Concepts for Developing a Fully Flexibly UI Control Matrix for Remote Data Capture

Sunil G. Singh, Jason Essig and Thomas Struzik

DBMS Consulting

13 October 2010

Remote Data Capture

Session 30

Acknowledgements

- Many thanks to the OHSUG for this opportunity to present to the OHSUG RDC Focus Group.
- Many thanks to the OHSUG RDC Focus Group Chairs for their infinite patience in receiving and expeditious review of this presentation.
- Many thanks to everyone who participated in the development of this presentation.

Assumptions

- Audience has a working knowledge of current OC/RDC 4.6 internals/architecture.
- Conceptual presentation only. Not any statement of future direction/product direction of Oracle.

Problem Statement: Many Places to Configure RDC Behavior

- Knowing where to go to make a change to something in RDC can be challenging.
  - RDC Configurator (change by user, study, <USER GROUP> or any combination thereof).
  - Reference Codelists for Discrepancy Management (DISCREPANCY ACTIONS/STATUS <USER GROUP>).
  - RDC Study/Site Administration (access to Study/Sites, Update, Update Disc, Verify, Approve).
  - OCL => Admin => DCI Form Local Database Settings (also at Study level).
  - WEB.XML (Timeouts, Report Server URL).
  - OpaConfig.do (Password for TMS$BROWSER, OC4J_INTERNAL, Server, Port, ORACLE_SID).

Problem Statement: Many Places to Configure RDC Behavior (2)

- Some changes are not possible from these access points, but can be done (disable a toolbar button).

Can Administration be Centralized: Learning from EBS

- In the Oracle E-Business Suite world, there was a similar issue. There were configuration files and settings for the system in many places and it was difficult to manage.
- In 11.5.6 and higher, Oracle release adconfig or autoconfig. This was a single tool (set of scripts) designed to update all of the system-level configurations centrally.
  - These configurations were also consolidated into 2 .xml files, which are not supposed to be manually edited.
- Furthermore, this autoconfig module is accessible from the OEM console (ias_admin login page).
- External Link to autoconfig.

Consolidating Configurations for RDC in .XML? Is it possible?

