

Oracle® E-Business Suite Concepts

Technical Configuration Tools

Introduction

Configuring a new installation of Oracle E-Business Suite includes a number of stages:

- Collection of information required to create the desired configuration
- Storage of configuration information in the correct locations on the relevant machines
- Creation of technology stack configuration files with the appropriate details
- Creation of Applications configuration files with the appropriate details
- Starting of all required processes in the correct order

AutoConfig is a tool that simplifies and standardizes configuration management tasks in an Oracle E-Business Suite environment. A fresh install of Release 12.2 includes AutoConfig as a standard (and required) configuration management tool. AutoConfig can also be used with earlier releases of Oracle E-Business Suite.

The Applications Context

Where the *System Identifier* (SID) was traditionally used to identify a file as belonging to a particular Oracle E-Business Suite environment, an *Applications context* is used in an AutoConfig-managed environment such as Oracle E-Business Suite Release 12.2. The default *context name*, also referred to in this chapter as `<CONTEXT_NAME>`, is `<SID>_<hostname>`.

Using an Applications context has a number of advantages:

- Simplifies overall management of an Oracle E-Business Suite system
- Allows easier startup and shutdown of applications services
- Permits services to be installed or deinstalled independently of others
- Integrates seamlessly with a shared application tier file system (see Chapter 9)
- Enables use of Oracle Applications Manager for configuration management (see Chapter 7)
- Facilitates support for Oracle Real Application Clusters (see Chapter 4)

Before AutoConfig and the Applications context were introduced, configuration management tasks could be time-consuming and prone to error, in some cases requiring manual changes to be made to several configuration files. While individual configuration files are still used in an AutoConfig-enabled environment, they play a secondary role to an XML-based repository of Oracle E-Business Suite environment information, called the *context file*.

By centralizing the configuration information, AutoConfig simplifies procedures for activities that range from upgrading a technology stack component to starting and stopping applications services. Another benefit is that the various files AutoConfig employs can be updated via standard Oracle E-Business Suite patches.

There are separate context files for the application and database tiers of an Oracle E-Business Suite system.

Applications Context File - The Applications context file, `$INST_TOP/app/admin/<CONTEXT_NAME>.xml`, is a repository for environment-specific details used by AutoConfig to configure the application tier. Information from this file is used to generate Oracle E-Business Suite configuration files and update relevant database profiles.

Information stored includes:

- Name and location of the database
- Port numbers for Forms and Web services
- Product-specific port numbers
- Information about application tier services controlled by AutoConfig

The values of the *context variables* that make up the context file are in part determined by the choices you make when you run Rapid Install. For example, when you specify that a particular application tier node is to be used as a concurrent processing server, the relevant variable will be set to execute the scripts that start the service.

The following fragment illustrates the structure, format, and content of the Applications context file:

```
<!-- Environment on this Node --> <oa_environments< <adconfig< <adconfig_file oa_var="s_adconfig_file"/>u01/R122_EBS/EBSapps/appl/admin/adconfig.txt</adconfig_file> <APPL_TOP oa_var="s_at_adconfig"/>u01/R122_EBS/EBSapps/appl/APPL_TOP> <APPL_TOP_CSET oa_var="s_at_cset">AL32UTF8</APPL_TOP_CSET> <APPS_ENV_NAME oa_var="s_appsEnvName">VISION</APPS_ENV_NAME> <TIER_ADADMIN oa_var="s_isAdAdmin">YES</TIER_ADADMIN> <TIER_ADWEB oa_var="s_isAdWeb">YES</TIER_ADWEB> <TIER_ADFORMS oa_var="s_isAdForms">YES</TIER_ADFORMS> <TIER_ADNODE oa_var="s_isAdConc">YES</TIER_ADNODE> <APPL_TOP_NAME oa_var="s_atName">testsys100</APPL_TOP_NAME> <TIER_ADFORMSDEV oa_var="s_isAdFormsDev">YES</TIER_ADFORMSDEV> <TIER_ADNODEDEV oa_var="s_isAdConcDev">YES</TIER_ADNODEDEV> <TIER_ADWEBDEV oa_var="s_isAdWebDev">YES</TIER_ADWEBDEV> <STAGING_DIRECTORY oa_var="s_staging_directory" />u01/R122_EBS/inst/apps/vison_test100/appltmp/updates</STAGING_DIRECTORY> </adconfig< <oa_environment type="generic_service"> <DISPLAY oa_var="s_display" osd="unix">LOCALHOST:5.0</DISPLAY> <java_aws_headless oa_var="s_java_aws_headless">TRUE</java_aws_headless> </oa_environment> <oa_environment type="rapid_install"> <APPS_BASE oa_var="s_base"/>u01/R122_EBS</APPS_BASE> <PATCH_BASE oa_var="s_patch_base"/>u02/R122_EBS</PATCH_BASE> <INST_BASE oa_var="s_inst_base"/>u01/R122_EBS</INST_BASE> <HTML_TOP oa_var="s_html"/>u01/R122_EBS/FMW_Home/Oracle_EBS-app1/applications/oaocore/html</HTML_TOP> <JDK_TOP oa_var="s_jdktop" osd="LINUX_X86-64"/>u01/R122_EBS/FMW_Home/jrockit_160_22_D1.1.1-3</JDK_TOP> <JRE_TOP oa_var="s_jre" />u01/R122_EBS/FMW_Home/jrockit_160_22_D1.1.1-3/jre</JRE_TOP>
```

