



СООБЩЕНИЯ  
ОБЪЕДИНЕННОГО  
ИНСТИТУТА  
ЯДЕРНЫХ  
ИССЛЕДОВАНИЙ

Дубна

00-286

E11-2000-286

V.I.Puzynin, C.Eck\*, B.Flockhard\*, T.Osborne\*,  
V.N.Samoilov

LICENSE MONITORING PROJECT  
FOR COMPUTING ENGINEERING  
APPLICATIONS IN CERN  
Wrapping Procedure for NICE Environment

---

\*CERN, Geneva, Switzerland

2000

## Introduction

Computing for Engineering (CE) group<sup>1</sup> has been installed and supports for NICE<sup>2</sup> and UNIX architectures a large number of applications, currently up to 70, used for administrative, engineering and scientific tasks. Such packages include computer aided design tools for mechanical and electronic engineering, field calculations, mathematical and structural analysis. These tools serve for designing the CERN accelerators, buildings, and their infrastructure as well as the experimental detectors, which these days consist of literally tens of millions of components. Some applications have been installed with the different versions and, as the commercial products, have the different license schemes for the license protection. As the result, CE administrators had:

- The occasional panic about license expiration.
- Inconsistent monitoring schemes (and some nonexistent or at least not accessible) for different licensing technologies.

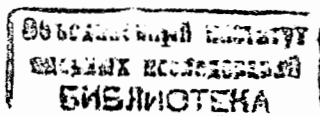
In order to make it possible to provide central services and version control, we have to know the following information:

- Who are our users (serious or casual)?
- Who is holding onto licenses for an excessive time (license hogging)?
- Do we have enough or too many licenses?
- Who is using different version of each application?
- When can we pull the plug on the old version?

The information obtained by collecting statistic information about all CE applications is called *license monitoring*. The tools described here cover the wrapping process for NICE applications under Microsoft Windows NT/95/2000 platforms. This

<sup>1</sup> [http://wwwinfo.cern.ch/ce/ce\\_structure.html](http://wwwinfo.cern.ch/ce/ce_structure.html) - Computing Support for Engineering

<sup>2</sup> <http://nicewww.cern.ch/homepage/home.htm> - NICE Home Page



process is not trivial and standard as for UNIX platform and has some specific features and problems.

## The Choice of the Monitoring Software

The technology used for the license server and associated license monitoring comes from Globetrotter<sup>3</sup>. Its license management system, Flexlm, is in fact becoming the de facto standard in the software industry for the type of products that CERN's engineering community requires. Their main product was called Flexlm, and has been recently changed to SAMwrap. The same company also offers SAMReport, the tool to analyze and provide reports from the log files generated by the different Flexlm daemons. SAMReport also provides graphical reports.

CE group has a redundant configuration of three dedicated Solaris based on license servers. License serving continues if at least two of three machines are up and running. These license servers named *licman1*, *licman2* and *licman3* store the Flexlm log files.

## Wrapping procedure

On the NICE local computers the applications are run from a special application server mapped as Z: disk to a local PC.

For wrapper procedure on the NICE the SAMwrap license server is installed on *sam\_srv1* PC under Windows NT Workstation. The server has the same license agreement as UNIX versions installed on *licmani* ( $i=1,2,3$ ) computers (see figure 1) and allow to wrap the applications and generate the license file only. For license control and report log generation the license information about Windows application from the license file *license.dat* should be copied to the correspondence license files on *licmani* ( $i=1,2,3$ ) servers.

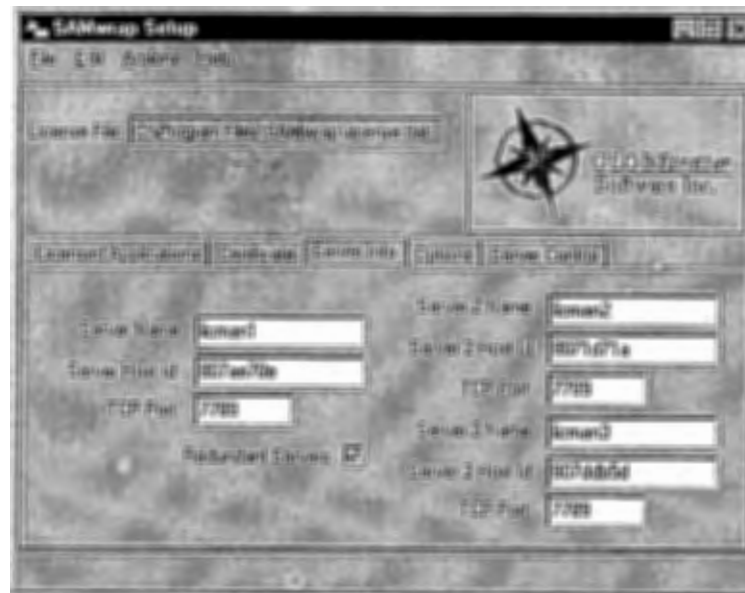


Figure 1. SAMwrap setup configuration.

