

## How to automate the deployment of a collection of scripts using Ixanos

Nowadays, many applications consist of several discrete files that need to be modified to fit to the environment of deployment. This means that several strings within these files need to be changed. These can be absolute locations (directories) or volumes or application names or connection strings etc. To deploy such an application on different environments can be a critical task and may require a person with application knowledge to guarantee success. Ixanos, is targeting this difficulty by storing a meta-file describing all the actions that need to take place. Further on, it can execute this meta-file and deploy the application on any environment automatically. This guarantees successful deployment every time and without the need of an expert.

To see this in praxis, let us assume the following example: We have a Java application that requires the following:

- a. Compilation of source java files
- b. Prepare an ant-build file to control the build-process
- c. Generate JavaDoc files and store them in specified directories
- d. Build of a jar archive corresponding to the application executable
- e. Copy certain files in specific locations to be used by the obfuscator
- f. Obfuscate the compiled classes
- g. Call an installer application to generate a setup executable
- h. Back up source files
- i. Back up Jar files

All these tasks have strong dependencies on the location of the required applications and will cause plenty of problems when porting them on a different environment. Reason being the fact that nobody guarantees that in the new environment the location of the ant.exe or java.exe or javac.exe or javaDoc, or obfuscator or ... will be in the same location. This means that you would have to go and change all these locations in all the files involved in the above process. As you understand, this is not a simple task. Using Ixanos it can be the click of a button only! For the above example we would do the following steps:

- a. Check all parameters in all involved files that need to be changed when going to an arbitrary different environment. For example, if you find in one of your scripts the following: c:\programms\sun\java\bin\java.exe ..... change it to [[&xxxx]] say to [[&JAVA\_EXE]]. Ixanos recognizes all strings in the form [[&xxxx]] as bind variables and will attempt to replace them with proper values.
- b. Decide upon a directory where you wish to copy the prepared scripts to. Our suggestion is to concentrate them in a single tree structure:



- c. Use the Ixanos wizard to create the meta-file.

d. Build the application and test the process

The meta-file is basically a text file in XML-format. We directly present it, as we believe that this way we facilitate the learning procedure:


```
<yx_application_metafile>
<!-- IXANOS Build File -->
<yx_root Name="C:\__test" />
<yx_tree>
<!-- IXANOS Build-File Target Directories -->
<yx_dire Target="archive" Source="C:\yxknowdb\archive_YXML" LookForBinds="true" />
<yx_dire Target="archivelight" Source="C:\yxknowdb\archivelight_YXML" LookForBinds="true" />
<!-- Deploy Options -->
<yx_file Target="C:\yx\proguard4.2\lib\yx.pro" Source="C:\yxknowdb\archive_YXML\yx.pro" LookForBinds="true" />
<yx_file Target="C:\yx\proguard4.2\lib\yxLight.pro" Source="C:\yxknowdb\archivelight_YXML\yxLight.pro" LookForBinds="true" />
</yx_tree>
<!-- BIND VARIABLES -->
<yx_bind Name="[[ $rootjar]]" Value="C:\yx\_yx\install\yx_archiver_sa\yx_archiver.jar" />
<yx_bind Name="[[ $rootobfuscjar]]" Value="C:\yx\_yx\install\yx_archiver_sa\yx_archiver2.jar" />
<yx_bind Name="[[ $rootjarlight]]" Value="C:\yx\_yx\install\yx_archiver_sa\yx_archiver_light.jar" />
<yx_bind Name="[[ $rootobfuscjarlight]]" Value="C:\yx\_yx\install\yx_archiver_sa\yx_archiver2_light.jar" />
<yx_bind Name="[[ $rtjar]]" Value="C:\yx\yxknowdb\ixanos\jre\lib\rt.jar" />
<yx_bind Name="[[ $avalonjar]]" Value="C:\yx\yxknowdb\ixanos\jre\lib\ext\avalon.jar" />
<yx_bind Name="[[ $src2]]" Value="C:\yx\_yx\programming\projects\java\yx_archiver_sa" />
<yx_bind Name="[[ $src]]" Value="C:\yx\_yx\programming\projects\java\yx_archiver_sa\src" />
<yx_bind Name="[[ $srcright]]" Value="C:\yx\_yx\programming\projects\java\yx_archiver_sa\srcright" />
<yx_bind Name="[[ $srcbackup]]" Value="C:\yx\_progs\ixanos_src_backup\src" />
<yx_bind Name="[[ $srcbackuplight]]" Value="C:\yx\_progs\ixanos_src_backup\srcright" />
<yx_bind Name="[[ $fullinstaller]]" Value="C:\yx\_progs\ixanos_installers\Ixanos_v10_setup.exe" />
<yx_bind Name="[[ $fullinstallernojre]]" Value="C:\yx\_progs\ixanos_installers\Ixanos_v10_setup_noJRE.exe" />
<yx_bind Name="[[ $lightinstaller]]" Value="C:\yx\_progs\ixanos_installers\IxanosLight_v10_setup.exe" />
<yx_bind Name="[[ $lightinstallernojre]]" Value="C:\yx\_progs\ixanos_installers\IxanosLight_v10_setup_noJRE.exe" />
<yx_bind Name="[[ $drive]]" Value="C:" />
<yx_bind Name="[[ $root]]" Value="C:\bbb" />
<yx_bind Name="[[ $locofincludes]]" Value="C:\yx\_yx\install\yx_archiver_sa" />
<yx_bind Name="[[ $locofjre]]" Value="C:\Program Files\Java\jre6" />
<yx_bind Name="[[ $locofant]]" Value="C:\yx\apache-ant-1.7.0" />
<yx_bind Name="[[ $locofnsis]]" Value="C:\Program Files\NSIS\makensis.exe" />
<yx_bind Name="[[ $locofobfuscator]]" Value="C:\yx\proguard4.2\lib" />
<yx_bind Name="[[ $locofixanosdir]]" Value="C:\yx\yxknowdb\ixanos" />
<yx_bind Name="[[ $locofixanosjar]]" Value="C:\yx\yxknowdb\ixanos\yx_archiver.jar" />
</yx_application_metafile>
```

