstly, the distributed system is used with tens of servers sometimes thousands in different oilfield, besides configuration files associated with system upgrading are distributed in many different files end with ".properties", so when configure these profiles manually will inevitably produce errors which are hard to find.

Secondly, relational database, real-time database and domain name address are different in different oilfield. Configuration items like these are countless. These conditions have increased the maintenance workload, and the work is needed to ensure the accuracy. From Figure 1 the relationship of A11's production management system, you can image how many configuration files and items they are!

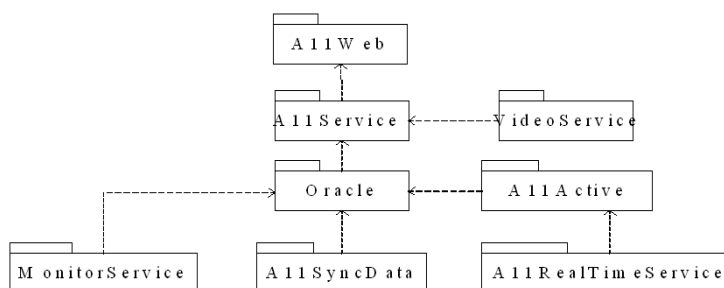


Figure 1. The relationship of A11project

Thirdly, When developers release a new version of program (war or jar), it's only for a certain oilfield. To deploy this program to another oilfield, operational people have to

decompress this war or jar, then find the right configuration files and modify the right items, compressed all the extracted files and modified files into the target war or jar packages, finally, deploy the target war/jar package to the application server of the target oilfield. In this operational work, decompression, compression and deployment are repetitive work. While in the larger sense, modify configuration files is also the repetitive work; but in the micro-scope, operational people need to understand each configuration file is what specific defined, and need to find the correct configuration files and items to modify. In this condition, it's normal to maintain one configuration files for every environment of every oilfield. It's very easy to make mistakes by doing this, and the cost and operational time will grow linearly along with the increase of deployment environments. Soitis of great practical significancetothe develop a intelligent auto-configuration and deployment tool [1].

There are many configuration tools about software development and testing. Autoconfig Toolkit of WebX which is Taobao's open-source web framework, define the variable configuration items as a template file, when running the tool can generate the specific profiles base on the corresponding template file [4]. But to use the tools needs through the command line or maven, it doesn't have a counterintuitive interface and it's complex to configure Autoconfig itself. Ansible is an automation operation management tool implemented by Python. It Contains the advantages of a number of maintenance tools (puppet, Cfengine, chef, func), but it is only applicable to Linux system. Puppet is a configuration management tools base on C/S framework. When using this tool, you have to Install Puppet master on a central server, Install the puppet client software on the target host. When the client is connected on the Puppet master, the configuration files Defined in the Puppet master are compiled, and run on the client. If use this tool, its way of communication will affect the A11's production management system and it's not applicable for our system [2-3].

Our goal is to design automation configuration and deployment tool which completely don't or need little manual work. We implement automation configuration and deployment tool based on XML metadata with Java. By using it we configure and deploy every program version of A11 easily, and effectively reduces the system operation time and the probability of making mistakes, which greatly improves the work efficiency.

## 2. Design and Implementation of Tool

Automation configuration and deployment tool (called as ConfigTool), we design its procedures as follows based on all of our research and the actual needs.

First, select the war/jar package(s) (called A) which need(s) to configure and deploy.

Second, read and parse the XML configuration template (called B), in which tag "conf" (called C) corresponds tothe name of the configuration file, subtag "conf" uses key-value pairs to express the associated configuration items. While Key (called D) is a configuration item's name in the ".properties" configuration file, value (called E) is this configuration item's value.

Third, decompress A, and find C in the decompressed files, modify D's value as E in C. Compress all the files as target war/jar package (called F).

At last, deploy F to the target application server.

### 2.1. Build and Parse Configuration Template

Configuration Templateisa XML file which contains a large number of configuration information, we need to extract the configuration files, configuration items and the target values needed to modify from the configuration template.

