

ActiveBase Performance™ for controlling Toad, SQL*Plus and other development tools

Rule Example Package

Example scenarios for AB*Performance and Priority

General

SQL*Plus, Toad, PL/SQL developer are among the tools that internal IT and outsourced DBA use for their daily operation. These tools are open and powerful, yet unintentionally can cause production downtime and performance degradation.

One of our largest customers managing more than 500 databases concluded that Toad misuse by DBAs and developers in production environments cause two downtime occurrences per database every year, as Quality of Service (QoS) plummet and precious time is wasted on corrective actions while enterprise productivity is lost of the business side.

ActiveBase Performance Suite protects your application performance and ensures high availability.

Our Performance software module gives you the capability to block uncontrolled Jobs and 'run-away' Reports and Queries.

Our Priority software module gives you the power to align in real-time computer resources with business objectives to optimize performance, mitigate risk and reduce costs.

It enables both ad-hoc resource management (for operators), as well as powerful rules that guarantee service levels

- Identify resource contentions in real-time
- Identify 'out-of-control' jobs 'run-away' reports and queries, restricting their resource consumption to acceptable consumption levels, preventing resource outage
- Restrict real-time OS process resource consumption (such as Zip or archiving process)

The attached XML files include the ActiveBase Performance and Priority rules, explained in this document. These files include:

1. ABPerformance Toad Rules.xml – rules to be imported into ActiveBase Performance rule editor
2. ABPriority_Toad_Session_Rules.xml – rules to be imported into ActiveBase Priority Session rules, and
3. ABPriority_Toad_Group_Rules.xml – rules to be imported into ActiveBase Priority Resource Group rules



Details

1. Controlling full-scan, parallel and DML commands from development tools

Create a 'routing action' for the development tools:

Edit Application

Application Name: toad

Include Table

Program	Host	OSUser
%toad	%	%
%pl/sql%	%	%

Insert Delete

Exclude Table

Program	Host	OSUser
%	dba1	%
%	dba2	%
%	dba2	%

Insert Delete

OK Cancel

Client application definition for grouping developers using a single rule tree

In this example, Application group includes all development tools except those used by the DBA team (we excluded them using the 'exclude table' hosts). This enables to create a dedicated routing action with rules specifically for developers running against production databases.

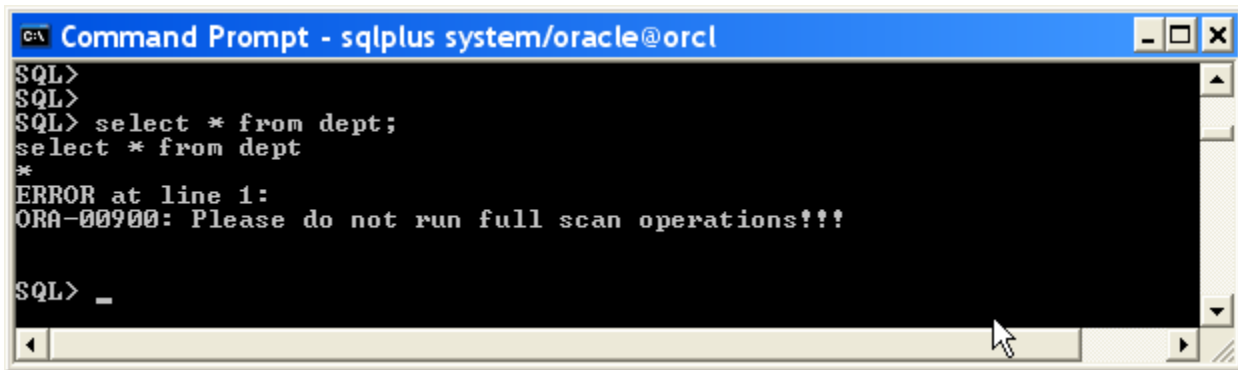
Import the ABPerformance Toad Rules.xml rules into the appropriate Routing action.