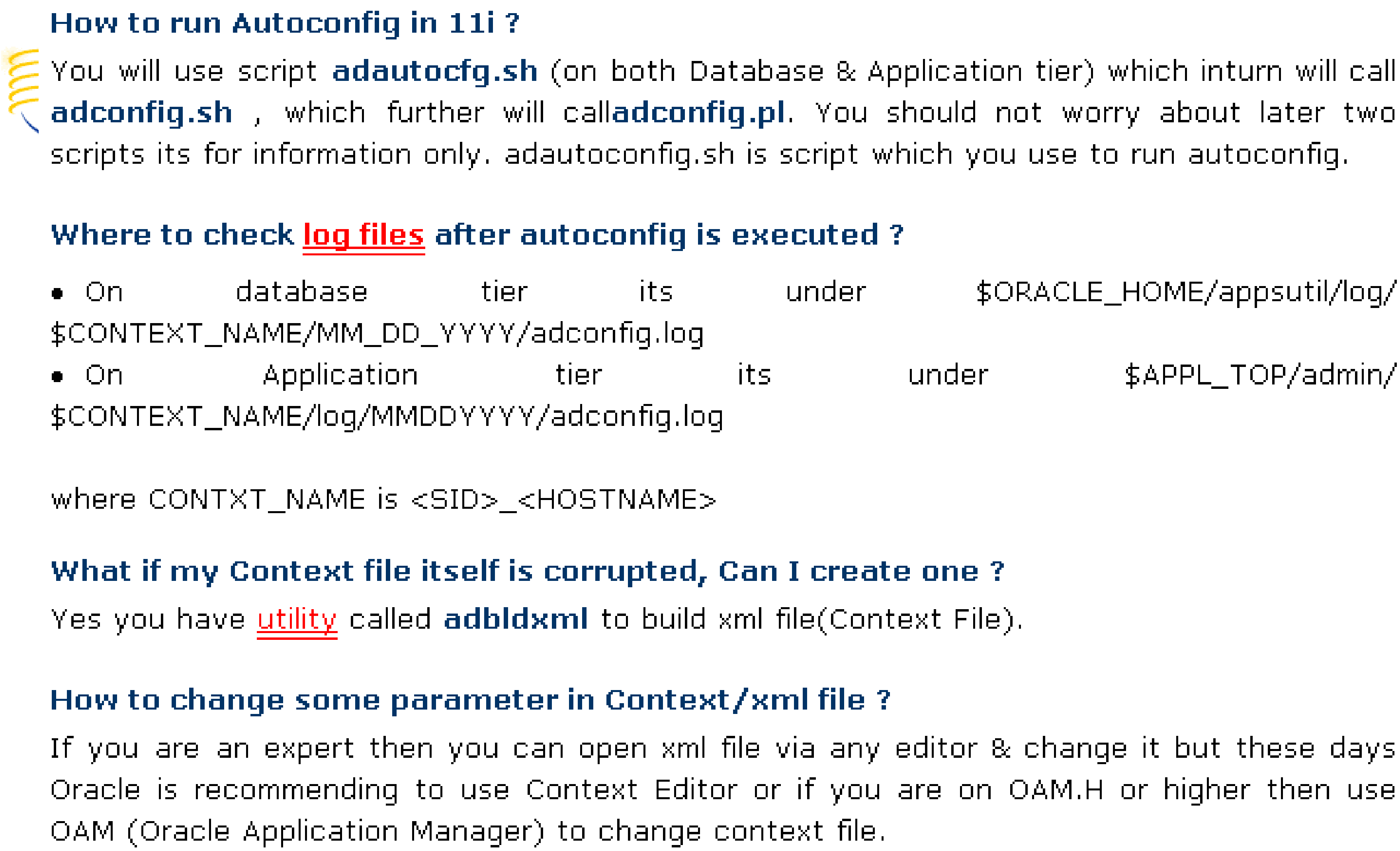
- Already happening with web.xml and OpaConfig.xml.
- There are several .xml files which are deployed by default with an Oracle J2EE application.
  - Default files are modifiable in OEM, IF you know where to look.
  - RDC-specific files are NOT modifiable this way (thus the need for a separate OpaConfig.do application).
- Reference codelists are not .xml dependent.
- Study/Site administration depends on table entries.

Creating an auto-config Like Application

- The EBS approach is a very useful approach for managing complex configurations.
- If OpaConfig.do is already a mini-application, why not "push" ALL relevant RDC configuration values into a central XML file?
- OpaConfig could then:
  - Modify the central XML file.
  - Update the database and reference codelists in the backend, using APIs.
  - Maintain transactional integrity.
  - Create automatic backups of the previous configuration.

Creating an auto-config Like Application (2)

- This could also be registered in dcmctl, so there would be database file backups for the last known configuration.
  - Could be more complex to handle changes, but behavior would be consistent.
  - External link is possible.



How to run Autoconfig in 11i ?

You will use script **adautocfg.sh** (on both Database & Application tier) which inturn will call **adconfig.sh** , which further will call**adconfig.pl**. You should not worry about later two scripts its for information only. adautoconfig.sh is script which you use to run autoconfig.

Where to check log files after autoconfig is executed ?

- On database tier its under $ORACLE_HOME/appsutil/log/ $CONTEXT_NAME/MM_DD_YYYY/adconfig.log
- On Application tier its under $APPL_TOP/admin/ $CONTEXT_NAME/log/MMDDYYYY/adconfig.log