The XML configuration template is designed as follows:

Tag "lib" indicates this configuration template is for which oillfield or production area. Tag "templates" includes all the program packages whose configuration files needs to modify. Each program package is a "template" tag, in which the "app" property indicates the name of this program package, "type" property identifies the type of environment (development, test or production), "version" property specifies the program's version number. The "src" property of tag "conf" indicates which configuration files of the current program package need to modify. Tag "conf" indicates which configuration items and their target values of the current configuration file need to modify, in which "key" property indicates which configuration item you want to modify,

“value” property indicates the corresponding target modified value. All the tags and properties mentioned above are required.

```

<?xml version="1.0" encoding="UTF-8"?>
<xml-body>
  <lib company="oilfield name">
    <param id="" prd="" dev="" tst="" desc=""/>
    <param id="" prd="" dev="" tst="" desc=""/>
  </lib>
  <templates>
    <template app="package name" type="environment type(1 development;2 test; 3 production)" version="version No.">
      <conffile src="configuration file name">
        <conf key="configuration item name" value="configuration item's value" paramid="" desc=""/>
        <conf key="" value="" paramid="" desc=""/>
      </conffile>
    </template>
  </templates>
</xml-body>

```

Figure 2. The Configuration Template of ConfigTool

ConfigTool abstract each tag of the XML file as a class, implements the analysis of XML Configuration Template based on JDOM, uses the Properties Class to achieve the reading and modifying of the configuration file.

## 2.2. Compression-Decompression Algorithms

ConfigTool uses the Ant library of Java to complete the decompression and compression operation. In the previous algorithm, we decompress all the files of war/jar package, then according to the analysis results of the configuration template modify the configuration items of configuration files. Finally compress all the files into target war/jar package.

The optimized algorithm is mainly for the compression process, because the previous algorithm is time-consuming in the compression process. The target war/jar package is obtained by two ways. One is for the configuration files needed to modify--first decompress, then modify, last compress. The other is for all the remaining files, just copy these files in the original war/jar to the target by the byte stream. Because the number of all the remaining files is far more than that of modified configuration files, so the efficiency is greatly raised [4-6].

We made a intercomparison between two algorithms. Using the two algorithms makes little difference on the configuration time of war package. But they have large variations in configuring jar package. Because the file structure of jar is more complicated than that of war. Of course, copy is easier. By using the optimized algorithm, the compression time of a common jar packet from initial 1-2 minutes reduced to a few seconds.

## 2.3. Automatic Deployment

ConfigTool's automatic deployment function can be used separately, that is, user can not use the function introduced in lesson 2.1 and 2.2. We define a flag in ConfigTool, when user uses the previous functions, flag=1, otherwise flag=0. If flag=1, ConfigTool use Hudson to complete the automatic construction of package which is decompressed in the automatic configuration process, then deploy the constructed application to the application server. If flag=0, user should select the program package to deploy first. In this section, we use Hudson which supports the distributed construction and abstract the application server consoles. Figure 3 is one of the application server consoles' abstract interface [7-9].

Figure 4 shows the ConfigTool' whole automatic deployment process. First get the program package to deploy, then select the application server you want to deploy (eg: Tomcat/WebSphere), then use the Hudson make the automatic construction of the package, upload the constructed program to the application server, according to program's construction structure to determine whether to deploy package warehouse [10]. At the same time, apply the resource deployment and agent middleware container, then deploy the application package, after a little while, ConfigTool will return the access address and report that deployment is successful.

```

public interface WebSphereAdminService {
    AdminClient getAdminClient(Endpoint endpoint) throws ConnectorException;
    boolean isConnected();
    void connect(Endpoint endpoint);
    void installApplication(ApplicationVO applicationVO, AdminClient adminClient) throws ConnectorException, Exception;
    String startApplication(String appName, AdminClient adminClient) throws Exception;
    String stopApplication(String appName, AdminClient adminClient) throws Exception;
    boolean isInstalled(String appName) throws Exception;
    List<J2EEApplication> listApplications() throws Exception;
    J2EEApplication getApplication(String appName) throws Exception;
    ListModules listModules(String appName) throws Exception;
    List listURIs(String appName) throws Exception;
    List<Server> listServers() throws Exception;
    List<JVM> listJVMs() throws Exception;
    void uninstallApplication(String appName) throws Exception;
    String listPorts(AdminClient adminClient, String target);
}
    
```