Routing action rules include the following rules:

- a. Block full scan operation on dept table rule

Matcher: identify by execution plan step, for table dept, operation=Table Access, Option=Full, action – Block,

Action: Block with error message returned: 'Please do not run full scan operations!'

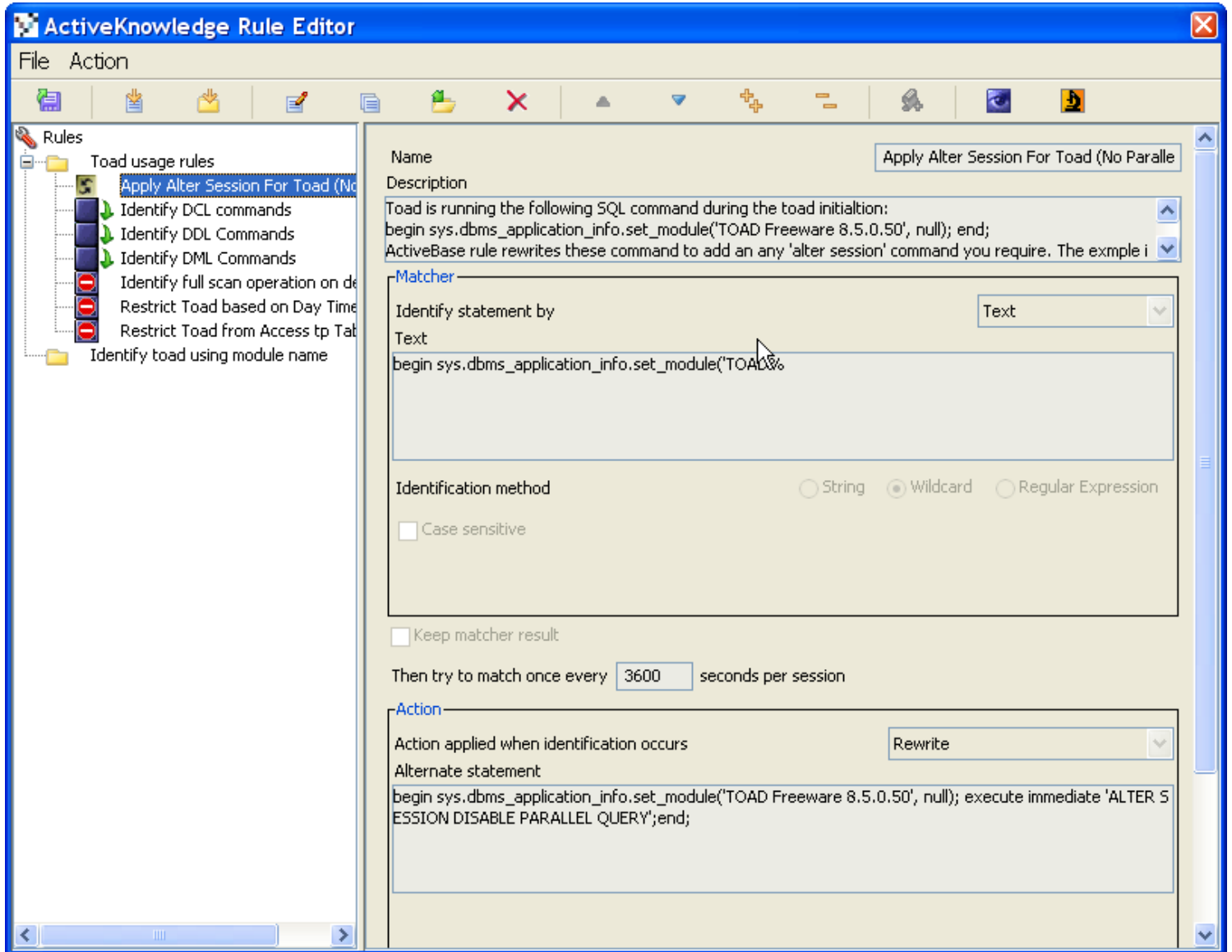
A screenshot of a Windows Command Prompt window titled "Command Prompt - sqlplus system/oracle@orcl". The window has a black background with white text. The text shows a SQL*Plus session where the user enters "SQL> select * from dept;" followed by "select * from dept" and a cursor. The output shows "ERROR at line 1:" and "ORA-00900: Please do not run full scan operations!!!". The prompt "SQL> _" is visible at the bottom.

```
CA: Command Prompt - sqlplus system/oracle@orcl
SQL>
SQL>
SQL> select * from dept;
select * from dept
*
ERROR at line 1:
ORA-00900: Please do not run full scan operations!!!

SQL> _
```