The user interface for the wrapping procedure is relatively simple and is shown in figure 2. The interface allows the license administrator creating the newly wrapped application (see figure 3), modify the current license information for wrapped applications, or delete an application from the license file.

<sup>3</sup> <http://www.globetrotter.com/products.shtml> – GLOBETrotter Software Inc.



Figure 2. Wrapping User Interface.

All applications are wrapped in a "failsafe" mode (uncheck "Deny loading application on license failure" box in figure 3). In the case where the wrapper can contact the license server, but all licenses are in use (checked out), a specification of failsafe mode allows the wrapped application to run without a license.

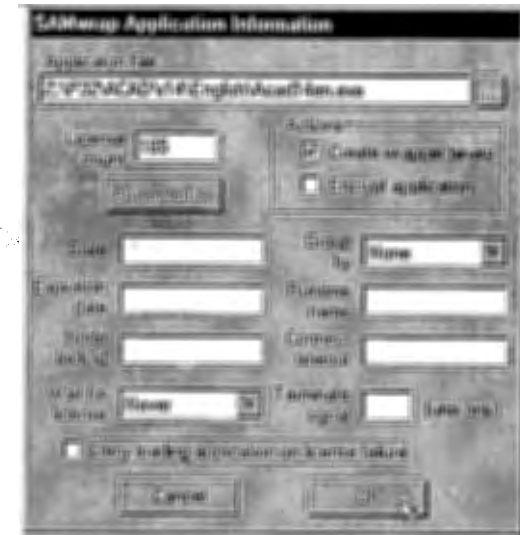


Figure 3. New wrapping of the application.

The results of the wrapping procedure are the following:

- Moving the original executable file `<binary>.exe` to `<binary>.fwr`.
- Generating the wrapper binary file with the name of the original executable file `<binary>.exe` and making a symbolic link from `<binary>.exe` to original `<binary>.fwr`.
- Creating `Swrapper.lf` file in the same directory as `<binary>.exe` file.
- Generating a FEATURE line for the wrapping application, and adding this line to the SAMwrap license file `license.dat`.

`Swrapper.lf` file is a plain text file containing the path to the license server. By default, the file is generated with the path to the preferred license server `licman1` only (see figure 1). If the license server is down or cannot be contacted via network, there

will be a problem and the wrapped application will not run. To solve this problem, the path to three license servers *licmani* ( $i=1,2,3$ ) was added. The individual server names are separated by a semi-colon (see figure 4).

```

Swrapper.lf - Notepad
File Edit Search Help
SWRAP_LICENSE_FILE=7789@licman1;7789@licman2;7789@licman3;
z:\p32\acad\r14\English\License.lic
  
```

Figure 4. *Swrapper.lf* file.

When this is done, the wrapper will attempt to connect to the available license server in the defined order. If the server is not found, the timeout period will be approximately 5 seconds, so in case the server is not available the increased delay for the application user will be minimal. If there was a network problem, or catastrophic server failure, there would still be a problem, as the wrapped application would still refuse to run. To get round this difficulty, it is possible to specify that the wrapper should read the license information from a *License.lic* file which is held on a NICE server (see figure 5). *License.lic* is the part of *license.dat* file and it consists of the license information for a specific application. The *License.lic* file is normally in the directory where the wrapped binary resides.

```

License.lic - Notepad
File Edit Search Help
#####
#
#           SAMwrap License File
#
#####
#           AUTOMATICALLY GENERATED. DO NOT EDIT.
#####
#SERVER licman1 807ae70e 7789
#SERVER licman2 8071d71a 7789
#SERVER licman3 807ddb5d 7789
DAEMON swrap "/opt/lngr/daemon/swrap" "/opt/Flexadm/Folders/fwrap/swrap.opt"
INCREMENT SAMwrap swrap 1.000 1-jan-0 1000 CCE8305DF1E7DB9E6E82 \
ISSUER="Globetrotter Software, Inc." \
VENDOR_STRING="licensed to CERN ECP Division"
FEATURE acad14en swrap 1.000 1-jan-00 165 6C387B9CB94263923F63 \
user_info="Z:\P32\ACAD\r14\English\Acad14en.exe" \
w_binary="z:\p32\acad\r14\english\acad.exe" \
VENDOR_STRING=failsafe
  
```

Figure 5. *License.lic* file.

