WHITE PAPER

Shortcut for Oracle Developer Runtime deployment
Version 1.0

Aleš Kavšek
OCP-DBA, MCSE, MCP+I, A+, Network+
September, 2001

OVERVIEW

The main objective of this document is to help you deploy Developer Runtime in client-server environment fast, effective and without additional costs (e.g. Forms Server, IAS) and complicated setup efforts (e.g. deployment through SMS or Group Policy in Windows 2000). Proposed solution (if prepared, tested and implemented carefully) will cut at least 90% of the time you need to upgrade your clients machines with new version of Developer Runtime.
More clients you have, more time (money) you'll save.
What motivated me to write this paper?
Again and again I read messages in newsgroups from developers, system administrators and mostly from newcomers such as this one: "Is it possible to compile Forms application in EXE file so that I can avoid installation of Runtime at my clients?" Well, the answer is NO, you can't do that.
Oracle position on the issue is clear: Thin-Is-In, you should convert all your C/S applications to web forms or use Developer Server to lower your application TCO.
But is there any hope for us (Lusers), that are not ready yet for the prime time, or for whatsoever reason can't move C/S application to 3-tier architecture, yet? You, bet!

At least you can do the following with Developer Runtime:

- **you can run Forms applications on "thin" client** (Windows 9x, ME, NT, W2K). The only thing you need on client PC is TCP/IP protocol (or any other supported communication protocol). There is no need for local SQL*Net/Net8, Developer Runtime or any other Oracle Software to be (pre)installed.

- **you can run ANY version of Developer Runtime concurrently** without side effects, even if Developer versions are declared by Oracle not to be compatible with each other! For example, if your customer is running application developed in Forms 6.0 Release 1 by company X and your company Y is using Forms 6.0 Release 3 then you can still deploy your application without any side effects on company X application. (Support for multi-homed installation of Developer products was introduced with Developer 6i, proposed solution works with any version!)

- **you don't need local administrator privileges to install Developer runtime** (or Discoverer) at clients. Users themselves can install runtime with some basic batch scripts from distribution file server or custom prepared CD-ROM. Imagine upgrading several hundred clients simply by emailing them shortcuts to run application from file server or by sending them installation script that will copy Developer Runtime, Net8, utilities...to client PC. (Can you still remember those days - without registry - when this was actually trivial task of editing/saving central INI files and running app. from file server?)

- **you don't need to perform any action on local Windows registry**, in fact if you prepare your runtime according to instructions in this paper, application doesn't need registry at all (you can test this after you prepare your runtime by deleting HKLM\Software\Oracle from registry). By the way, this is nothing new, but most Oracle developers and system administrators are convinced that Oracle Software doesn't work on Windows 9x/NT/W2K without values in registry key HKLM\Software\Oracle. You would be surprised if you knew how many applications and utilities are working fine if you replace all those values with environment variables and simply launch application from batch file.

- **you can run Forms application from read-only media (including runtime)**. For example you can write your runtime and your Forms/Reports application on CD-ROM, take disc to the customer and run application from CD-ROM on any Windows PC, all you need is TCP/IP (or any other supported protocol) on client machine with access to database server to login -- you don't need locally installed Oracle Software whatsoever!

KNOWN ISSUES AND LIMITATIONS

-   during testing and runtime deployment at customer site with Windows 9x clients I found out that default environment size for shell (command prompt) is usually set too low. Depending on the size of your RunFrm.bat file (number of SET variables) you'll need to increase environment size in config.sys and reboot:

    Config.sys
    SHELL=C:\DOS\COMMAND.COM **/e:4096   <- increase this number if necessary**

    Valid range of values for /E switch is 160-32768 bytes, start with 4096 and then increase the value by 2048 bytes.

Note: This is not an issue with Windows NT/2000/XP.

-   due to the fact that we don't use OCX/VBX/OLE in our forms I didn't truly test if there are any issues running forms that contain these items. I think you shouldn't have problems running such forms -- in the worst scenario you'll need to register component(s) with regsvr32.exe while logged on with administrative privileges. You can use simple script that will do that:

    copy myform.ocx %systemroot%\system32
    regsvr32 /s %systemroot%\myform.ocx
    ...