Figure 3. The Abstract Interface of IBMWebSphere Application Server

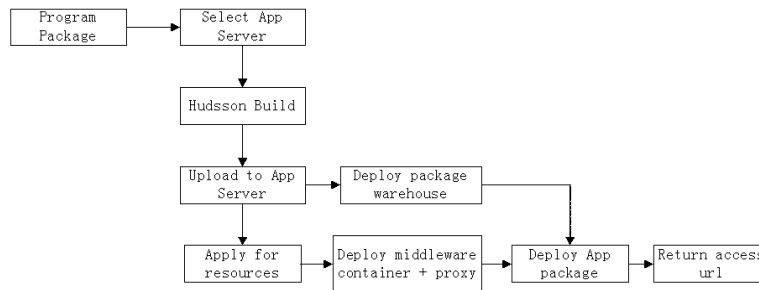


Figure 4. Flow Chart of ConfigTool's Automatic Deployment

### 3. Results and Analysis

ConfigTool is a stand-alone program on auto-configuration and deployment tool implemented by Java, regards XML file as information repositories, its core function is to automatically complete the configuration process from the original WAR/WAR package to the target ones. With the experience of the trial run in the oilfields of Petro China, using ConfigTool can smoothly manage the process of configuration and deployment of hundreds of sets of environment in dozens of areas currently. Table 1 shows the basic principle of the tool:

Table 1. Principle of ConfigTool

A. Oilfield's parameter library	B. Configuration template	C. Directory of configuration files	D. Target program package
(Every oilfield has one)	(Every oilfield has one which includes all the packages of three environments)	(corresponding to war/jar packages' directory)	(original package + C)
Library of Southwest Oilfield	Southwest's template	--A11Web's directory --A11Service's directory --A11SyncData's directory --A11RealTimeService's Other Apps' directories...	--A11Web_V1.1.war --A11Service_V1.2.war --A11 SyncData_V1.1.war --A11RealTimeServie.war Other version...
Library of Xinjiang Oilfield	Xinjiang's template	--A11Web's directory --A11Service's directory --A11SyncData's directory --A11RealTimeService's Other Apps' directories...	--A11Web_V1.1.war --A11Service_V1.2.war --A11 SyncData_V1.1.war -A11RealTimeServie.war Other version...

A represents the parameter library of oilfield, each oilfield has one library. B represents the configuration template which is corresponding to one system. Its contents involve the configuration of development, test and production environment. C represents the directory of configuration files which are exactly corresponding to that of war/jar package. D represents target program package which consists of original package and the files in C.

From left to right, ConfigTool automatically generate deployable WAR/JAR package. A and B can be edited in the XML/Excel editor. C can quickly generate the appropriate configuration file directory by ConfigTool. D program will automatically overwrite the configuration files in original package's C directory, and get the target deployment package.

ConfigTool mainly contains four panels which are "Original Package", "Configuration Parameters", "Package", and "Deploy". Figure 5 shows the "original package" panel. Select the path of program's original package in this panel, ConfigTool can automatically list all the packages in this path, then users can select the packages they want to configure and deploy.



Figure 5. ConfigTool's "Original Package" panel

Figure 6 shows the "Configuration Parameters" panel. In this panel, select the path of configuration templates and the environment that the target package to deploy, ConfigTool can parse all the configuration templates in that path, and list the results. And according to the environment ConfigTool can generate the corresponding package. In "Package" panel, click "package" button, ConfigTool first decompress the original packages, and then parse the configuration templates, according to the analytical results modify the corresponding configuration items in configuration files, finally compress all the original and modified files to the target packages. To ensure safety, some private information have been covered in Figure 6.



Figure 6. ConfigTool's "Configuration Parameters" panel

Figure 7 shows the "Package" panel. In "Package" panel, click "start package" button, ConfigTool first decompress the original packages, and then parse the configuration templates, according to the analytical results modify the corresponding configuration items in configuration files, finally compress all the original and modified files to the target packages.

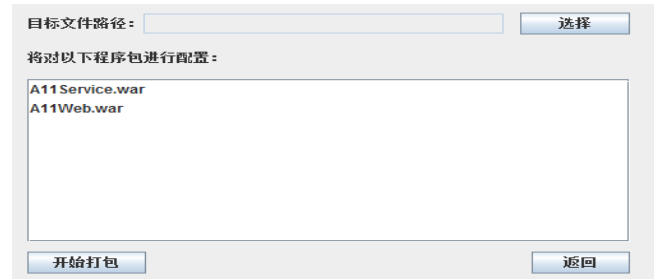


Figure 7. ConfigTool's "Package" panel

Figure 8 shows the "Deploy" panel. In this panel, select the type of application server, Fill in the server IP, user name, password, and the deployment path, ConfigTool will use Hudson to automatically build the target package and upload it, and then call the appropriate application server's abstract class to complete the automatic deployment of the target package.