In NICE environment from "Start" menu on a local PC, a user runs the wrapper binary application *<binary>.exe*. The application first requests a license from a license server. The license server collects in its log file some information about the application (FEATURE name, start time) and user (his current login name, name of PC). If the license is available (it is always in a "failsafe" mode) the wrapper binary will load the original executable *<binary>.fwr*. When the application is terminated the license server fixes the statistical information.

Experience has shown that wrapping is not a trivial procedure. For some applications it turns out that a complex and individual approach is required what is not visible from the outset.

### Problems and solutions

License server recognizes the requests from the wrapped applications by their FEATURE names (see the second position in FEATURE line on figure 5). The name

should be original for each application as identification. The name is generated in wrapper procedure automatically and has exactly the same name as original executable file *<binary>.exe*. Any modifications of FEATURE name after the wrapper procedure are impossible.

One of the main difficulties was how to handle different software versions when each one had the same name. As an illustration, AutoCAD<sup>4</sup> is actually installed on the NICE servers as 8 independent applications – AutoCAD 13 English and French language versions, AutoCAD 14, AutoCAD 14 Mechanical and AutoCAD 14 Mechanical Desktop also in both English and French language versions. However, the name of executable file installed in different folders is always *Acad.exe*. With the standard wrapping procedure the *Acad* FEATURE name will be used for all executables, and license server will not be able to distinguish between the different versions.

The possibility to rename the binary file as *<binaryrenamed>.exe* and save as *<binaryrenamed>.fwr* in the wrapping procedure does not work for most applications. When the application is installed on a PC running Windows, all information about the application and its associated modules is stored in the system registry database. The registry consists of the names of the main program and its modules, the paths to the installation folders and many other pieces of information. This stored information is much richer for such complex applications as AutoCAD. When the binary files are renamed and corresponding information is not updated in the registry, the application will not work. The modification of the registry is a very delicate task and such manipulation is very dangerous in a large network environment such as NICE.

An elegant solution to the problem was however found. The original executable file *<binary>.exe* was renamed as *<binaryrenamed>.exe*. During the wrapping process, the renamed binary was saved using the previous name as *<binary>.exe* (not as *<binaryrenamed>.fwr*). For AutoCAD 14 English the result is shown in figure 5:

- The FEATURE name is *acad14en*. The name is unique.
- Wrapped file in *user\_info* string is *acad14en.exe*.
- Original binary in *w\_binary* string is *acad.exe*.

<sup>4</sup> <http://www.autodesk.com> – AUTODESK Home Page.

This approach allowed the normal installation structure being kept on the application servers and no modifications were required to the software on the users PCs'. After the wrapping procedure, only 3 files - *<binaryrenamed>.exe*, *Swrapper.lf* and *License.lic* were added to the installation folders on the NICE Z: application disk.

The second example of individual approach is that of the wrapping for ANSYS<sup>5</sup> under Windows. The application has a single executable file and choice of the license possibilities – University, Research, Multi-physics or Mechanical. Each option offers a different set of technical possibilities within the Ansys application, and also a different number of licenses. Under Windows, the versions produce different entries in the system registry (see figure 6).

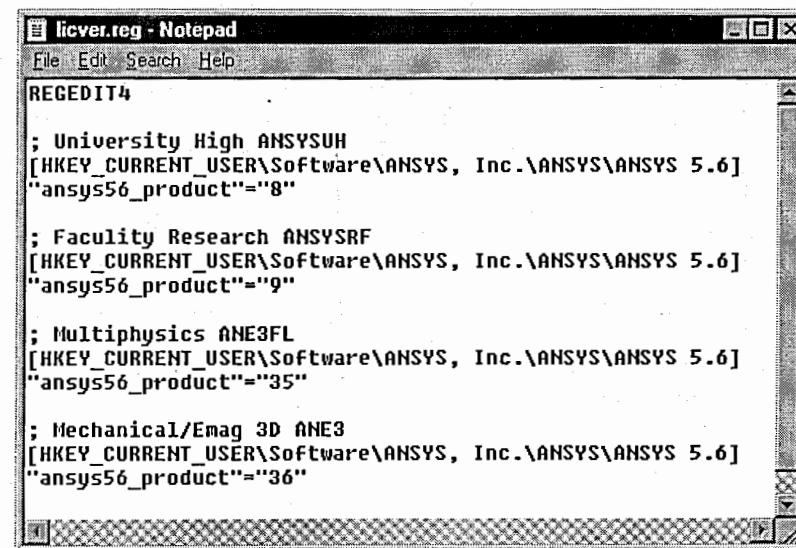


Figure 6. ANSYS settings in the registry.