Note: You can usually find installation script for OCX registration on Developer CD-ROM.
 Batch file is usually in directory with OCX files, just search for *.OCX files on \WIN32 subtree and check for any *.bat files that are in the same directory.

-   I didn't test Graphics Runtime for the same reason mentioned before -- we don't use this tool -- and I believe that most of you neither. However, I'm pretty sure that Graphics Runtime will work for you if you install and use them by following instructions in next section (I always include Graphics Runtime in our deployments - just in case).

-   and last but not least, as you know there is no such thing as free lunch, so what are the shortcomings of proposed method for rapid runtime deployment? First, you need some basic skills and patience to prepare runtime - take time to test runtime environment and don't rush to deploy first version you make.
    Second, if you already have Oracle Client Software installed at your client site then you'll need some extra disk space for your runtime on these machines locally (or on file server if you choose to run runtime from server!).
    Personally, I don't count much on Oracle Support Services in case of difficulties, so lack of official support from Oracle is not an issue for me -- if their support is of most importance to you then you should reconsider other deployment options such as Developer Server. (So far I didn't encounter single problem with Forms/Reports that could be linked to the way runtime is prepared and deployed at our site.)

Note: I'm sure you'll find some additional limitations and/or workarounds to improve proposed solution. Please share your findings and experience with others!
You can always send comments to kavsek@yahoo.com.

STEP BY STEP INSTRUCTION TO COOK YOUR RUNTIME

What ingredients do you need ?

- one 'clean' PC with Windows OS (preferable Windows NT) without any Oracle Software. Make sure that protocol of your choice (most likely TCP/IP) is installed on this PC and that it works!

> You don't have such PC? No problem. Before you read any further be warned that you should backup your machine in case you do something stupid :-)
> You can temporarily "remove" Oracle Software simply by renaming %OracleHome% (e.g. C:\ORANT) into something else (e.g. C:\ORANT_BACK) then export HKEY_LOCAL_MACHINE\Software\ORACLE key into file (e.g. orant_back.reg) with Registry->Save Key. Save this file in C:\ORANT_BACK because you'll need it later to restore your original Oracle Software. Finally delete HKEY_LOCAL_MACHINE\Software\ORACLE key. Now you have 'clean' PC.

> **Warning** - I'm presuming that you're comfortable with regedt32.exe and the registry structure and that it's not your first time using this tool! If I'm wrong then all I can tell you is: I hope you have a good backup and some guru at your disposal! :-)

- you'll need some third party tools that can help you prepare runtime environment and in the process of troubleshooting (tools are not necessary, just recommended!):

  - GNU diff.exe is used to compare the content of %systemroot%\system32 before and after Oracle Software installation to identify any new library installed by Oracle Installer. You can use other tool if you have one, for example Windiff from NT4/W2K Resource Kit.

    You can download free version of diff from mirror site near you:

    http://www.gnu.org/order/ftp.html

    Mirror site in Slovenia: ftp://ftp.arnes.si.org/gnu/djgpp/v2gnu/diff272b.zip

  - GNU AWK (gawk.exe) if you wish to use script in Appendix A (see later in this section) that will help you prepare RunFrm.Bat file.

    You can download free version of gawk from mirror site near you:

    http://www.gnu.org/order/ftp.html

    Mirror site in Slovenia: ftp://ftp.arnes.si/gnu/gnuish/gawk303x.zip

  - Process Explorer (ex. HandleEx) from Sysinternals is invaluable tool when you need to understand what resources (files, DLL's, registry keys) your application is using. It's my favorite tool when it comes to troubleshooting Windows applications. You can download free version from:

    http://www.sysinternals.com/ntw2k/freeware/procexp.shtml

Cooking time? Approx. 60-120 minutes (less if you're chef :-).