The meta file above is sufficient for Ixanos to deploy this application. It contains the following information:

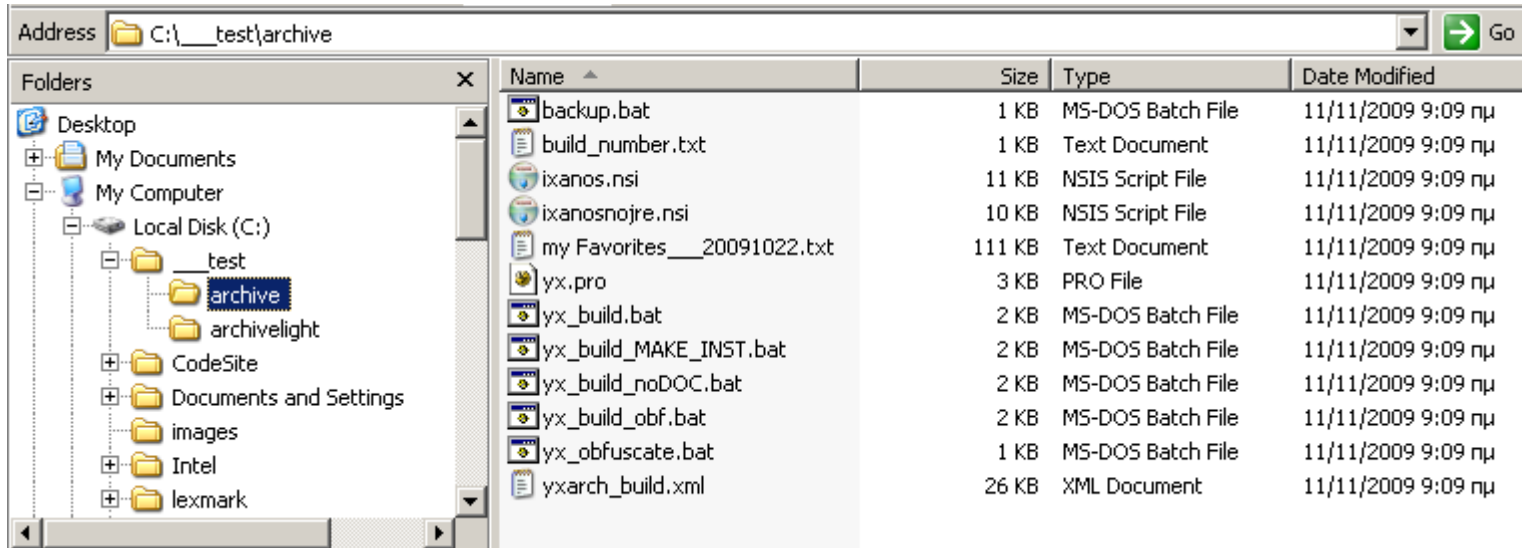
a. The root directory where the application will be deployed ("C:\\_\_test")