Note: The `s_base` and `s_inst_base` context variables were introduced in Release 12.2 to support Oracle WebLogic Server, and, like other context variables, should not be edited manually.

Database Context File - The database context file, `<s_base>/11.2.0/appsutil/<CONTEXT_NAME>.xml`, performs an equivalent role on the database tier. Information from this file is used to generate configuration files used on the database tier when AutoConfig is next run.

AutoConfig Scripts and Directories

A number of key configuration and control scripts are employed in an AutoConfig environment. AutoConfig creates several directories for these scripts and their associated files.

AutoConfig Scripts

Key AutoConfig configuration scripts on UNIX (command files on Windows) include:

- **adautoconfig.sh** - Wrapper script that passes the name of the specific environment context file to `adconfig.sh`. Located in `$INST_TOP/admin/scripts` on the application tier, and `<s_base>/11.2.0/appsutil/scripts/<CONTEXT_NAME>` on the database tier.

• **adconfig.sh** - Invoked by **adautoconfig.sh**, this script is a wrapper for **adconfig.pl**. Located in `$AD_TOP/bin` on the application tier, and `<s_base>/11.2.0/appsutil/bin` on the database tier.

• **adconfig.pl** - Invoked by **adconfig.sh**, this Perl script calls the Java API to carry out the actual configuration tasks. Located in `$AD_TOP/bin` on the application tier, and `<s_base>/11.2.0/appsutil/bin` on the database tier.

In addition to these configuration scripts, several additional AutoConfig scripts are used to help manage the Oracle E-Business Suite system. These are described later in this chapter, under *Management Tasks*.

AutoConfig Directories

Several directories are created by AutoConfig, as shown in the following table.

AutoConfig Directories	
Directory Name	Directory Contents
<code>\$INST_TOP/admin/install</code>	Install scripts
<code>\$INST_TOP/admin/scripts</code>	Control scripts
<code>\$INST_TOP/admin/log</code>	Log files

AutoConfig Operation

As AutoConfig is used for a wide range of system configuration activities, from installation to maintenance, the following discussion of its operations is divided into several sections.

Context Value Management

Context Value Management (CVM) is an AutoConfig component that is used to manage the values of variables in the context file, and automate required updates to it. CVM supports updates to both the application tier and database tier context files.

CVM actions include:

- Adding new variables to a context file
- Updating values of variables in an existing context file
- Applying new versions of context file templates
- Executing scripts or configuration tools that must complete before the AutoConfig engine starts, for example when generating the `tnsnames.ora` file

CVM can execute scripts or other tools to manipulate any required file on the file system, and allow the appropriate settings to be propagated as needed to both the file system and database. For example, it is possible to update values in the context file which will then be propagated to the file system.

AutoConfig does not undertake all aspects of configuration management, such as changes at operating system level that may have implications outside the context of Oracle E-Business Suite. In addition, it does not manage technology components that have their own native management tools: typically, these are components that are not specific to Oracle E-Business Suite, such as Oracle WebLogic Server.

Like the core AutoConfig components, CVM utilizes configuration files on both the application and database tiers, as shown in the following tables.