STEP 1

Turn on your 'clean' PC, logon as administrator and make a list of files in %systemroot%\system32 directory:

C:> dir /b %systemroot%\system32 > c:\sys32_1.txt

STEP 2

- install basic Oracle Client software (such as SQL*Net/Net8, SQL*Loader, exp, imp, SQL*Plus) in default oracle home (e.g. C:\ORANT)
- install your version of runtime for Forms/Reports/Graphics from Developer CD in the same Oracle Home (e.g. C:\ORANT).
- install Headstart if your Forms application depends on it
- setup SQL*Net/Net8 (TNSNAMES.ORA, SQLNET.ORA)

> **Note:** Remember, that you're not limited only to Developer Runtime products! We're using Discoverer 4 successfully prepared with the same procedure. Feel free to experiment!

STEP 3

Make another list of files in %systemroot%\system32 directory after Oracle installation:

C:> dir /b %systemroot%\system32 > c:\sys32_2.txt

Compare both files with DIFF or WINDIFF and copy any new files (DLL/OCX files) from \SYSTEM32 directory to %Oracle_Home%\BIN (e.g. C:\ORANT\BIN)

Example of file comparison using DIFF:

C:> diff -y -W 50 C:\sys32_1.txt C:\sys32_2.txt | MORE

You'll see newly installed files on the right side marked with **>**. Copy marked files to your %Oracle_Home%\BIN directory (e.g. C:\ORANT\BIN)

> **Note:** This approach has some weaknesses. What if your machine already had up-to-date Microsoft system files needed by Oracle. Oracle installer will not install them in that case and you'll not catch them with dir /b %systemroot%\system32. See troubleshooting section.

STEP 4

Test your installation by running your Forms/Reports application. Don't skip this step! It's important that you're sure that you have working environment before proceeding.
Make any changes to HKEY_LOCAL_MACHINE\Software\Oracle if necessary; such as forms/reports paths, nls_lang, nls_date, tns_admin, headstart etc.

STEP 5

Dump registry key HKEY_LOCAL_MACHINE\Software\Oracle to text file with REGEDT32. Open REGEDT32, navigate to HKEY_LOCAL_MACHINE\Software\Oracle, select Oracle key and then save key to file C:\ORAREG.TXT using menu option Registry->Save Subtree as.

> **Note:** Make sure that you don't use Registry -> Save Key option!

STEP 6

Prepare batch file that you'll use for launching Forms/Reports application. You can use GAWK script from Appendix A to parse your ORAREG.TXT file.

6.1. Prepare draft version of batch file -- let's call this batch file RunFrm.bat

C:>   gawk -f parsereg.awk ORAREG.TXT > RunFrm.bat

Open RunFrm.bat with Notepad and you'll see numerous SET statements, something like this:

```
SET API=C:\ORANT\DBS
SET BROWSER60=C:\ORANT\BROWSE60
SET CA_GPREFS=C:\ORANT
SET CA_UPREFS=C:\ORANT
SET COMPANY_NAME=Your Company
SET CORE40=C:\ORANT\CORE40
...
...
...
```

6.2 Replace drive letter (C:) with variable (%1)

Use Notepad Find/Replace to replace all your drive letters in batch file RunFrm.bat (in our case is C:) with environment variable %1. You'll get something like this:

```
SET API=%1\ORANT\DBS
SET BROWSER60=%1\ORANT\BROWSE60
SET CA_GPREFS=%1\ORANT
SET CA_UPREFS=%1\ORANT
SET COMPANY_NAME=Your Company
SET CORE40=%1\ORANT\CORE40
...
...
...
```

6.3. Add some lines at the end of batch file

See Appendix B for example of batch file that I'm using for launching Forms 6i application. Last three lines (in bold) I added manually.

I included SET TNS_ADMIN=%1\ORANT\BIN into batch file to prevent ORA-12154 errors in case somebody already put variable in registry and variable is pointing to wrong (proprietary) TNSADMIN.ORA file.

