Web-Oracle gateway Oralink: Accessing Oracle7 databases from the Web

V.V.Korenkov, A.B.Rybalkin and I.S.Tkatchenko

LCTA Joint Institute for Nuclear Research, 141980 Dubna Russia

Web-Oracle gateway ORALink enables World Wide Web administrators to tightly integrate Oracle7 databases with the Web and to bring to the Internet and Intranets exciting new services and informational systems. Using the ORALink is a much quicker than starting up Oracle Forms, and gave every client's hardware platform almost equal access to the information worldwide. ORALink was developed as CGI 1.1 compliant script which allows to use it with any CGI 1.1 compliant HTTP server. Build-in interpreter allows use of substitutions and should hopefully not limit document designer neither in SQL statements length nor in SQL sintax.
The purpose of this paper is to aid web and database administrators in the usage of ORALink gateway; tasks needed to use ORALink are explained, as well as detailed description of Oralink's programming. This paper describes ORALink version 2.0, which is the current version at the moment. The up-to-date version of this document available on-line as http://oradb1.jinr.ru/software/oralink/guide.htm

1 Operation

It's assumed you have an appropriate knowledge in writing HTML forms before using ORALink. ORALink is run by invoking the oralink.cgi script by your HTTP server, and uses either POST or GET method to get output from form.

To use ORALink gateway you should design a HTML form first and then call oralink.cgi from this form. ORALink does not allow construction of the forms on the fly (because i still belive that only human can do it well). Within the form you can use ORALink' commands and substitutions in any combination and quantity. So the feel and look you will get using ORALink is only limited by currently implemented commands and your imagination. If you have an idea about new commands to implement, your suggestions would be appreciated by authors , and probably will be implemented in the next versions of ORALink.

2 Commands

There're some commands based on HTML 3.0 specification, which are not fully supported by all browsers at the moment of writing this note. So use them according to features of your browser.

Commands will be executed in order of appearence inside form, except ones (which if provided will be executed first):

- http_referer_allow
- connect
- profile_home
- config_cmd
- oracle_library

This approach allows the form author to change different settings for query output within single form. Commands are not case-sensetive.

Commands are:

1. connect

   Oracle connect string (please consult your Oracle documentation about connect string syntax).
   Default: do not connect to Oracle.
   Notes: 1, 4, 5
   See also: Security considerations
   Eg (form): <input name="connect" type="hidden" value="UnsecureName/UnsecurePassword">
   Eg (profile): connect UnsecureName/UnsecurePassword

2. title

   HTML document title.
   Eg (form): <input name="title" type="hidden"
   value="My query output">
   Eg (profile): title My query output

3. sql

   Any SQL statement or statements, separated by ";".
   Eg (form): <input name="sql" type="hidden"
   value="select id, salary from employers where salary > 120000">
   Eg (profile): sql select id, salary from employers where salary > 120000

4. table_output

   Representation of table output.
   Possible values:
   - table - represent output via HTML <TABLE> element.
   - pre {} - represent output via HTML <PRE> element.
   - plain - non-formatted output.
   Default: pre
   Eg (form): <input name="table_output" type="hidden" value="table">
   Eg (profile): table_output table

5. table_header

   Allows to control the table header printing.
   Possible values:
   - on - print table header.
   - off - do not print table header.
   Default: on Eg (form): <input name="table_header" type="hidden" value="off">
   Eg (profile): table_header off

6. table_tag

   Allows to control the attributes of <TABLE> tag.
   Eg (form): <input name="table_tag" type="hidden" value="BORDER=0 CELLPADDING=5">
   Eg (profile): table_tag BORDER=0 CELLPADDING=5

7. table_tag_tr

   Allows to control the attributes of <TR> tag.
   Eg (form): <input name="table_tag_tr" type="hidden" value="ALIGN=middle">
   Eg (profile): table_tag_tr ALIGN=middle

8. table_tag_th

   Allows to control the attributes of <TH> tag.
   Eg (form): <input name=”table_tag_th” type=”hidden” value=”ROWSPAN=2”>
   Eg (profile): table_tag_th ROWSPAN=2

9. table_tag_td

   Allows to control the attributes of <TD> tag.
   Eg (form): <input name=”table_tag_td” type=”hidden” value=”ALIGN=middle”>
   Eg (profile): table_tag_td ALIGN=middle

10. html

    Allow you to merge any HTML to output produced by ORALink.
    Notes: 3
    Eg (form): <input name=”html” type=”hidden” value=”<h2>
    Dear customer, </h2>
<p>
    The sales on eggs are represented in the following table:<hr>”>
    Eg (profile): html <h2>Dear customer,</h2>

<p> The sales on eggs are represented in the following table:<hr>

11. file

    Allow you to merge the content of any file to output produced by ORALink.
    Notes: 3
    Eg (form): <input name=”file” type=”hidden” value=”/usr/etc/httpd/htdocs/header.html”>
    Eg (profile): file /usr/etc/httpd/htdocs/header.html

12. rec_limit

    Limit the number of records to output. No limit, if negative.
    Default: No limit.
    Eg (form): <input name=”rec_limit” type=”hidden” value=”50”>
    Eg (profile): rec_limit 50