Figure 8. ConfigTool's "Deploy" panel

Using ConfigTool can solve the operation and maintenance problems, for the A11 project, simplify the configuration and deployment work of A11 system in all oilfields of PrtroChina, effectively reduce the error rate and configuration-deployment time, greatly improve the work efficiency, and the tool is easy to use. Table 10 shows the effect of contrast before and after the application of ConfigTool.

Table 2. The effect of contrast before and after the application of ConfigTool for A11 project

	Configuration and Development Time	No. of Operation and Maintenance Staff	Training Time	Error Rate
before	1-2day/version	5-10	1h/person	70%
After	15min/version	1-2	0	10%

For A11 project, before the application of ConfigTool, every version of every oilfield's program (includes 4 war packages and 10 jar packages) needs to decompress, modify configuration files, compress and then deploy to the server, in the case of zero error rate, this process will take 1-2 days for each version, if an error occurs, this will last longer, usually about

one week. While using ConfigTool only spends 15 minutes with nearly zero error rate. The number of operation and maintenance personnel from 1-2 person(s) from each oilfield reduces to 1-2 person(s) for all the oilfields. After using ConfigTool, not only reduce the error rate, but also don't need training time. Because operation and maintenance personnel are not required to understand the relationship and meaning of each package and configuration file. More important, ConfigTool applies to the configuration and deployment of all J2EE projects, it has broad application prospects, users have to do is just writing a configuration template for each package in accordance with the format which introduced earlier.

#### 4. Conclusion and Further Study

From all our work we can see, XML configuration template contains a large number of project's configuration information. Using ConfigTool can effectively reduce the configuration and deployment time of maintenance staff, also help reduce the error rate by avoiding manually change the configuration file, and improve the work efficiency. To fix the time-consuming challenge for the compression process, we described a highly-available algorithm. The improvement of the decompression process's algorithm is more practical than ever, and make ConfigTool's interface more user-friendly is also important.

#### References

- [1] Guo An. *Study of Software Configuration Management Applied in Project Development*. 6<sup>th</sup> National Conference on Computer Application. Beijing, China. 2002; 3: 1082-1084.
- [2] Dianxi Shi, Bo Ding, Wei Cui, Yiming Zhang, Huaimin Wang. Research and Implementation of Deployment and Configuration Tool for Light-weight Components. *Computer Science*. 2011; 06: 165-168.
- [3] Zhang Yiming, Shi Dianxi, Ding Bo, Yin Gang. *Research and Implementation of Deployment and Configuration Tools for Pervasive Computing Middleware*. 4<sup>th</sup> Jiont Conference on Harmonious Human Machine Environment. Wuhan, China. 2008; 1: 574-580.
- [4] Pham Duc-Son, Venkatesh Svetha. Efficient Algorithms for Robust Recovery of Images from Compressed Data. *IEEE Transactions on Image Processing*. 2013; 41(6): 2212-2221.
- [5] BL Kess, DR Steinwand, SE Reichenbach. Compression of the Global Land 1-km AVHRR Dataset. *International Journal of Remote Sensing*. 1996; 39(2): 1715-1723.
- [6] Hach Faraz, Numanagic Ibrahim, Alkan Can, Sahinalp S Cenk. SCALCE: Boosting Sequence Compression Algorithms Using Locally Consistent Encoding. *Oxford Journal of Science and Mathematics*. 2012; 21(4): 2823-2830.
- [7] Lin Bo, Zhou Minghui, Liu Tiancheng, Huang Gang, Mei Hong. A Web Container Integration Framework in J2EE Application Servers. *Journal of Softwar*. 2006; 17(5): 1195-1203.
- [8] Li Bingpeng, Sun Changqi, Lan Yang. Method of Software Intelligent Deployment. *Journal of Computer Applications*. 2009; 01: 248-252.
- [9] Zhang Yuan, Zhang Yunyong, Fang Bingyi. An Automated Deployment Architecture of Cloud System Based on Workflow. *Telecommunications Science*. 2014; 11: 14-21.
- [10] Zhao Tianqi. Automatic Deployment and Resources Allocation of Web Application in Cloud Computing Environment. Harbin Institute of Technology. 2014.