DEXPRO Invoice: Archivierung für unterschiedliche Mandanten

```
/* ================================================================================== */
/** Archiving is always custom designed!
 * This function is called for all technical archiving actions ("Archiving", "ArchivingDirect", ...)
 * and for user defined action "Disqualify".  You should call this function with archiving rights.
 * Use "context.changeScriptUser()" to change script user context and do not forget to switch back context before "return" statement!
 * @param {string} type Archiving type you can use to address different archives ("Job" / "Direct" / "Disqualify").
 * @param {DocFile} docFile DocFile object.
 * @returns {boolean} true (archived) / false (error).
 **/
function ue_Archiving(type, docFile){
    /* Switch user context */
    var currentUser = context.currentUser;
    context.changeScriptUser("job");
    var view = '';

    var template = docFile.getAttribute("FromTemplate");
    switch(template){
        case "Mailroom":
        case "Invoice":
            if (docFile.Principal === "Company Name 00") {
                view = "Unit=Default/Instance=Default/View=Company00";
            } else if (docFile.Principal === "Company Name 01") {
                view = "Unit=Default/Instance=Default/View=Company01";
            } else if (docFile.Principal === "Company Name 02") {
                view = "Unit=Default/Instance=Default/View=Company02";
            } else {
                docFile.insertStatusEntry("ErrArchiving", "Can't find
Principal");
                docFile.sync();
                return false;
            }
            var schema = "Unit=Default/Instance=Default/DocumentSchema=Invoice";

            var ad = new ArchivingDescription();
            ad.targetView = view;
            ad.targetSchema = schema;
            ad.archiveServer = "EEx"; // since Documents 4.0 using multi archive
server
            ad.archiveMonitor = true;
            ad.archiveStatus = true;
            ad.addRegister("Documents");

            if( docFile.archive(ad) ){
                context.changeScriptUser(currentUser);
                var archiveKey = docFile.getArchiveKey();
                if( archiveKey==="" || archiveKey===false ){
                    docFile.insertStatusEntry("ErrArchiving", "Missing archive
key");
                    docFile.sync();
                    return false;
                }
                return true;
            }
            else{
                docFile.insertStatusEntry("ErrArchiving",
docFile.getLastError());
                docFile.sync();
            }
            break;
        default:
            docFile.insertStatusEntry("UndefinedFileTemplate", template);
            docFile.sync();
            break;
    }
    context.changeScriptUser(currentUser);
    return false;
}
```

Revision #2
Created 19 April 2021 16:33:50 by Simon Pust
Updated 18 October 2021 12:27:55 by Phillip Langer