13. profile_home

    Sets up profiles home directory ORALink will use to look for a profiles.
    Eg (form): <input name=”profile_home” type=”hidden” value=”/usr/etc/httpd/oralink_profiles”>
    Eg (profile): profile_home /usr/etc/httpd/oralink_profiles

14. profile

    Allows to read ORALink arguments from profile file.
    See also: Profiles
    Eg (form): <input name=”profile” type=”hidden” value=”/usr/etc/httpd/oralink_profiles/general”>
    Eg (profile): profile /usr/etc/httpd/oralink_profiles/general

15. config_left_subst_sep

    Allows to define left substitution separator ORALink use to identify substitutions.
    Default: ::
    Eg (form): <input name=”config_left_subst_sep” type=”hidden” value=”!!”>
    Eg (profile): config_left_subst_sep !!

16. config_right_subst_sep

    Allows to define right substitution separator ORALink use to identify substitutions.

Default: ::
Eg (form): <input name=”config_right_subst_sep” type=”hidden” value=”!!”>
Eg (profile): config_left_subst_sep !!

17. config_sql_sep

    Allows to define SQL statements separator.
    Default: ;
    Eg (form): <input name=”config_sql_sep” type=”hidden” value=”;;”>
    Eg (profile): config_sql_sep ;;

18. config_sql

    Allows to define SQL command name.
    Default: sql
    Eg (form): <input name=”config_sql” type=”hidden” value=”query”>
    Eg (profile): config_sql_sep query

19. config_cmd

    Allows to provide the following settings (former cmd arguments):

    - -d Debug mode. Will provide you with alot of debug output, and could be used to track down the problem.
    - -s1 Suppress output of connect string.
    - -s2 Supress output of copyright info.
    - -s3 Supress output of tailer.
    - -s4 Supress output of rows processed count.
    - -s5 Supress output of rows limit reached count.
    - -s6 Supress output of time counters.

    Eg (form): <input name=”config_cmd” type=”hidden” value=”-s1s2s3s4s5s6”>
    Eg (profile): config_cmd -s1s2s3s4s5s6

20. http_referer_allow

    Allows to restrict access to ORALink only from ”trusted” URL’s (based on HTTP_REFERER header). Note not all client programs supply this header.
    Default: no restrictions
    Eg (form): <input name=”http_referer_allow” type=”hidden” value=”http://myhost”>
    Eg (profile): http_referer_allow http://myhost

21. oracle_library

    Required and defines an OCI library (DLL) to use on Win32 platform. OCI library is included in Oracle products and could be located at %ORACLE_HOME%\bin directory. Normally, it’s named either oraVER.dll or orantVER.dll, where VER> is an Oracle version number.
    Notes: 6
    Eg (form): <input name=”oracle_library” type=”hidden” value=”c:\orant\bin\ora73.dll”>
    Eg (profile): oracle_library c:\orant\bin\ora73.dll
    Notes:

1. Could be redefined multiply times within form, but only first definition is actually used.
2. Doesn’t work after any output produced.
3. Doesn’t go through sintax analyzer, so no sintax errors will be detected.
4. All other commands which are affected by this command are ignored.
5. Connect string (if not suppressed) outputs at the place this command appeared inside form.

6. Only available on Win32 platform.

3 Substitutions

Substitution tags are used by ORALink' build-in interpreter to identify substitution. Any statements (except spaces) between left and right substitution tags within the value of argument are treated to be substituted, and the same within the name of argument are treated as substitution. Use of substitutions allows form author to construct a non-trivial statements. Substitutions are not case-sensetive. Substitutions could be used with arguments of both forms and profiles.

4 Profiles

Another way to pass arguments to ORALink is by using profiles. Profiles are plain-text files which contain arguments one per line (that is, each line should contain no more than one argument). Argument names are separated from argument values by ' ' (space). The 2 probable reasons you would like prefer to use profile over form elements are:

- You don't need to "escape" HTML special characters inside values of arguments passed via profiles.
- Profiles are readed directly from disc by ORALink, so it's no way for a client to be able to view its content.

The tipical profile can contain:

```
connect username/password
profile_home /usr/etc/httpd/oralink_profiles
# profile_home c:\httpd\oralink.pro
# oracle_library c:\orant\bin\ora73.dll
config_cmd -s1s2s3s4s5s6
sql select secret_data from secret_table
```

Profile is invoked by profile command and their arguments are merged in place of this profile command. There's a possibility to make ORALink always to use at least one profile for each request: if there will be an ORALINK_PROFILE variable in the ORALink process environment, its value will be treated as a full pathname of profile which will be processed by ORALink at first.

There's a special command profile_home, which can be used to set up profiles home directory ORALink will use to look for a profiles.

You can use substitutions inside profiles.

5 Security considerations

It's recommended to use aceess from Web to Oracle under special Oracle accounts. Use of profiles can help to hide confidential information such as Oracle loginname/password, database object names and logical structure. http_referer_allow command can be used to allow access to ORALink only from documents located on "trusted" websites.