

How to make a software project ready for external backup – rules to be followed

Felician ALECU, PhD, University Lecturer

Department of Economic Informatics

Academy of Economic Studies, Bucharest, Romania

E-mail: [alecu\[at\]ase\[dot\]ro](mailto:alecu[at]ase[dot]ro); Web Page: <http://alecu.ase.ro>

Abstract: *The purpose of this article is to explain the way in which a software project (old, current or future) should be tailored in order to be ready for an external backup.*

Keywords: *external backup, global repository.*

Introduction

All the modifications described here are actually related to project structure rather than project content that will remain the same but divided into several parts hosted in different locations by using various tools.

Related to the way in which the files are located and managed, the software projects of our days are subscribing to one of the following patterns (Table 1):

- REGULAR – the entire content is stored in a single repository, like VSS, SVN, CVS and so on;
- MIXED – source files are stored in several repositories, like a

project that is using VSS for C source files, documents and so on and CVS for the JAVA part;

- MIXED + DOCS – a MIXED type project that is using a dedicated location for the documents, like a separate CVS repository where the documents are stored;

Project Type	Repository Location for ...		
	Source Files TypeA	Source Files TypeB	Documents
REGULAR	RepositoryX	RepositoryX	RepositoryX
MIXED	RepositoryX	RepositoryY	RepositoryX/Y
MIXED+DOCS	RepositoryX	RepositoryY	RepositoryZ

Table 1 – Types of software projects

Rules

The rules to be used to modify the structure of the projects in order to make them ready for external backup are detailed below.

- RULE #1: Sensitive data
 - ⊗ Any sensitive data that should not be copied on the data center servers must be removed from the current repository (repositories) and should be placed in some dedicated locations outside the backup process area.
 - ⊗ Example: ProjectX files, located in a common VSS repository called projectXSpecialVer (together with many other projects, like ProjectY, ProjectZ, ProjectT, ...), should be moved in a separate VSS database named projectXSpecialVer_X. The same should be applicable for

CVS, SVN and any other related tool.

- RULE #2: Documents (RULE #1 should be already met), not applicable for projects of MIXED+DOCS type that will use the existing repository
 - ⊗ The project documents should be located in a separate repository, so these files will not be mixed anymore with any other types, like sources, scripts and so on. For such cases, a global repository for docs will be used, called GLOBAL_DOCS. Inside this repository, the files will be stored at project level, in a structure similar to the one described below (Figure 1). In such a way, all the documents will be located at the same location, so it should become easier to locate details related to some of the software projects.

- Rule #3: Structure Updates (RULES #1 and #2 should be already met)
 - ⊗ Each time a project structure is modified, the changes must be documented in the table below (Table 2). The table will be located on a SharePoint location so it will be accessible by browsing

- Rule #4: Synchronization (RULES #1, #2, #3 should be already met)
 - ⊗ Only the repositories marked as containing SOURCE FILES will be synchronized to data centers, all the others will remain for internal use.