where CONTXT_NAME is <SID>_<HOSTNAME>

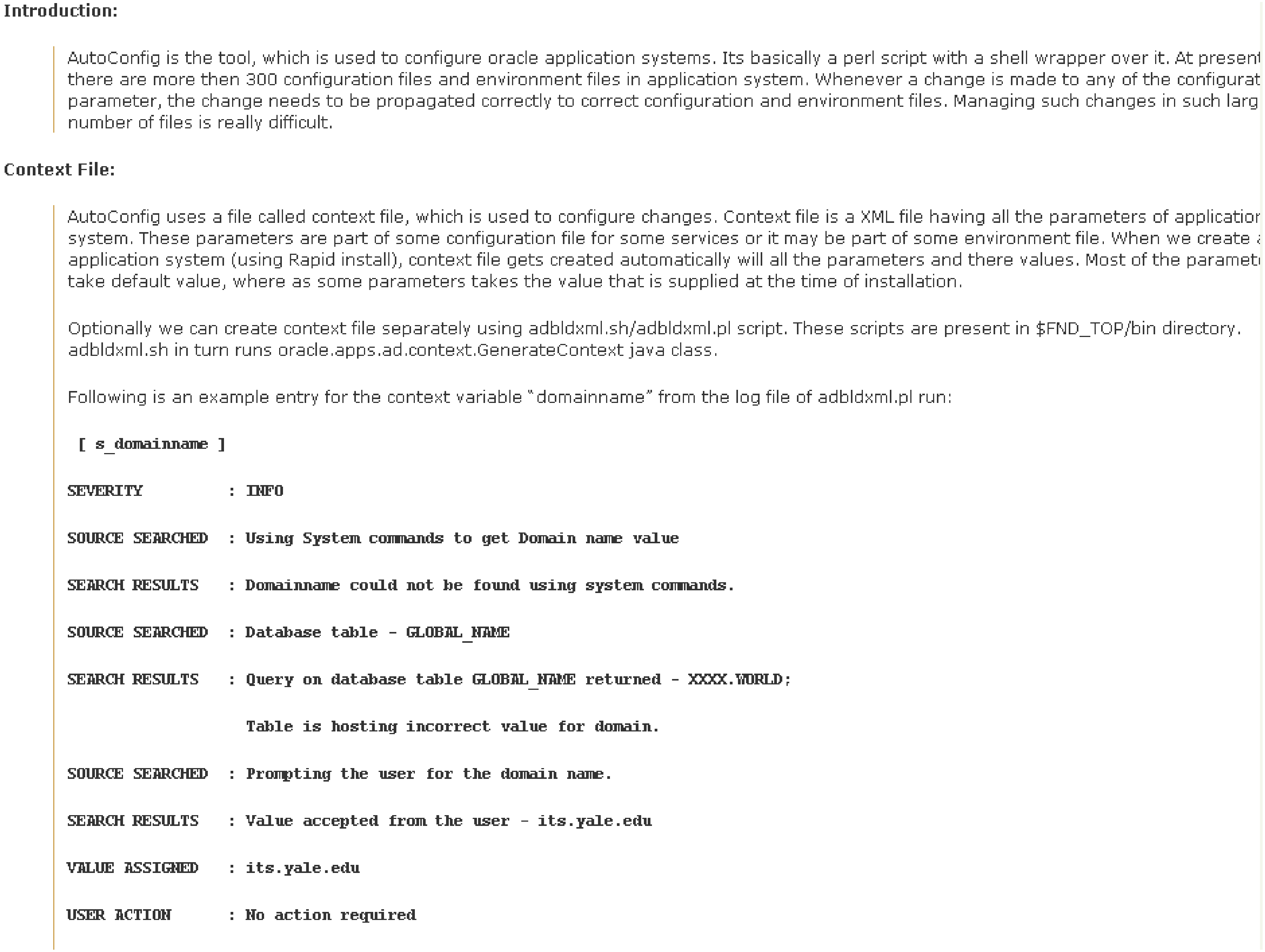
What if my Context file itself is corrupted, Can I create one ?

Yes you have utility called **adbldxml** to build xml file(Context File).

How to change some parameter in Context/xml file ?

If you are an expert then you can open xml file via any editor & change it but these days Oracle is recommending to use Context Editor or if you are on OAM.H or higher then use OAM (Oracle Application Manager) to change context file.

- http://www.teachmeoracle.com/autoconfig02.html



Introduction:

AutoConfig is the tool, which is used to configure oracle application systems. Its basically a perl script with a shell wrapper over it. At present there are more then 300 configuration files and environment files in application system. Whenever a change is made to any of the configurat parameter, the change needs to be propagated correctly to correct configuration and environment files. Managing such changes in such larg number of files is really difficult.