SQL*Plus session with a full scan request blocked, returning a message back to the client

- a. Rule name: Apply Alter Session for Toad (No Parallel). The rule purpose is to prevent high parallel queries executed by Toad users.
Matcher: identify SQL request - 'begin sys.dbms_application_info.set_module('TOAD%' ' which toad uses during the connection phase,
Action: Rewriting it with: 'begin sys.dbms_application_info.set_module('TOAD', null);
execute immediate 'ALTER SESSION DISABLE PARALLEL QUERY';end;'
- b. Rule name: Identify DCL, DDL and DML for auditing or for blocking unauthorized commands to production



ActiveBase Performance Rule tree

2. Controlling resource consumption on the session level, using the Session rule engine during peak load

Import the AB*Priority ABPriority_Toad_Session_Rules.xml rules into the Session rules. These rules have been created as follows:

- a. A folder identifies server high-CPU load condition (identification: Server CPU > 70%)
- b. Within the folder:

Identify: Toad sessions (identified by session module name '.*TOAD.*' using Session module identification using regular expression – where in regular expression '.*' equals '%'), and toad sessions that run more than 600 seconds (identified using session connect time from Oracle statistics > 60000).

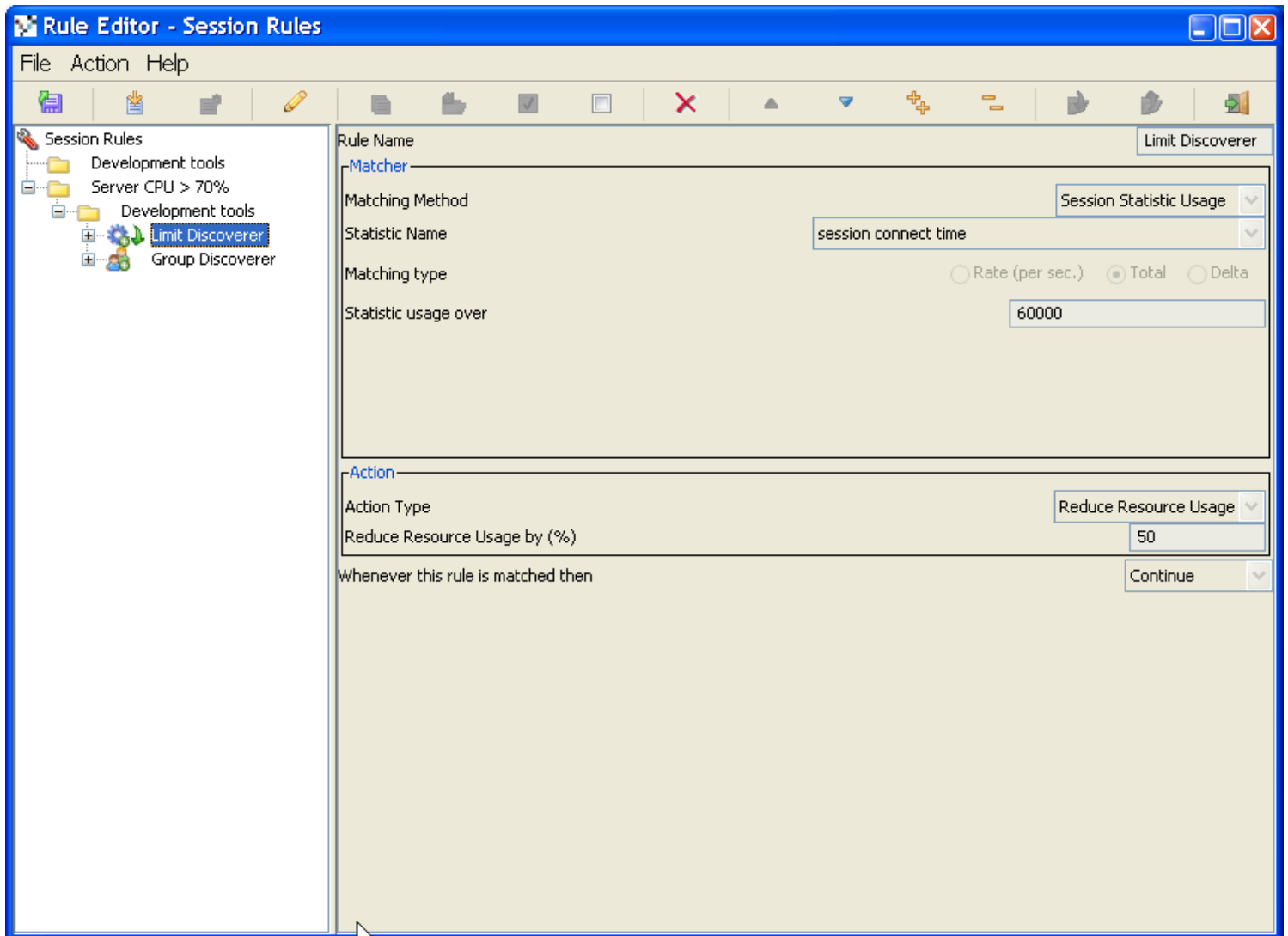
Action: Reduce toad resource consumption by 50%,

- c. Within the folder (when high-load occurs), group all development tools together for managing their group resources in the Session Grouping Rules.

Identify: If the server CPU level is above 70% assign resource group 'dev tools' to all development tools identified by session-module '*.toad.*'

(**note:** in regular expression, identifying both toad **OR** navigator programs use: '.*(toad)|(navigator).*' where the pipe sign means 'OR' condition).

Action: Reduce toad resource consumption by 50%, identified by session module name '*.TOAD.*' (using Session module identification using regular expression where '.' equals '%').



Session rule tree structure for development tools

3. Controlling resource consumption on the Group level, using the Resource Group rule engine

AB*Priority enables both controlling resources on the session level as well as restricting resource consumption to a group of sessions.

AB*Priority provides the ability to group similar activities and control the resource consumption of the whole group of sessions (e.g., the total load of the Toad and other development tools on the production server), complementary to the specific session level resource management.

An action 'assign resource group' has been added to the session rules in order to categorize sessions into different groups. Groups are validated in the Resource Grouping Rules, immediately after all the sessions have completed validation in the session rules.