> **Note:** Remember one important principle of running (launching) forms/reports from batch file: environment variables (SET var=variable) always take precedence over values in registry, however if value for some variable is not set within batch file then value from registry is used. Common mistake is to skip TNS_ADMIN variable in batch file -- if such variable is present in registry on client machine and it points to wrong TNSNAMES.ORA file you can get error ORA-12154: TNS:could not resolve service name. This is why I explicitly included TNS_ADMIN at the end of batch file to prevent surprises!
> Some other interesting variables to look after are NLS_LANG and NLS_DATE_FORMAT.

Statement SET PATH=%1\ORANT\BIN;%PATH% is also very important. This statement will isolate your runtime from other Oracle Software that might be present on client machine. Your \ORANT\BIN will always be first in the search path which will greatly reduce any chance for dll conflicts. Second, if you copied Microsoft required files to \ORANT\BIN in STEP 3 you'll be able to run application even if those files are not present on client machine or are of different version.

Finally you start your Forms application with START command like this one:

        START %1\ORANT\BIN\ifrun60 MyFormsAPP.fmx

or you can make START command more generic to run any fmx:

        START %1\ORANT\BIN\ifrun60 %2

in this case you need to provide forms name as second command parameter.

Save any changes to RunFrm.bat file and proceed to step 7.

STEP 7

Test your runtime. First on your source PC and then on some other PC (see step 8)! Create new shortcut on your desktop with properties:

Target: C:\RunFrm.bat C:
Working Directory: C:\

 As you can see you need to point shortcut to your RunFrm.bat file and provide path to directory with your runtime, in our case it's C: (this parameter passed on command line will replace all %1 variables in your batch file and you'll see later that it can be any path).

When you click on shortcut you'll briefly see command prompt window and after awhile Forms runtime will show up.

Forms application should start if you carefully followed instructions (see section on Troubleshooting in case of difficulties).

STEP 8

Prepare distribution points.

> **Note:** I would strongly suggest that you write your \ORANT and RunFrm.Bat file on CD at this point for backup reason. You should also write a copy of %systemroot%\SYSTEM32 directory on CD in case you'll later find out that some DLLs are missing in your \ORANT\BIN directory.

Now you can copy \ORANT directory from your test PC to any location, for example on CD-ROM, to file server or in ANY directory you wish. For example you can copy \ORANT directory to F:\MYAPPS\APP6i\ORANT and all you need to run Forms from this directory is to change shortcut:

Target: C:\RUNFRM.BAT F:\MYAPPS\APP6i
Working Directory: C:\

You can also use UNC path to call your application from file server (let's say that you saved \ORANT on file server share \\FILESERVER\APP1\ORANT)

Shortcut:
Target: \\FileServer\App1\RunFrm.Bat \\FileServer\App1
Working Directory: C:\

- As you can see it's possible to run Forms on ANY client without manually installing ANY software locally, all you need to do is to copy "compact runtime" from your distribution point to local PC's and provide users with proper shortcuts that are calling RunFrm.bat file.

- You can prepare some batch procedure that you can email to users - perhaps in the form of menu - that will guide end user to copy runtime to his/her PC from distribution point (e.g. file server share). Logon scripts are good alternative for large deployments.

- Another option is to run application from file server. You can even burn CD-ROM with runtime and your Forms/Reports application and run directly from it.

- Now, you can run concurrently different version of Developer Forms without side effects because each process is totally separated from one another.

STEP 9

At this point you definitely deserve a cold beer :-)

TROUBLESHOOTING

Immediately after I run RunFrm.bat to start my application Runtime just disappears without any error message!?

This is almost certainly indication that some variable is missing in RunFrm.bat file. Perhaps you removed some variable from this file thinking that it's not important. Go back to your PC with original installation, repeat steps 5 - 6 and check if some variable is missing in your RunFrm.bat.

I'm receiving ORA-12154: TNS:could not resolve service name after entering Username, Password and Host. I already checked tnsnames.ora file in my runtime and it's OK. What is going on?