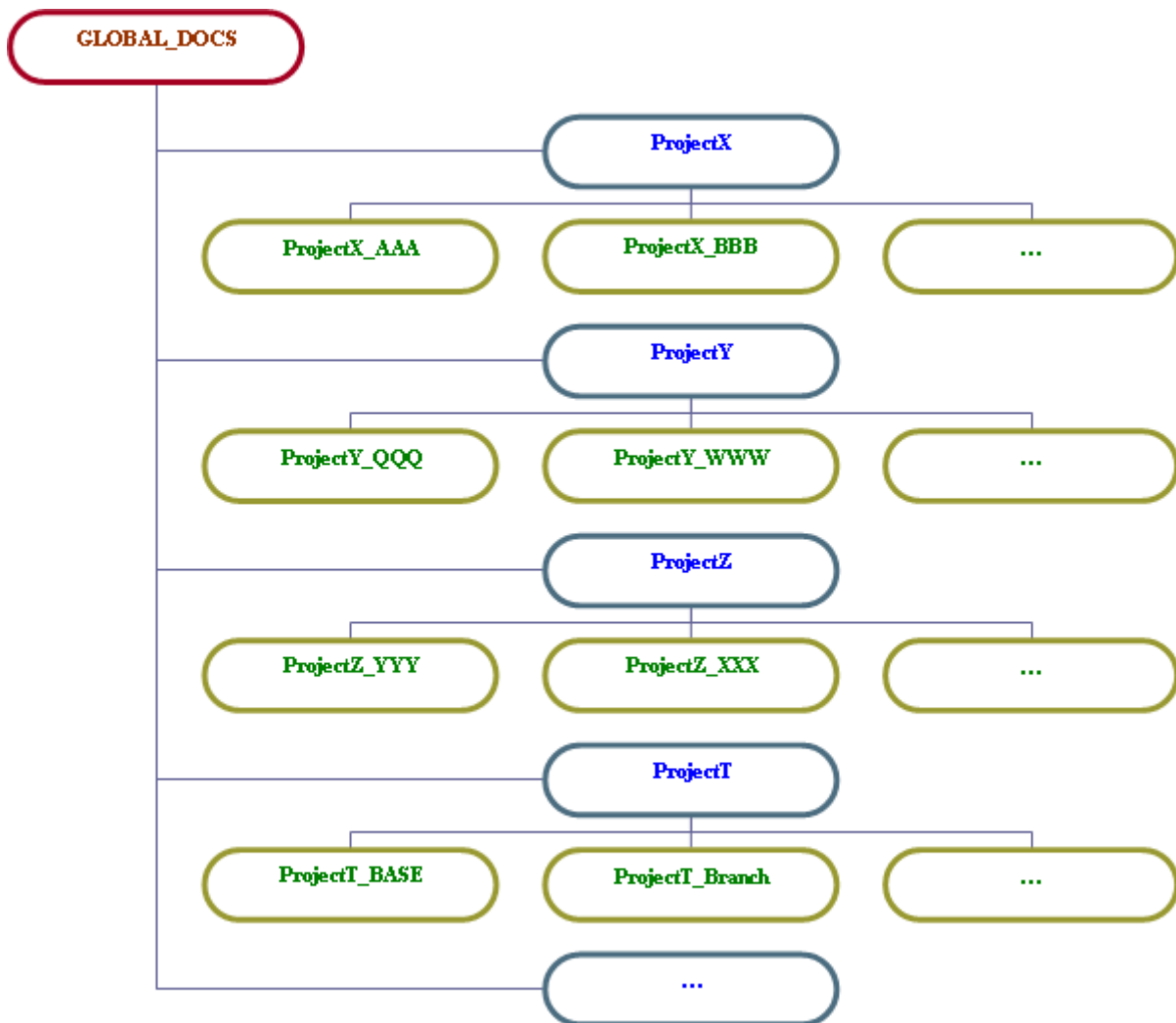


Figure 1 – GLOBAL_DOCS repository structure

- Rule #5: New Projects
 - ⊗ Each time a new project is started, at least the following actions should be performed:
 - ✦ define the details (name, type, location) of the repositories used by the project (source files, scripts and so on, except the documents);

- ✦ always store the documents completely separated by the source files, so a new entry will be added to the existing GLOBAL_DOCS repository;
- ✦ fill in the project details in the table below (Table 2) so anyone will find out the details of the repositories involved
 - ⊗ Project Name and Details, like
 - ◇ Description
 - ◇ Content
 - ◇ Name
 - ◇ Type
 - ◇ Location
 - ◇ Comments

Table 2 – Repository Locations of Software Projects

Project Name	Repository Details					
	Description	Content	Name	Type	Location	Comments
ProjectX						
	C Files	Source Files	ProjectX_Rep	VSS	VSS_ProjectX_Database	
	Java Files	Source Files	ProjectX_Rep	CVS	/cvsrep/ProjectX	Do not backup
	Python Files	Source Files	ProjectX_Rep	SVN	/svnrep/ProjectX	
	Test Cases Docs	Documentation	GLOBAL_DOCS	CVS	/cvsrep/GLOBAL_DOCS	Public, do not backup
	Documents	Documentation	GLOBAL_DOCS	CVS	/cvsrep/GLOBAL_DOCS	Public, do not backup

- ✦ always place sensitive data in a totally separated repository, so there will not be any sensitive data inside the sources and documents storage areas;
- ✦ make sure all the previous rules (RULES #1, #2, #3, #4) are completely followed.

Conclusions

The purpose of the rules is to guarantee there will be no sensitive data and documents inside the repositories that are copied on the servers of the external partners or data centers. These rules should also be followed when a new project arrives and its structure is about to be defined.

References

- [1] <http://en.wikipedia.org/wiki/Repository>
- [2] http://en.wikipedia.org/wiki/Software_repository
- [3] http://en.wikipedia.org/wiki/Concurrent_Versions_System
- [4] http://en.wikipedia.org/wiki/Apache_Subversion
- [5] http://en.wikipedia.org/wiki/Microsoft_Visual_SourceSafe