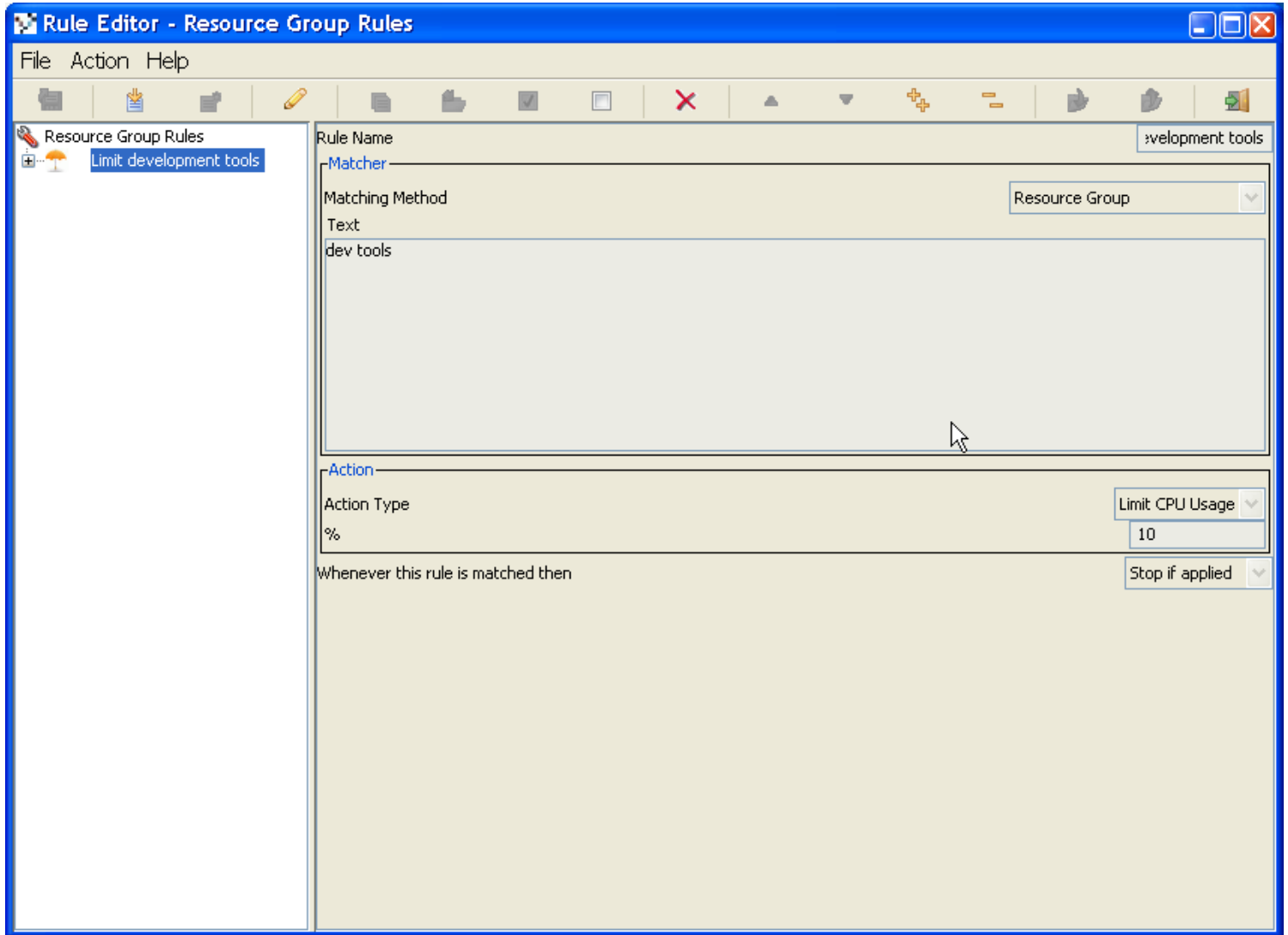
AB*Priority manages resource allocation dynamically within the group sessions. Example: a 10% resource restriction was applied on Toad sessions. A single Toad session consumes 5% of server resource would not trigger AB*Priority resource reduction, yet in case 4 Toad sessions run concurrently, consuming 20% server resources, AB*Priority reduces all of them by 50% each.

Import the ABPriority_Toad_Group_Rules.xml rules into the Resource Group rules. These rules have been created as follows:

- a. Control the group of Toad and other development tools sessions.

Identify: If the server CPU level is above 70%

Action: Limit the resource consumption of the group to 10%



Resource Group rule tree structure for development tools

www.active-base.com

400-00101-311 | 02/09 | © 2009 ActiveBase, Ltd. All rights reserved. All other third-party trademarks are the property of their respective owners.