Context File:

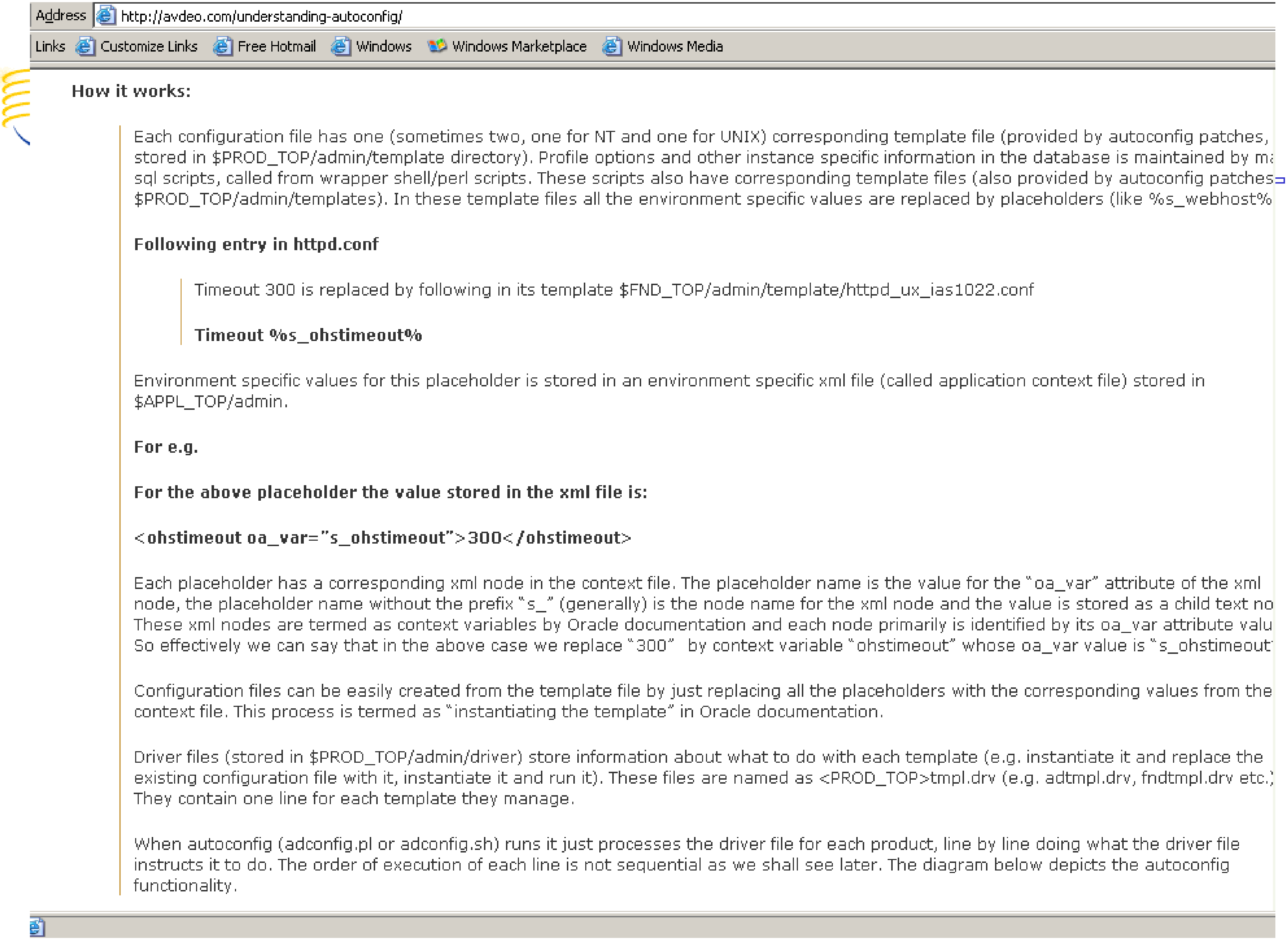
AutoConfig uses a file called context file, which is used to configure changes. Context file is a XML file having all the parameters of applicatior system. These parameters are part of some configuration file for some services or it may be part of some environment file. When we create a application system (using Rapid install), context file gets created automatically will all the parameters and there values. Most of the paramete take default value, where as some parameters takes the value that is supplied at the time of installation.

