Versteckte User-Exits Mappe

An den Mappentypen sind Skripte für bestimmte Aktionen hinterlegt. Einige der Skripte werden komplett unverschlüsselt als User-Exit-Skripte herausgegeben. Die verschlüsselten Skripte enthalten dafür User-Exits. Die verschlüsselten Skripte importieren auch die spezifischen Modul-Libs sowie die Modul-Costom-Libs.

> Es bietet sich an die UserExits in den spezifischen User-Exit-Libs für das entsprechende Modul hinzuzufügen. Für Invoice wäre dies zum Beispiel die "**Invoice__UserExit_CustomLib**".

BeforeEdit

```
/** Das UserExit wird vor der Bearbeitung einer Mappe direkt zu Beginn im Skript aufgerufen.
 * Bei einem Fehler muss "context.errorMessage" in der Funktion gesetzt werden!
 * @return {boolean} true / false.
**/
function ue_BeforeEdit_Start(){
//context.errorMessage = "...";
    return true;
}

/** Das UserExit wird vor der Bearbeitung einer Mappe am Ende des Skripts aufgerufen.
 * Bei einem Fehler muss "context.errorMessage" in der Funktion gesetzt werden!
 * @return {boolean} true / false.
**/
function ue_BeforeEdit_End(){
//context.errorMessage = "...";
    return true;
}
```

OnSave

```
/** Das UserExit wird beim Speichern einer Mappe direkt zu Beginn im Skript aufgerufen.
 * Bei einem Fehler muss "context.errorMessage" in der Funktion gesetzt werden!
 * @return {boolean} true / false.
**/
```

```
function ue_OnSave_Start(){
	//context.errorMessage = "...";
    return true;
}

/** Das UserExit wird beim Speichern einer Mappe am Ende des Skripts aufgerufen.
 * Bei einem Fehler muss "context.errorMessage" in der Funktion gesetzt werden!
 * @return {boolean} true / false.
**/
function ue_OnSave_End(){
	//context.errorMessage = "...";
    return true;
}

/** Es gibt ein extra UserExit für Gentable-Operationen.
 * Der Vorteil hiervon ist, dass das Gentable nicht noch einmal ausgelesen und geschrieben werden muss.
 * Das erfolgt automatisch im Anschluss.
 * Bei einem Fehler muss in "this.Error" die Fehlermeldung geschrieben werden und "this.Error" muss auf false gesetzt werden!
 * @return Kein Rückgabewert erforderlich!
**/
Gentable.prototype.ue_InvoiceOnSave = function(){
	this.Error  = "";
	this.Result = true;
}
Gentable.prototype.ue_ProcurementOnSave = function(){
	this.Error  = "";
	this.Result = true;
}
```

DecreaseFieldRights

```
/** Im DecreaseFieldRightsOnFileView-Skript wird beim Durchlauf der Felder für jedes Feld dieses User Exit ausgeführt
 * @param {string} fieldName Aktueller technischer Feldname
 * @param {number} e Nummer für enumval
 * @param {Log} log Optionales Logging-Objekt (nur wenn das Logging aktiv ist)
**/
function ue_ChangeFieldRights(fieldName, e, log){
```

```
enumval[e] = "-rw";
}
```

