

# Suche einschränken

Ein Wirtschaftsprüfer soll zum Beispiel eine eingeschränkte Suchmöglichkeit über das aktuelle Wirtschaftsjahr haben. Hierfür kann die Suchmaske manipuliert werden und die Volltextsuche kann deaktiviert werden.

- [FillSearchMaskScript](#)
- [OnSearchScript](#)

# FillSearchMaskScript

Über die globale Eigenschaft "FillSearchMaskScript" kann ein Skript eingebunden werden, über welches die Suchmaske manipuliert werden kann. Bei der Einschränkung eines Datumfeldes muss berücksichtigt werden, dass in der Suche 2 Felder "von" und "bis" eingeblendet werden. Technisch müssen diese Felder über die technischen Präfixe "SOH" bzw. "STX" ergänzt werden. Das sieht in Notepad++ zum Beispiel wie folgt aus:

```
Client 4: DEX: SOHPostingDate.  
Client 4: DEX: STXPostingDate.
```

Das folgende Skript holt sich die Abfrage-Parameter und iteriert die Felder. Bei den Feldern für das Rechnungsdatum werden die Datumsfelder vorbelegt und das Feld selber erhält einen Schreibschutz.

```
/** User exit on search.  
 * set: DlcGlobalOptions FillSearchMaskScript  
 * @author DEXPRO MM  
 * @since 10.08.2022  
 **/  
  
var user    = context.getSystemUser();  
var params = context.getQueryParams();  
  
var cnt = 0;  
const count = params.searchFieldCount;  
for( ; cnt<count; ++cnt ){  
    var searchField = params.getSearchField(cnt, false);  
    if( searchField.name==="InvoiceDate" ){ // Feldname anpassen  
        searchField.setDefault("01.01.2022", true);  
        var filter = searchField.name + searchField.compOp + searchField.valueExpr;  
        util.out("[ "+cnt+" ] [SOH] InvoiceDate.setDefault(01.01.2022, true) new filter: " +  
filter);  
    }  
    else if( searchField.name==="InvoiceDate" ){ // Feldname anpassen  
        searchField.setDefault("31.12.2022", true);  
        var filter = searchField.name + searchField.compOp + searchField.valueExpr;  
        util.out("[ "+cnt+" ] [STX] InvoiceDate.setDefault(31.12.2022, true) new filter: " +  
filter);  
    }  
}
```

```

else{
    //var filter = searchField.name + searchField.compOp + searchField.valueExpr;
    //util.out("[ Cant+] - " + (searchField.name).toString() + " fullfilter: " +
filter);
}
}
//util.out(context.scriptName + " #[" + count+] " + context.event + " -- " + user.login);

```

```

11  for( ; cnt<count; ++cnt ){
12      var searchField = params.getSearchField(cnt, false);
13      if( searchField.name=="SOH InvoiceDate" ){
14          searchField.setDefault("01.01.2022", true);
15          var filter = searchField.name + searchField.compOp + searchField.valueExpr;
16          util.out("[ Cant+] [SOH]InvoiceDate.setDefault(01.01.2022, true) new filter: " + filter);
17      }
18      else if( searchField.name=="STX InvoiceDate" ){
19          searchField.setDefault("31.12.2022", true);
20          var filter = searchField.name + searchField.compOp + searchField.valueExpr;
21          util.out("[ Cant+] [STX]InvoiceDate.setDefault(31.12.2022, true) new filter: " + filter);
22      }
23      else{
24          //var filter = searchField.name + searchField.compOp + searchField.valueExpr;
25          //util.out("[ Cant+] - " + (searchField.name).toString() + " fullfilter: " + filter);
26      }
27  }

```

# OnSearchScript

Über den Documents-Manager müssen 2 Eigenschaften "**OnSearchScript**" und "**FillSearchMaskScript**" hinzugefügt werden. Für die Eigenschaften werden 2 UserExit-Skripte "**DEXPRO\_UserExit\_OnSearch**" und "**DEXPRO\_UserExit\_FillSearchMask**" mit ausgeliefert. Die Eigenschaften werden im Standard nicht mehr gesetzt, da das OnSearch-Skript auch bei jedem **FileResultset** ausgeführt wird!