Application Tier CVM Files	
File Location	Description
<code>\$AD_TOP/bin/adcvms.sh</code>	Main CVM script
<code>\$AD_TOP/admin/template/adcvmat.xml</code>	Stores CVM-related data for the application tier

Database Tier CVM Files	
File Location	Description
<code><s_base>/11.2.0/appsutil/bin/adcvms.sh</code>	Main CVM script
<code><s_base>/11.2.0/appsutil/template/adcvmdb.xml</code>	Stores CVM-related data for the database tier

AutoConfig Files

As well as the context files and configuration scripts that have already been described, AutoConfig uses several other types of file in its configuration management activities. These may themselves be divided into different categories.

Template Files

AutoConfig template files are used as the starting point for creating site-specific configuration files. AutoConfig evaluates the context variables in a template file, determines the actual values required, and creates a configuration file with these values substituted. This process, described in more detail later in this chapter, is called *instantiation*. There is one template file for each configuration file. Template files are located in the various `$(PRODUCT)_TOP/admin/template` directories on the application tier, and in the `<s_base>/11.2.0/appsutil/template` directory on the database tier.

Template files used by AutoConfig can be divided into the following categories:

Templates for APPL_TOP Configuration Files - These are either files requiring configuration-specific information in the `APPL_TOP`, or files used to load configuration profiles into the Oracle E-Business Suite database.

Templates for Management Scripts - To run all the standard processes required by Oracle E-Business Suite, Rapid Install creates scripts to start and stop each of these required processes. These scripts need configuration information in order to:

- Create the correct environments for each process
- Start the processes with the correct parameters
- Point the processes at the correct database instance (if applicable)

Driver Files

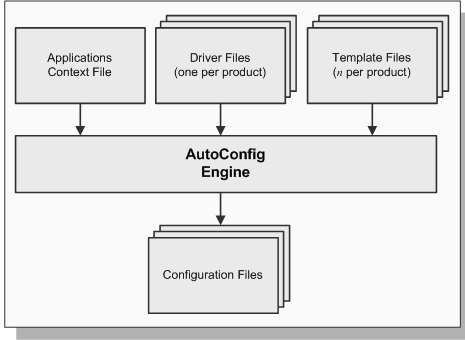
AutoConfig driver files are used to list the corresponding template files and locations, and specify the commands to be executed. For example, the commands might update profile options.

Driver files are located in each `$(PRODUCT)_TOP/admin/driver` directory on the application tier, and in the `<s_base>/11.2.0/appsutil/template` directory on the database tier.

Configuration Files

AutoConfig configuration files, with a `.conf` extension, are created when AutoConfig instantiates the corresponding template files. The values in the configuration files correspond to site-specific settings. AutoConfig creates numerous configuration files in various directories.

Relationship Between AutoConfig Files



Instantiation

As mentioned, instantiation is the process whereby AutoConfig creates a configuration file with contents tailored for a specific environment. AutoConfig can be used to instantiate files or scripts, and then execute them for installation and configuration.

Examples of instantiation include:

- Instantiation of a configuration file to be used at runtime
- Instantiation of an SQL script to set profile options
- Instantiation of a shell script or Windows command file to run an SQL script in SQL*Plus
- Instantiation of scripts to start up and shut down application tier services

The `adautocfg.sh` script updates configuration files and profile options in the following way:

1. Instantiates template files with instance-specific values derived from the relevant context file
2. Copies in any customizations
3. Overwrites existing configuration files with newly instantiated ones
4. Runs SQL scripts to update database profile options

Role of the template and driver files

AutoConfig uses *template* files to determine the basic settings needed. There is one template file for each configuration file. The *driver* files list the names and locations of the files that need to have context variables replaced. They also define the phases into which instantiation is divided, and specify the commands that are to be executed for specific products. When AutoConfig runs, it cycles through the various `$(PRODUCT)_TOP/admin/driver` directories looking for driver files such as `adtmpl.drv`, `fdtmpl.drv`, and `icxtempl.drv`.

Additional Information: For further details of AutoConfig operation, refer to Chapter 3, Technical Configuration, of *Oracle E-Business Suite Setup Guide*.

Execution of Scripts

As well as its instantiation activities, AutoConfig may execute other scripts, depending on the requirements of the specific Oracle E-Business Suite system.