Optionally we can create context file separately using adbldxml.sh/adbldxml.pl script. These scripts are present in $FND_TOP/bin directory. adbldxml.sh in turn runs oracle.apps.ad.context.GenerateContext java class.

Following is an example entry for the context variable "domainname" from the log file of adbldxml.pl run:

```
 [ s_domainname ]

SEVERITY          : INFO

SOURCE SEARCHED   : Using System commands to get Domain name value

SEARCH RESULTS    : Domainname could not be found using system commands.

SOURCE SEARCHED   : Database table - GLOBAL_NAME

SEARCH RESULTS    : Query on database table GLOBAL_NAME returned - XXXX.WORLD;

                    Table is hosting incorrect value for domain.

SOURCE SEARCHED   : Prompting the user for the domain name.

SEARCH RESULTS    : Value accepted from the user - its.yale.edu

VALUE ASSIGNED    : its.yale.edu

USER ACTION       : No action required
```

- http://avdeo.com/understanding-autoconfig/


Address http://avdeo.com/understanding-autoconfig/
Links Customize Links Free Hotmail Windows Windows Marketplace Windows Media

How it works:

Each configuration file has one (sometimes two, one for NT and one for UNIX) corresponding template file (provided by autoconfig patches, stored in $PROD_TOP/admin/template directory). Profile options and other instance specific information in the database is maintained by ma sql scripts, called from wrapper shell/perl scripts. These scripts also have corresponding template files (also provided by autoconfig patches $PROD_TOP/admin/templates). In these template files all the environment specific values are replaced by placeholders (like %s_webhost%

Following entry in httpd.conf

> Timeout 300 is replaced by following in its template $FND_TOP/admin/template/httpd_ux_ias1022.conf
>
> **Timeout %s_ohstimeout%**

Environment specific values for this placeholder is stored in an environment specific xml file (called application context file) stored in $APPL_TOP/admin.

For e.g.

For the above placeholder the value stored in the xml file is:

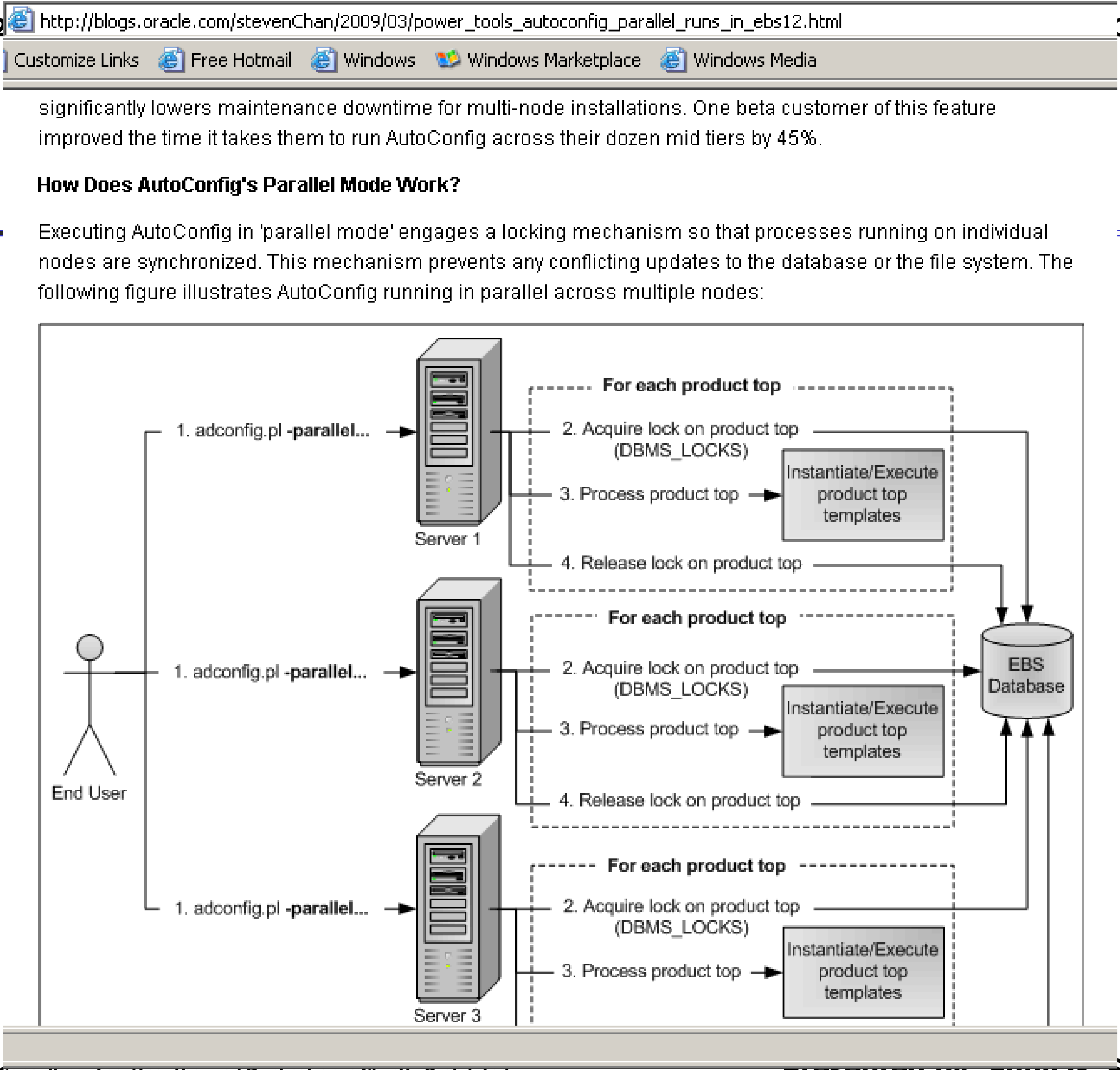
<ohstimeout oa_var="s_ohstimeout">300</ohstimeout>