Aufzählungs-Skripte Invoice

Auf einigen Aufzählungs-Feldern liegen Aufzählungs-Skripte und diese enthalten ebenfalls User-Exit-Funktionen zur Manipulation der Werte.

```
/** Bei der Neu-Erstellung des PropCaches für den Mandant können die Daten manipuliert werden (propCache.InvoicePrincipal).
 * Feld: Principal
 * @param {*} principalObject Eingehendes Objekt
 * @param {string} template Modul (Invoice/Mailroom/Procurement)
 * @param {Log} log Logging Objekt
 * @returns {*} Manipuliertes Objekt
**/
function ue_AdjustEnumPrincipals(principalObject, template, log){
return principalObject;
}
function ue_AdjustEnumPrincipalsPropCache_Invoice(principalObject, log){
return principalObject;
}
/** Am Ende des Skripts kann das enumval für den Mandanten manipuliert werden.
 * Feld: Principal
 * @param {*} enumval Eingehendes enumval
 * @returns {*} Manipuliertes enumval
**/
function ue_Adjust_Invoice_Enum_Principals(enumval){
return enumval;
}

/** Bei der Neu-Erstellung des PropCaches für den Buchungskreis können die Daten manipuliert werden (propCache.InvoicePrincipal).
 * Feld: CompanyCode
 * @param {*} companyCodeObject Eingehendes Objekt
 * @param {string} template Modul (Invoice/Mailroom/Procurement)
 * @param {Log} log Logging Objekt
 * @returns {*} Manipuliertes Objekt
**/
function ue_AdjustEnumCompanyCodes(companyCodeObject, template, log){
```

```
return companyCodeObject;
}
function ue_AdjustEnumPrincipalsPropCache_Invoice(companyCodeObject, log){
return companyCodeObject;
}
/** Am Ende des Skripts kann das enumval für den Buchungskreis manipuliert werden.
 * Feld: CompanyCode
 * @param {*} enumval Eingehendes enumval
 * @returns {*} Manipuliertes enumval
**/
function ue_Adjust_Invoice_Enum_CompanyCodes(enumval){
return enumval;
}

/** Über die Funktion können neben Rechnung und Gutschrift weitere Rechnungstypen definiert
werden.
 * Feld: InvoiceCreditVoucher
 * Hierfür kann direkt auf das enumal zugegriffen werden.
 * @returns {*} Kein Rückgabewert!
**/
function ue_AddInvoiceTypes(){
enumval.push("myinvtype;pf:MyInvoiceType");
}

/** Über die Funktion können zusätzliche Zahlstatus definiert werden.
 * Feld: PaymentStatus
 * Hierfür kann direkt auf das enumal zugegriffen werden.
 * @returns {*} Kein Rückgabewert!
**/
function ue_AddPaymentStatus(){
enumval.push("mypaytype;pf:MyPaymentType");
}
```

Beim Aussteuern

```
/** Das User-Exit wird direkt zu Beginn im Skript "DEXPRO_Action_Disqualify" ausgeführt.
 * @param {Log} log Logging-Objekt
 * @param {string} comment Pflicht-Kommentar des Benutzers
 * @returns {boolean} true / false (Ausgabe Fehlermeldung)
**/
function ue_Disqualify_Start(log, comment){
```

```
    // context.errorMessage = "";
return true;
}

/** Das User-Exit archiviert den Vorgang.
 * @param {string} type Bei ausgesteuerten Belegen immer "Disqualify"
 * @param {DocFile} docFile Auszusteuerndes DocFile-Objekt
 * @param {Log} log Logging-Objekt
 * @returns {boolean} true (Archiviert) / false (Ausgabe Fehlermeldung)
**/
function ue_Archiving(type, docFile, log){
    // context.errorMessage = "";
return true;
}

/** Das User-Exit wird nach erfolgreicher Archivierung ausgeführt.
 * @param {Log} log Logging-Objekt
 * @param {string} comment Pflicht-Kommentar des Benutzers
 * @returns {boolean} true / false (Ausgabe Fehlermeldung)
**/
function ue_Disqualify_AfterArchiving(log, comment){
    // context.errorMessage = "";
return true;
}

/** Das User-Exit wird nach erfolgreichem Löschen der Mappe ausgeführt.
 * @param {Log} log Logging-Objekt
 * @param {string} comment Pflicht-Kommentar des Benutzers
 * @returns {boolean} true / false (Ausgabe Fehlermeldung)
**/
function ue_Disqualify_AfterDelete(log, comment){
    // context.errorMessage = "";
return true;
}
```

Revision #7
Created 2022-08-05 10:55:09 UTC by Markus Meisner
Updated 2022-08-05 12:56:55 UTC by Markus Meisner