Phases of Operation

As AutoConfig parses the driver files, it carries out a series of actions, grouped into several distinct phases:

- **INSTE8** - Instantiates AutoConfig template files to the AutoConfig configuration files specified in the relevant template driver files.
- **INSTE8_SETUP** - Executes setup scripts that carry out activities not involving connection to the database.
- **INSTE8_PRF** - Executes setup scripts that update profile options.
- **INSTE8_APPLY** - Executes setup scripts that carry out activities involving updates to the database.
- **BINCPY** - Copies the file mentioned from the source file to the configuration file, creating parent directories for the latter if necessary. AutoConfig will report an error if the source file cannot be found.
- **BINCPY_IGERR** - Copies the file mentioned from the source file to the configuration file, creating parent directories for the latter if necessary. AutoConfig will *not* report an error if the source file cannot be found.

AutoConfig carries out these actions in the following order:

1. **All INSTE8 and BINCPY actions** - Carries out all file instantiations called for during INSTE8, INSTE8_SETUP, INSTE8_PRF and INSTE8_APPLY, and all copying from source files to target configuration files.
2. **INSTE8_SETUP actions** - For the files that were instantiated in Step 1, AutoConfig runs all SETUP scripts.
3. **INSTE8_PRF actions** - For the files that were instantiated in Step 1, AutoConfig runs all PRF scripts.
4. **INSTE8_APPLY actions** - For the files that were instantiated in Step 1, AutoConfig runs all APPLY scripts.

At the end of this process, the required configuration files and profile options have been created for the Oracle E-Business Suite installation.

Management Tasks

There are several areas in which an administrator can use AutoConfig to update, control, and monitor an Oracle E-Business Suite system. While many tasks will normally be undertaken from Oracle Applications Manager, you may on occasion need to run a script (command file on Windows) from the command line.

Managing the Context

Oracle Applications Manager (see Chapter 7) enables you to edit the Applications context as required. From the Administration tab, choose AutoConfig and click on *Edit Parameters* for the relevant context file. After making a change to the context, you must run AutoConfig to update the relevant configuration files. Before doing so, you should examine the proposed changes by running the `adchkcfg.sh` configuration check script (described below under *Checking the System*).

Warning: Do not edit E-Business Suite configuration files manually. Any changes will be lost when AutoConfig is next run.

An AutoConfig-managed environment such as Release 12.2 will also be updated by applying an Oracle E-Business Suite patch that adds or modifies variables in the Applications context file, AutoConfig template files, or AutoConfig driver files.

On occasion, you may need to undo configuration changes that have been made. The previous configuration can be restored by running the `restore.sh` utility, which enables you to roll back the changes made by an AutoConfig run. This is achieved by utilizing the backup copies of the configuration files that are created when AutoConfig is run.

The backup files are located in `$(INST_TOP)/admin/out/<MMDDhhmm>` on the application tier, and `<s_base>/11.2.0/appsutil/out/<CONTEXT_NAME><MMDDhhmm>` on the database tier, where the `<MMDDhhmm>` directory name indicates the month, day, hour and minute of the AutoConfig run.

You can restore the configuration that existed immediately before the current one by navigating to the current directory and running the `restore.sh` script. To restore an earlier configuration, you must use the *Context File History* feature of Oracle Applications Manager.

Build Context Utility

The `adblxml` utility, which was introduced in Release 12.1, can be used to create a new context file on the database tier. The new file acts as a central repository for database configuration information, and is essential in enabling AutoConfig on the database tier of an Oracle E-Business Suite instance.

A new context file on the database tier can be created using the following commands:

```
$ cd <s_base>/11.2.0/appsuil/bin $ perl adblxml.pl \ [template=<contextfile_template>] \ [out=<contextfile_name>
```

Where `<contextfile_template>` is the location of the context template (default is `<s_base>/11.2.0/appsuil/template/adxdbctx.tmp`), and `<contextfile_name>` is the absolute path of the context file to be generated (default is `<s_base>/11.2.0/appsuil/<context_name>.xml`).

Note: The `adblxml` utility is only supported on the database tier, not on the application tier.

Controlling the System

AutoConfig utilizes a number of application tier control scripts, located in `$INST_TOP/admin/scripts`.

Key Application Tier Control Scripts	
Script Name	Function
<code>adstrtal.sh</code>	Starts all application tier server processes
<code>adstpall.sh</code>	Stops all application tier server processes
<code>adautoCfg.sh</code>	Runs AutoConfig

The corresponding directory on the database tier is `<s_base>/11.2.0/appsuil/scripts/<CONTEXT_NAME>`, where control scripts allow the database and database listener processes to be started and stopped, and AutoConfig to be run.