OnSearchScript	DEXPRO_UserExit_OnSearch
FillSearchMaskScript	DEXPRO_UserExit_FillSearchMask

## Anwendungsfall: Wirtschaftsprüfer

Ein Wirtschaftsprüfer soll zum Beispiel nur Zugriff auf Belege zu einem fest definierten Zeitraum erhalten. Über das "**FillSearchMaskScript**" kann das Belegdatum automatisch in der Suchmaske vorbelegt und mit einem Schreibschutz versehen werden. Datumswerte haben allerdings immer eine 'von'- und 'bis'-Angabe und hier wird jeweils ein Sonderzeichen (**SOH/STX**) vor den Feldnamen gesetzt. Diese Sonderzeichen führen beim XML-Import des Skripts zu einem Fehler. Daher muss das Skript manuell angelegt werden und der Skriptinhalt muss manuell kopiert werden.

```
88 var searchField = qParams.getSearchField(j, false);
89 switch( searchField.name ){
90     case "SOHDateOfReceipt":
91         searchField.setDefault(parseDateStrToLang(paramDateOfReceipt.Value2Specific, lang), true);
92         break;
93     case "STXDateOfReceipt":
94         searchField.setDefault(parseDateStrToLang(paramDateOfReceipt.Value3Specific, lang), true);
95         break;
96 }
```

```
/** User exit beim Öffnen der Suchmaske zur Vorbelegung von Such-Feldern.
 * Das Skript verwendet die Parameter:
 * - SearchRestriction_InvoiceDate
 * - SearchRestriction_DateOfReceipt
 * DlcGlobalOptions: FillSearchMaskScript
 * @author DEXPRO MM
 * @since 18.10.2022
 */
// #import "DEXPRO__ImportLib";

/** Wandelt das Datum in das passende Datumsformat zur Anmelde-Sprache des Benutzers.
 * @param {string} strDate Eingehender Datums-String im Format 'dd.mm.yyyy'
 * @param {string} lang Anmelde-Sprache des aktuellen Benutzers ("de"/"en"/...)
```

```

/**/
function parseDateStrToLang(strDate, lang){
    var useD = new Date();
    try{
        var d = util.convertStringToDate(strDate, "dd. mm. yyyy");
        if( d instanceof Date ){
            useD = d;
        }
    }
    catch(e){ return e;}
    var dFormat = "dd. mm. yyyy";
    for( var l=1; l<=6; l++ ){
        var locale = context.getGlobalAttribute("Locale"+l+". Locale");
        if( locale===lang ){
            context.setClientLang("en");
            dFormat =
context.getGlobalAttribute("Locale"+l+". DateFormat"). toLowerCase();
            context.setClientLang(lang);
        }
    }
    try{
        var dStr = util.convertDateToString(useD, dFormat);
    }
    catch(e){ return e; }
    return dStr;
}

/** Entfernt die Custom-Eigenschaft am Benutzer, welche die Suchmasken-Einstellungen
speichert.
* @param {SystemUser} usr Aktuell angemeldeter Benutzer
* @returns {boolean} true / false
**/
function removeDefaultExtendedSearch(usr){
    var itProp = usr.getCustomProperties("defaultExtendedSearchQuery", "");
    for( var prop of itProp ){
        if( prop.name==="defaultExtendedSearchQuery" ){
            prop.deleteCustomProperty( true);
        }
    }
    return true;
}