Each placeholder has a corresponding xml node in the context file. The placeholder name is the value for the "oa_var" attribute of the xml node, the placeholder name without the prefix "s_" (generally) is the node name for the xml node and the value is stored as a child text no These xml nodes are termed as context variables by Oracle documentation and each node primarily is identified by its oa_var attribute valu So effectively we can say that in the above case we replace "300" by context variable "ohstimeout" whose oa_var value is "s_ohstimeout

Configuration files can be easily created from the template file by just replacing all the placeholders with the corresponding values from the context file. This process is termed as "instantiating the template" in Oracle documentation.

Driver files (stored in $PROD_TOP/admin/driver) store information about what to do with each template (e.g. instantiate it and replace the existing configuration file with it, instantiate it and run it). These files are named as <PROD_TOP>tmpl.drv (e.g. adtmpl.drv, fndtmpl.drv etc.) They contain one line for each template they manage.

When autoconfig (adconfig.pl or adconfig.sh) runs it just processes the driver file for each product, line by line doing what the driver file instructs it to do. The order of execution of each line is not sequential as we shall see later. The diagram below depicts the autoconfig functionality.



http://blogs.oracle.com/stevenChan/2009/03/power_tools_autoconfig_parallel_runs_in_ebs12.html
Customize Links
Free Hotmail
Windows
Windows Marketplace
Windows Media

significantly lowers maintenance downtime for multi-node installations. One beta customer of this feature improved the time it takes them to run AutoConfig across their dozen mid tiers by 45%.