Checking the System

Several other scripts facilitate system management in an AutoConfig-managed environment such as Release 12. For example, you can identify the effects of proposed changes before you make them.

Configuration Check Utility

`adchkcfg.sh` is located in `<AD_TOP>/bin` on the application tier, and in `<s_base>/11.2.0/appsuil/bin` on the database tier.

This utility generates a report that highlights differences between existing configuration files and the new ones that AutoConfig will generate. The report is called `cfgcheck.html`.

Running `adchkcfg.sh` is useful both in carrying out a test run before a planned environment change is made, and when investigating problems. In Release 12.1, the tool was enhanced to report important database updates, generate a report for database changes, and improve readability of reports.

Additional Information: For further details on managing system configuration parameters, see *Oracle E-Business Suite Maintenance Guide*.

AutoConfig Profiler

Introduced in Release 12.1, the profiling feature provides a consolidated HTML report of an AutoConfig run. The top level of the report displays a summarized view that lists all the product tops, along with the total instantiation and execution time of the templates within each product top. You can drill down into the report to view additional details, including the source and target location of each template, the time taken to instantiate and execute individual template scripts, and the execution report for each template script.

AutoConfig can be run in profile mode on the application tier using the following command:

```
$AD_TOP/bin/adconfig.pl contextfile=<CtxFile> \ [product=<product_top>] -profile
```

Where `<CtxFile>` is the absolute path to the application tier context file, and `<product_top>` is the short name of the product to configure.

AutoConfig can be run in profile mode on the database tier using the following command:

```
perl <s_base>/11.2.0/appsuil/bin/adconfig.pl \ contextfile=<CtxFile> -profile
```

Where `<CtxFile>` is the absolute path to the database tier context file.

Enhanced Service Dependency Management

In addition to the above utilities, AutoConfig's service management infrastructure has been enhanced with dependency information between service groups and services. This dependency information is stored as metadata in the context file.

The following XML attributes were added in Release 12.1:

- **services:** contains a comma-separated list of the services that belong to a particular service group.
- **incompatible_with** (optional): contains a comma-separated list of service groups and services that are mutually exclusive with the service group or service for which they are defined. For example, Forms Socket mode and Forms Servlet mode are mutually incompatible.
- **requires** (optional): contains a comma-separated list of the required service groups and services.

Running AutoConfig in Parallel

Introduced in Release 12.1, the *parallel mode* feature enables AutoConfig to be run simultaneously across multiple nodes of an Oracle E-Business Suite system. When running in parallel mode, AutoConfig uses the `dbms_locks` PL/SQL package to ensure that configuration of one node does not interfere with configuration of other nodes: this is required because certain AutoConfig configurations of one node depend on the configurations of other nodes.

Additional Information: During the instantiation phase, locking is done at script level. During the execution phase, locking is done at `product_top` level.

AutoConfig can be run in parallel mode on the application tier using the following command:

```
perl $AD_TOP/bin/adconfig.pl contextfile=<CtxFile> \ [product=<product_top>] -parallel
```

Where `<CtxFile>` is the absolute path of the application tier context file and `<product_top>` is the short name of the product to be configured.

AutoConfig can be run in parallel mode on the database tier using the following command:

```
perl <s_base>/11.2.0/appsuil/bin/adconfig.pl \ contextfile=<CtxFile> -parallel
```

Where `<CtxFile>` is the full path to the database tier context file.

Note: When using parallel mode, AutoConfig must be invoked with the `-parallel` option on *all* nodes of the relevant tier.



[About Oracle](http://www.oracle.com/corporate/index.html) | [Contact Us](http://www.oracle.com/us/corporate/contact/index.html) | [Legal Notices](http://www.oracle.com/us/legal/index.html) | [Terms of Use](#)

<http://www.oracle.com/us/legal/terms/index.html> | [Your Privacy Rights](http://www.oracle.com/us/legal/privacy/index.html) | [Cookie Preferences](#) | [Ad Choices](#)
privacy-policy.html#12) | [Copyright © 2025, Oracle and/or its affiliates.](http://www.oracle.com/pls/topic/lookup?ctx=cpyr&id=en)