```

```

}

/* MAIN */
var qParams = context.getQueryParams();
var count   = qParams.searchFieldCount;
var user     = context.getSystemUser();
var lang     = context.getClientLang();

var paramInvDate = new Param("SearchRestriction_InvoiceDate", "Invoice");
if( paramInvDate.Result===true ){
    var ap_InvDate = paramInvDate.Value1Specific;
    if( context.findAccessProfile(ap_InvDate) ){
        if( user.hasAccessProfile(ap_InvDate) ){
            removeDefaultExtendedSearch(user);
            for( var i=0; i<count; i++ ){
                var searchField = qParams.getSearchField(i, false);
                switch( searchField.name ){
                    case "InvoiceDate":

searchField.setDefault(parseDateStrToLang(paramInvDate.Value2Specific, lang),
true);

                        break;
                    case "InvoiceDate":

searchField.setDefault(parseDateStrToLang(paramInvDate.Value3Specific, lang),
true);

                        break;
                }
            }
        }
    }
}

var paramDateOfReceipt = new Param("SearchRestriction_DateOfReceipt", "Invoice");
if( paramDateOfReceipt.Result===true ){
    var ap_DateOfReceipt = paramDateOfReceipt.Value1Specific;
    if( context.findAccessProfile(ap_DateOfReceipt) ){
        if( user.hasAccessProfile(ap_DateOfReceipt) ){
            removeDefaultExtendedSearch(user);
            for( var j=0; j<count; j++ ){

```

```

        var searchField = qParams.getSearchField(j, false);
        switch( searchField.name ){
            case "DateOfReceipt":

searchField.setDefault(parseDateStrToLang( paramDateOfReceipt. Value2Specific, lang),
true);

                break;
            case "DateOfReceipt":

searchField.setDefault(parseDateStrToLang( paramDateOfReceipt. Value3Specific, lang),
true);

                break;
        }
    }
}
}
}
}

```

Allerdings hat der Anwender immer noch die Option über die Volltextsuche zu suchen. Hierzu kann wiederum das "**OnSearchScript**" helfen. Die Volltextsuche wird verwendet, wenn die Suche nur ein Feld beinhaltet.

```

/** User exit bei der Ausführung einer Suche.
 * Benutzer einer definierten Gruppe dürfen nicht die einfache Volltext-Suche verwenden.
 * Die Gruppe wird über Parameter ermittelt und die Volltextsuche wird über die Anzahl der
Suchfelder ermittelt.
 * DlcGlobalOptions: OnSearchScript
 * @author DEXPRO MM
 * @since 10.08.2022
**/
// #import "DEXPRO__ImportLib";
var params = context.getQueryParams();
var count = params.searchFieldCount;
var type = params.requestType;
var mask = params.searchMaskName;
var user = context.getSystemUser();

if( type===1 || type=="1" ){
    var paramInvDate = new Param("SearchRestriction_InvoiceDate", "Invoice");
    if( paramInvDate.Result===true ){
        var apInvDate = paramInvDate.Value1Specific;

```

```

        if( context.findAccessProfile( apInvDate) ){
            if( user.hasAccessProfile( apInvDate) ){
                if( count===1 ){
                    context.errorMessage =
context.getFromSystemTable("Err_UseExtendetSearch") +
"\n";

                                // + "(requestType: "+type+" ) search
mask( "+mask+" ) \n";

                    context.returnValue  = -142;
                }
            }
        }
    }

    var paramDateOfReceipt = new Param("SearchRestriction_InvoiceDate", "Invoice");
    if( paramDateOfReceipt.Result===true ){
        var apDateOfReceipt = paramDateOfReceipt.Value1Specific;
        if( context.findAccessProfile( apDateOfReceipt) ){
            if( user.hasAccessProfile( apDateOfReceipt) ){
                if( count===1 ){
                    context.errorMessage =
context.getFromSystemTable("Err_UseExtendetSearch") +
"\n";

                                // + "(requestType: "+type+" ) search
mask( "+mask+" ) \n";

                    context.returnValue  = -142;
                }
            }
        }
    }
}

```