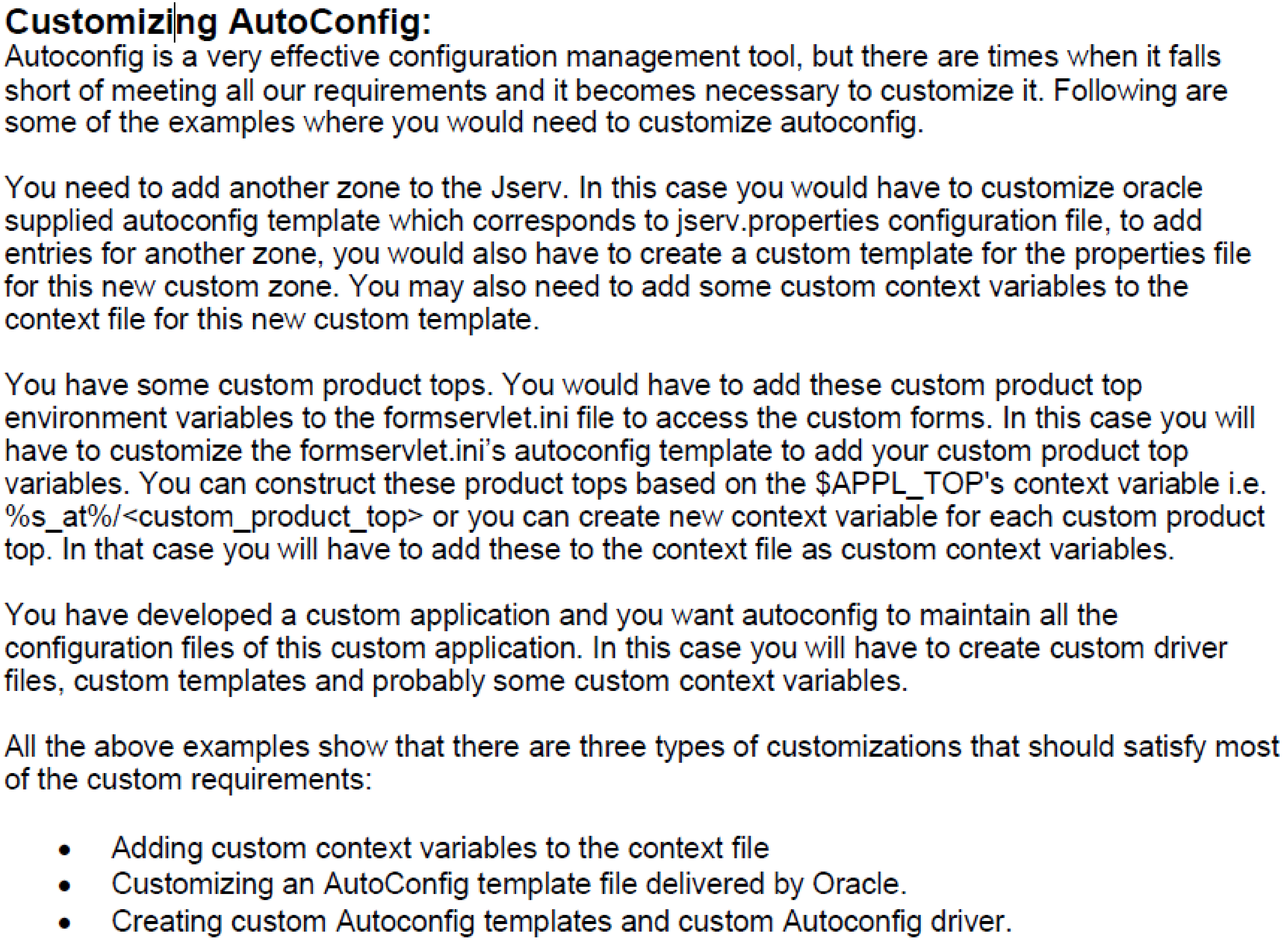
How Does AutoConfig's Parallel Mode Work?

Executing AutoConfig in 'parallel mode' engages a locking mechanism so that processes running on individual nodes are synchronized. This mechanism prevents any conflicting updates to the database or the file system. The following figure illustrates AutoConfig running in parallel across multiple nodes:

End User
1. adconfig.pl -parallel...
Server 1
For each product top
2. Acquire lock on product top (DBMS_LOCKS)
3. Process product top
Instantiate/Execute product top templates
4. Release lock on product top
1. adconfig.pl -parallel...
Server 2
For each product top
2. Acquire lock on product top (DBMS_LOCKS)
3. Process product top
Instantiate/Execute product top templates
4. Release lock on product top
EBS Database
1. adconfig.pl -parallel...
Server 3
For each product top
2. Acquire lock on product top (DBMS_LOCKS)
3. Process product top
Instantiate/Execute product top templates

What About Custom Configurations?