You didn't include SET TNS_ADMIN=%1\ORANT\NET80\ADMIN in your batch file. On client PC where you're running application is this variable probably present in registry and points to some TNSNAMES.ORA file with different service names (e.g. perhaps you're in situation when you're supporting customers with different contractors, maybe you're one of them, and obviously somebody added TNS_ADMIN in registry without notifying others - that's exactly what happened to me once). See Note on page 11.

After distributing runtime to several Windows 9x PC's I found out that application on some clients doesn't work. I'm receiving error messages that some files could not be found. I already checked shortcut and batch file RunFrm.bat and they're OK. What's wrong?

Check environment size for command prompt in CONFIG.SYS on problematic clients (this is not an issue on NT/W2K!).

Config.sys
SHELL=C:\DOS\COMMAND.COM **/e:4096   <- increase this number if necessary**

Valid range of values for /E switch is 160-32768 (in bytes), start with 4096 and then increase the value (if necessary) by 2048.

After starting runtime I receive message that some DLL could not be found!?

Check your RunFrm.bat file and see if you have valid SET PATH=%1\ORANT\BIN;%PATH% line before START command. If PATH variable is OK and you're still receiving error message then you missed something during test phase (step 7/8). Find missing library and copy it to \ORANT\BIN directory. It's highly recommended that you make copy of your %systemroot%\SYSTEM32 directory on CD-ROM in case you run into problems with DLL's.

Note: With Sysinternals Process Explorer you can easily find out what resources your application is using, such as DLL, OCX on source machine. When it comes to troubleshooting files related problems this is invaluable tool!

APPENDIX A

```
###########################################################
# PARSEREG.AWK by Aleš Kavšek -- Tested on NT4/W2K registry.
# This simple Awk script will parse ascii export file from
# registry HKLM\Software\Oracle and prepare SET statements
# that you can use in your RunBatch.cmd.
# You can export Oracle Key from registry with REGEDT32. Type
# regedt32 in Start -> Run then open
# HKEY_LOCAL_MACHINE\Software\Oracle and export the key with
# menu option "Registry" -> "Save Subtree As" to a file
# orareg.txt.
# Important! Don't use "Save Key" because this option creates
# binary file which could not be parsed with this script!
# How to run this script?
# Save this script to parsereg.awk file and run command:
# C:> gawk -f parsereg.awk orareg.txt > RunFrm.bat
# Open RunBatch.bat with Notepad and remove any lines that you
# KNOW are unnecessary.
# Important! If your application just disapper after you run
# your RunBatch.bat then you removed some line that is
# necessary for runtime -- be carefull while editing
# RunFrm.bat!
###########################################################
BEGIN {FIELDWIDTHS="19 200";OFS=""}
{if (substr($1,1,5)=="Value") {
   getline var_name
   getline var_type
   getline var_value
   print "SET ", substr(var_name,20,20),"=",substr(var_value,20,length(var_value)-19)
 }
}
###########################################################
```