- b. To copy all files from the directory "C:\yxknowdb\archive\_YXML" to the directory "C:\\_\_test\ archive"
- c. To copy all files from the directory "C:\yxknowdb\archivelight\_YXML" to the directory "C:\\_\_test\ archivelight"
- d. To copy the file "C:\yxknowdb\archive\_YXML\yx.pro" to "C:\yx\proguard4.2\lib\yx.pro"
- e. To copy the file "C:\yxknowdb\archive\_YXML\yxLight.pro" to "C:\yx\proguard4.2\lib\yxLight.pro"
- f. Then it tells Ixanos that all files directly under the directory "C:\yxknowdb\archive\_YXML" and all its sub-directories shall have their bind variables replaced.  
(LookForBinds="true")
- g. It also tells Ixanos that all files directly under the directory "C:\yxknowdb\ archivelight \_YXML" and all its sub-directories shall have their bind variables replaced.  
(LookForBinds="true")
- h. It also tells Ixanos that the file "C:\yxknowdb\archive\_YXML\yx.pro" and "C:\yxknowdb\archive\_YXML\yxLight.pro" shall have their bind variables replaced  
(LookForBinds="true")
- i. It also tells Ixanos that wherever a bind variable is found (ie: something in the form **[[ \$xxxxx ]]** it should be replaced according to the following map:

"[[ \$rootjar ]]"	=>	"C:\yx\_yx\install\yx_archiver_sa\yx_archiver.jar"
"[[ \$rootobfuscjar ]]"	=>	"C:\yx\_yx\install\yx_archiver_sa\yx_archiver2.jar"
"[[ \$rootjarlight ]]"	=>	"C:\yx\_yx\install\yx_archiver_sa\yx_archiver_light.jar"
"[[ \$rootobfuscjarlight ]]"	=>	"C:\yx\_yx\install\yx_archiver_sa\yx_archiver2_light.jar"
"[[ \$rtjar ]]"	=>	"C:\yx\yxknowdb\ixanos\jre\lib\rt.jar"
"[[ \$avalonjar ]]"	=>	"C:\yx\yxknowdb\ixanos\jre\lib\ext\avalon.jar"
"[[ \$src2 ]]"	=>	"C:\yx\_yx\programming\projects\java\yx_archiver_sa"
"[[ \$src ]]"	=>	"C:\yx\_yx\programming\projects\java\yx_archiver_sa\src"
"[[ \$srclight ]]"	=>	"C:\yx\_yx\programming\projects\java\yx_archiver_sa\srclight"
"[[ \$srcbackup ]]"	=>	"C:\yx\_progs\ixanos_src_backup\src"
"[[ \$srcbackuplight ]]"	=>	"C:\yx\_progs\ixanos_src_backup\srclight"
"[[ \$fullinstaller ]]"	=>	"C:\yx\_progs\ixanos_installers\Ixanos_v10_setup.exe"
"[[ \$fullinstallernojre ]]"	=>	"C:\yx\_progs\ixanos_installers\Ixanos_v10_setup_noJRE.exe"
"[[ \$lightinstaller ]]"	=>	"C:\yx\_progs\ixanos_installers\IxanosLight_v10_setup.exe"
"[[ \$lightinstallernojre ]]"	=>	"C:\yx\_progs\ixanos_installers\IxanosLight_v10_setup_noJRE.exe"
"[[ \$drive ]]"	=>	"C:"
"[[ \$root ]]"	=>	"C:\bbb"
"[[ \$locofincludes ]]"	=>	"C:\yx\_yx\install\yx_archiver_sa"
"[[ \$locofjre ]]"	=>	"C:\Program Files\Java\jre6"
"[[ \$locofant ]]"	=>	"C:\yx\apache-ant-1.7.0"
"[[ \$locofnsis ]]"	=>	"C:\Program Files\NSIS\makensis.exe"
"[[ \$locofobfuscator ]]"	=>	"C:\yx\proguard4.2\lib"
"[[ \$locofixanosdir ]]"	=>	"C:\yx\yxknowdb\ixanos"

When this is in place you can run the meta-file (use this button ) and see the results:

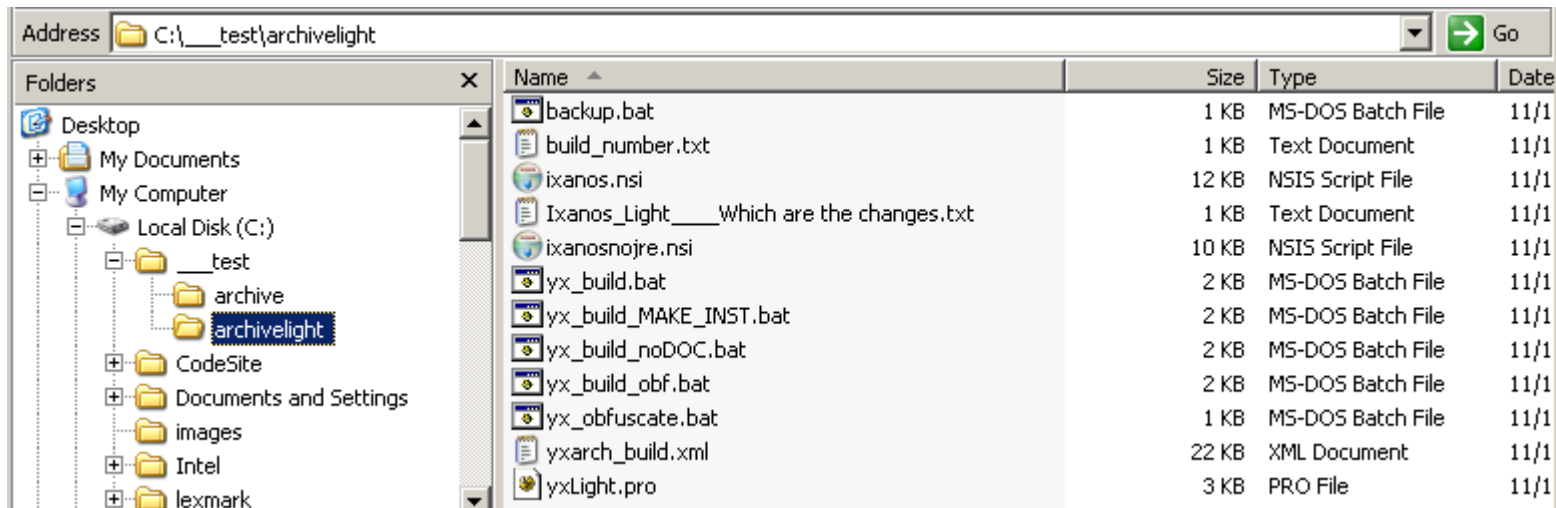
Two directory roots have been created. First is:



The screenshot shows a Windows Explorer window with the address bar set to `C:\__test\archive`. The left pane shows the folder tree with `archive` selected. The right pane displays a list of files with columns for Name, Size, Type, and Date Modified.

Name	Size	Type	Date Modified
backup.bat	1 KB	MS-DOS Batch File	11/11/2009 9:09 AM
build_number.txt	1 KB	Text Document	11/11/2009 9:09 AM
ixanos.nsi	11 KB	NSIS Script File	11/11/2009 9:09 AM
ixanosnojre.nsi	10 KB	NSIS Script File	11/11/2009 9:09 AM
my Favorites__20091022.txt	111 KB	Text Document	11/11/2009 9:09 AM
yx.pro	3 KB	PRO File	11/11/2009 9:09 AM
yx_build.bat	2 KB	MS-DOS Batch File	11/11/2009 9:09 AM
yx_build_MAKE_INST.bat	2 KB	MS-DOS Batch File	11/11/2009 9:09 AM
yx_build_noDOC.bat	2 KB	MS-DOS Batch File	11/11/2009 9:09 AM
yx_build_obf.bat	2 KB	MS-DOS Batch File	11/11/2009 9:09 AM
yx_obfuscate.bat	1 KB	MS-DOS Batch File	11/11/2009 9:09 AM
yxarch_build.xml	26 KB	XML Document	11/11/2009 9:09 AM

And Second is:



The screenshot shows a Windows Explorer window with the address bar set to `C:\__test\archivelight`. The left pane shows the folder tree with `archivelight` selected. The right pane displays a list of files with columns for Name, Size, Type, and Date.

Name	Size	Type	Date
backup.bat	1 KB	MS-DOS Batch File	11/1
build_number.txt	1 KB	Text Document	11/1
ixanos.nsi	12 KB	NSIS Script File	11/1
Ixanos_Light__Which are the changes.txt	1 KB	Text Document	11/1
ixanosnojre.nsi	10 KB	NSIS Script File	11/1
yx_build.bat	2 KB	MS-DOS Batch File	11/1
yx_build_MAKE_INST.bat	2 KB	MS-DOS Batch File	11/1
yx_build_noDOC.bat	2 KB	MS-DOS Batch File	11/1
yx_build_obf.bat	2 KB	MS-DOS Batch File	11/1
yx_obfuscate.bat	1 KB	MS-DOS Batch File	11/1
yxarch_build.xml	22 KB	XML Document	11/1
yxLight.pro	3 KB	PRO File	11/1

If you open each file you can see that the corresponding bind variables have been replaced by Ixanos.

To summarize, in case you wish to deploy the above meta-file on any other environment you only need to change the bind variables to reflect the directories of the new environment and hit on execute to generate the new build files. That's it. Note that you only change the directories in a single file, namely, the Ixanos meta-file.

The [ixanos.com](http://ixanos.com) team.