- To really control certain aspects of RDC, the master .js file has to be examined.
- Not much documentation on contents, but the business rules for the behavior of many components of RDC are stored here.
- This file exists in the .ear (Enterprise Archive) and .war (Web Archive) as well as in the J2EE filesystem.
    - Updated with every RDC patch which re-deploys the servlet.
- Complex structure.



Customizing AutoConfig:

Autoconfig is a very effective configuration management tool, but there are times when it falls short of meeting all our requirements and it becomes necessary to customize it. Following are some of the examples where you would need to customize autoconfig.

You need to add another zone to the Jserv. In this case you would have to customize oracle supplied autoconfig template which corresponds to jserv.properties configuration file, to add entries for another zone, you would also have to create a custom template for the properties file for this new custom zone. You may also need to add some custom context variables to the context file for this new custom template.

You have some custom product tops. You would have to add these custom product top environment variables to the formservlet.ini file to access the custom forms. In this case you will have to customize the formservlet.ini's autoconfig template to add your custom product top variables. You can construct these product tops based on the $APPL_TOP's context variable i.e. %s_at%/<custom_product_top> or you can create new context variable for each custom product top. In that case you will have to add these to the context file as custom context variables.

You have developed a custom application and you want autoconfig to maintain all the configuration files of this custom application. In this case you will have to create custom driver files, custom templates and probably some custom context variables.

All the above examples show that there are three types of customizations that should satisfy most of the custom requirements:

- Adding custom context variables to the context file
- Customizing an AutoConfig template file delivered by Oracle.
- Creating custom Autoconfig templates and custom Autoconfig driver.

- http://www.oraclegeek.net/downloads/autoconfig_whitepaper.pdf

Changing Master .js File Could Be Done By an auto-config Like Application Too!

- Since the changes to the master .js file are complex AND also require updating the .ear/.war as well as the .js file itself, this should not be done manually.
- Possible to change in JDeveloper, but not every administrator of RDC also wants to be a Java Developer.

Changing Master .js File Could Be Done By an auto-config Like Application Too! (2)

- Instead, use the same auto-config application to perform the modification of the .js and then redeploy!
  - Isolate specific logical components, such as enabling and disabling controls, in a UI which then modifies the corresponding .js component.
  - Incorporate checks to ensure syntax is correct, and also business rules are valid.

Use of XML for Master Configurations: API/SOA Calls for Configurations

- If an auto-config like application tool exists which synchronizes database changes and file changes with a master XML file, this file could also be updated through APIs/SOA calls.
- This could allow changes to Study Site configuration to be made by a SOA application which would be part of the auto-config like application tool.
- This could allow real-time workflow processing of account administration requests for RDC.

Feasibility With Future RDC Releases?

- Since there exists already:
  - An OpaConfig tool.
  - AutoConfig tool for EBS.
  - Development for Fusion platform in the future It may not be so unlikely to have a tool like this for centralized administration created as a patch release.
- Considerations for centrally administering other products, such as PhaseForward or Siebel, could lead to a similar solution but using alternate technologies.
  - Result is still beneficial to the greater RDC user community.

Biographies

Sunil G. Singh, President & CEO, DBMS Consulting, Inc.

- Sunil is a Global Oracle Health Sciences deployment expert for DBMS Consulting. He has been an active member of the OHSUG community since 1996 and is extremely grateful for this opportunity to makes these presentations at OHSUG 2010.

Jason Essig, Senior Siebel Architect Expert, DBMS Consulting, Inc.

- Jason is a Senior Siebel Architect Expert for DBMS Consulting, with 10+ years of Siebel configuration and architecture experience.

Biographies (2)

Thomas Struzik , DBMS Consulting, Inc.

- Tom Struzik is a Senior Java and .Net developer and Java development team leader for several years, with an extensive background in the financial sector. He has developed extensive customizations for DBMS Consulting including multi-lingual RDC deployments, a customized SMQ application, and support for mobile deployments of RDC.

Contact Information

Sunil G. Singh

DBMS Consulting, Inc

singh@clinicalserver.com

+1-860-983-5848