<sup>5</sup> <http://www.ansys.com> - ANSYS, Inc. Home Page.

Since the license statistics on these four versions was required, there must be 4 original FEATURE names in the license file after the wrapping procedure.

To solve this problem 4 identical C programs were created. The programs modified the corresponding "ansys56\_product" value in the Windows registry and after this they called the ANSYS executable file as external process (see appendix A). The standard wrapping procedure was then carried out on these programs having got different names.

The approach is very perspective for wrapping of UNIX applications, where the choice of the versions is organized by modifications of the parameters in the command string.

The next example of individual approach is that of the wrapping for MATHEMATICA 4<sup>6</sup>. The application has a multi-module structure. The main executable program loads another module as external processes with the same name and after this the main program is terminated. As the result, after the wrapping procedure of the main program only the license server had incorrect information that the application was used for a very short time, approximately 6 sec. To solve this problem all executable modules were wrapped with the same FEATURE name.

## Conclusion

Today the wrapping procedure and corresponding license monitoring cover up to 40 applications in NICE environment (Windows NT/95 platforms). Less then 5% of these applications have been wrapped with standard SAMwrap tools without special approaches described above.

The license monitoring system has been in production for several months without any problems.

## Appendix A

```
/*
The program is for executing of the ANSYS 5.6 University
license version as the external processes.
The program will modify the corresponding registry
```

<sup>6</sup> <http://www.wolfram.com> - WOLFRAM Research Home Page.

```
; University High ANSYSUH
[HKEY_CURRENT_USER\Software\ANSYS, Inc.\ANSYS\ANSYS 5.6]
"ansys56_product"="8"
```

```
and will run the general ANSYS executable program
z:\p32\Ansys\v5.6\bin\intel\ansysir.exe
```

```
Created 13 June 2000, Victor Puzynin IT/CE CERN.
```

```
*/
```

```
#include <windows.h>
```

```
int main()
{
```

```
char *oscmdline;
SECURITY_ATTRIBUTES sa;
STARTUPINFO si;
PROCESS_INFORMATION pi;
HKEY nsiskey;
char RegistryRoot[] = "Software\\ANSYS, Inc.\\ANSYS\\ANSYS 5.6";
char *value;
DWORD vallen;
DWORD ERRNO = 0;
BOOL ok = TRUE;
```

```
value = "8\0x00"; // University High
vallen = sizeof(value) + 1;
```

```
// Modification of the registry
if (RegOpenKeyEx(HKEY_CURRENT_USER, RegistryRoot, 0, KEY_ALL_ACCESS,
&nsiskey) == ERROR_SUCCESS)
{
if ( RegSetValueEx(nsiskey, "ansys56_product", 0, REG_SZ, (BYTE
*)value, vallen) != ERROR_SUCCESS )
{
ERRNO = GetLastError();
}
}
RegCloseKey(nsiskey);
```

```
// Execution of the external process
oscmdline = "z:\p32\Ansys\v5.6\bin\intel\ansysir.exe";
sa.bInheritHandle = TRUE;
```

```
sa.nLength = sizeof(SEcurity_ATTRIBUTES);
sa.lpSecurityDescriptor = NULL;

memset((void *)&si, 0, sizeof(STARTUPINFO));
si.cb = sizeof(STARTUPINFO);
si.dwFlags = STARTF_USESHOWWINDOW;
si.wShowWindow = SW_HIDE;

ok = CreateProcess(NULL, oscmdline, &sa, &sa, TRUE, 0, NULL, NULL,
&si, &pi);
if (!ok)
{
ERRNO = GetLastError();
}
WaitForSingleObject(pi.hProcess, INFINITE);
ERRNO = GetLastError();
CloseHandle(pi.hThread);
CloseHandle(pi.hProcess);
return 0;
}
```

Received by Publishing Department  
on November 23, 2000.