APPENDIX B

```
Rem ##################################################
Rem RUNFRM.BAT by Aleš Kavšek
Rem ----------------------------------------------------------------------------------------------------
Rem Sample batch file for running Developer 6i applications - in our case it's Forms
Rem application.
Rem First draft of this batch file was prepared with script from Appendix A and then edited
Rem with notepad (see bold lines).
Rem Also notice that I replaced all absolute drive letters (C: in my case) with variable %1.
Rem Common mistake that you can make is to remove some lines that you think are not
Rem necessary causing your runtime to simply disappear immediately after start. If you're
Rem not sure leave all variables intact!
Rem ----------------------------------------------------------------------------------------------------
Rem Remember one important principle of running (launching) forms/reports from batch
Rem file like this one: environment variables (SET var=variable) always take precedence
Rem over values in registry, however if value for some variable is not set within batch file
Rem then value from registry is used. Common mistake is to skip TNS_ADMIN variable in
Rem batch file -- if such variable is present in registry on client machine you can run into
Rem ORA-12 error. This is why I explicitly included TNS_ADMIN at the end of batch file!
Rem Some other interesting variables to look after are NLS_LANG and NLS_DATE_FORMAT.
Rem ----------------------------------------------------------------------------------------------------
Rem How to run your application from this batch file?
Rem
Rem You can run your application from command line or shortcut:
Rem C:> RunFrm D:            ...if your copy of runtime is on D: drive.
Rem ##################################################

SET API=%1\ORANT\DBS
SET BROWSER60=%1\ORANT\BROWSE60
SET CA_GPREFS=%1\ORANT
SET CA_UPREFS=%1\ORANT
SET COMPANY_NAME=Your Company
SET CORE40=%1\ORANT\CORE40
SET DE60=%1\ORANT\TOOLS\COMMON60
SET DEV2000_PROJECT=NONE
SET DEV2000_SCCONVERT=FALSE
SET DEV2000_SCVIEWER=NOTEPAD.EXE
SET DEV2000_SOURCECONTROL=NONE
SET DOC60=%1\ORANT\TOOLS\DOC60
SET FORMS60=%1\ORANT\FORMS60
SET FORMS60_PATH=%1\MYAPPDIR
SET GRAPHBMP60=%1\ORANT\GRAPH60\GWIZ_BMP
SET GRAPHICS60=%1\ORANT\GRAPH60
SET GRAPHOGD60=%1\ORANT\GRAPH60\GWIZ_OGD
SET GSCONNECTDIR=%1\ORANT\olap\ecf620
SET GSINSTALLDIR=%1\ORANT\olap\ece620
SET GWIZARD60=%1\ORANT\GRAPH60
SET MM60=%1\ORANT\TOOLS\COMMON60
SET NET80=%1\ORANT\NET80
SET NLS_LANG=SLOVENIAN_SLOVENIA.EE8MSWIN1250
SET NLS_DATE_FORMAT=DD.MM.RRRR
SET NLSRTL33=%1\ORANT\NLSRTL33
SET OCL60=%1\ORANT\GRAPH60
SET OIN60=%1\ORANT\OIN60
```

```
SET ORA_NLS33=%1\ORANT\NLSRTL33\DATA
SET ORACLE_GROUP_NAME=Oracle for Windows NT
SET ORACLE_HOME=%1\ORANT
SET ORACLE_HOME_NAME=DEFAULT_HOME
SET ORACLE_JDK=%1\ORANT\JDK\
SET ORAINFONAV_CDPATH=I:\INSTALL
SET ORAINFONAV_DOCPATH=%1\ORANT\OIN
SET OTRACE80=%1\ORANT\OTRACE80
SET PLSQL80=%1\ORANT\PLSQL80
SET PRO80=%1\ORANT\PRO80
SET QT_PLAYER=OMO
SET RDBMS80=%1\ORANT\RDBMS80
SET REPORTS60_CLASSPATH=%1\ORANT\jdk\lib\classes.zip;
                        %1\ORANT\REPORT60\java\jars\myreports60.jar;
                        %1\ORANT\REPORT60\java\jars\xmlparser.jar
SET REPORTS60_EXPRESS_UI=%1\ORANT\bin
SET REPORTS60_JNI_LIB=%1\ORANT\jdk\bin\javai.dll
SET REPORTS60_PATH=%1\ORANT\REPORT60\ADMIN\TEMPLATE\US;
                   %1\ORANT\REPORT60
SET REPORTS60_SPLASH=TRUE
SET REPORTS60_TMP=%1\ORANT\REPORT60\TMP
SET REPORTS_BUILDER=rwbld60.exe
SET REPORTS_RUNTIME=rwrun60.exe
SET RW60=%1\ORANT\REPORT60
SET SHARED_ORACLE_HOME=%1\ORANT
SET TK60=%1\ORANT\TOOLS\COMMON60
SET UI60=%1\ORANT\TOOLS\COMMON60
SET VGS60=%1\ORANT\TOOLS\COMMON60
SET TNS_ADMIN==%1\ORANT\NET80\Admin
PATH=%1\ORANT\BIN;%PATH%
START %1\ORANT\BIN\ifrun60  MyFormsAPP.fmx
```