# Mandanten Protection in the Service Manager® Knowledge Management Module

## How to Implement Mandanten Security in the Knowledge Management Module

HP® Management Software — Service Management

## Introduction

Service Manager® offers a security feature called *Mandanten* for any searches performed within Service Manager. Because the Knowledge Management module uses a third-party search engine (the Verity® K2 search engine), it does not apply the settings for Mandanten protection that may have been defined for Service Manager searches against these tables by the customer. This document shows how to modify the JavaScript® in Service Manager so that you can utilize Mandanten protection for searches executed by the Knowledge Management module to ensure that all searches against these tables comply with the security requirements defined by Mandanten.

## Prerequisites

A license for the Service Manager Knowledge Management module and pre-existing setup of Mandanten security are required.

Knowledge of JavaScript is strongly recommended, as is familiarity with the setup and use of the Service Manager Knowledge Management module.

## Short introduction to Mandanten security

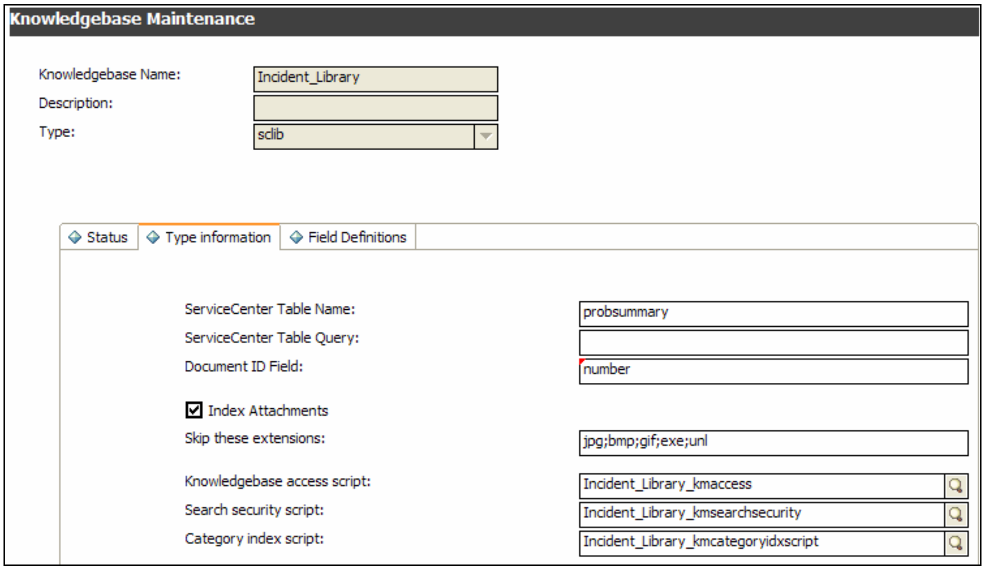
**Note**: Mandanten file security is described in the Service Manager online documentation in the Server Security section.

Typically, Mandanten is set up based on the company of the user who is accessing the system, though it can be set up based on any value in any table that needs to be protected. Mandanten protection is set up on a per-table basis. The operator can be a member of one, many, or no security groups. The security groups (scsecuritygroups table) set values that define which records the user is allowed to see based on the content of the mandant field; the mandant field is set up for each table in the scmandant record. More flexible queries for each table and security group can be added to the scaccess table. However, when a user enters a search anywhere within Service Manager, the Mandanten restrictions are appended to that query upon execution, and restricted records will not be part of the returned record set.

Queries executed outside of Service Manager, such as with the K2 search engine, are not Mandanten protected. Information shown in the Knowledge Management hit list is not yet retrieved from the Service Manager internal files. When you select a record from the hit list for viewing, it will then access the Service Manager internal file (such as probsummary) that is under Mandanten protection. Access to the record will then be denied based on the Mandanten restrictions, even if the record was displayed in the hit list. To prevent this from happening, follow the steps outlined in this document that will modify the search security scripts to read Mandanten settings and apply these settings to the hit list as well.

## Introduction to security in the Knowledge Management module

The Service Manager Knowledge Management module defines a search security script in the Manage Knowledgebase's format:



Knowledgebase Maintenance

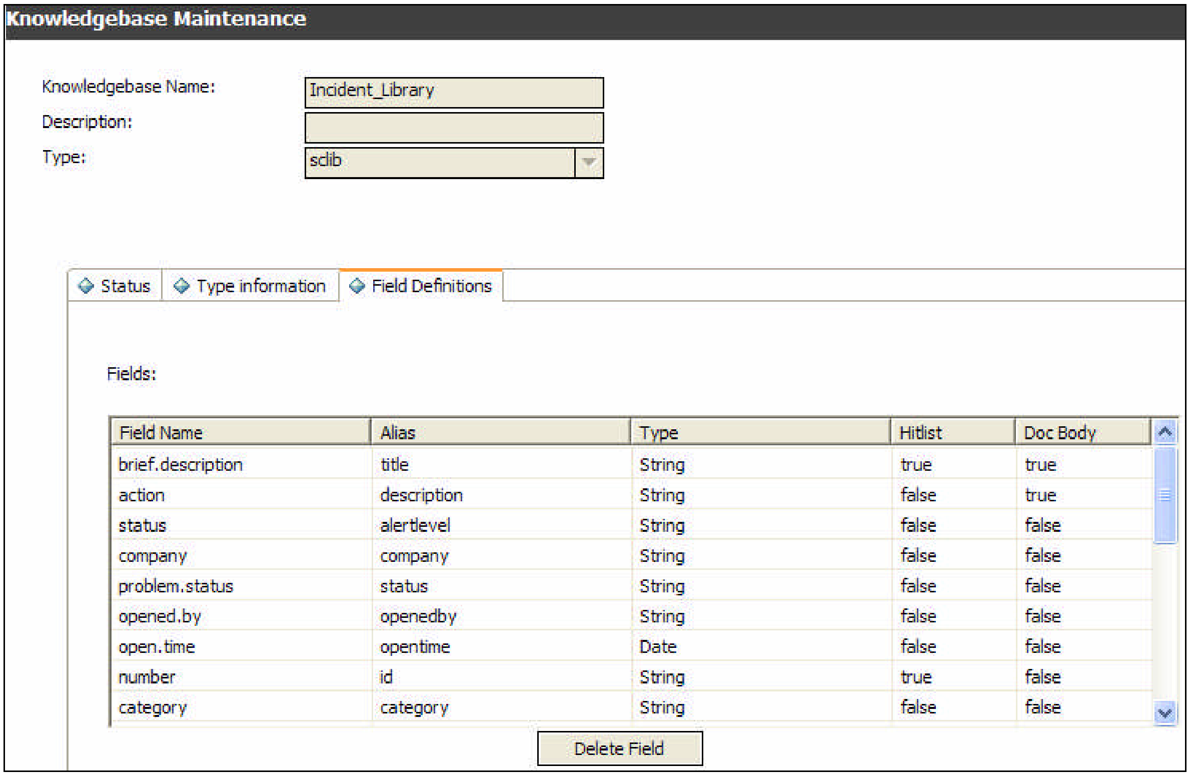
Knowledgebase Name: Incident_Library
Description:
Type: sclib

ServiceCenter Table Name: probsummary
ServiceCenter Table Query:
Document ID Field: number

☑ Index Attachments
Skip these extensions: jpg;bmp;gif;exe;unl

Knowledgebase access script: Incident_Library_kmaccess
Search security script: Incident_Library_kmsearchsecurity
Category index script: Incident_Library_kmcategoryidxscript

With this security script, the query string sent to the search engine can be modified for more restrictive results. Any field entered in the query must be defined on the Field Definitions tab:



Knowledgebase Maintenance

Knowledgebase Name: Incident_Library
Description:
Type: sclib

Fields:

| Field Name | Alias | Type | Hitlist | Doc Body |
|---|---|---|---|---|
| brief.description | title | String | true | true |
| action | description | String | false | true |
| status | alertlevel | String | false | false |
| company | company | String | false | false |
| problem.status | status | String | false | false |
| opened.by | openedby | String | false | false |
| open.time | opentime | Date | false | false |
| number | id | String | true | false |
| category | category | String | false | false |

With these features, it is possible to use Java Script to select the correct security groups, and to transform the Service Manager queries to match the query language used by the search engine so that it indirectly enforces Mandanten security.

One possible use for setting up Mandanten security in the Knowledge Management module is provide company-specific knowledge in a single Knowledge Base, but then segregate the information based on the individual end-user's company.

## Enforcing Mandanten security via the search security script

The following is an example on how to enforce Mandanten security for Incident Management by modifying the Incident_Library_kmsearchsecurity JavaScript in the Service Manager ScriptLibrary. The out-of-box security script is as follows:

```
function getSecurityInfo(user, record)
{
  return "";
}
```

To check for valid mandant field settings, and to modify the query sent to the search engine accordingly, change the script to:

```
// This script returns a string containing a search query
// that filters the search based on the user's rights and
// permissions.


function getSecurityInfo(user, record)
{
   var querystr = "";
   var operatorFile = new SCFile("operator");
   var rc = operatorFile.doSelect("name=\""+user+"\"");

   if ( operatorFile != null )
   {
      var securitygroups = operatorFile.security_group;
      var mandant = new SCFile("scmandant");
      var rc_mandant=mandant.doSelect("filename=\""+"probsummary"+"\"");
      if (rc_mandant == RC_SUCCESS)
      {
         var mandantField=mandant.fieldname;
         querystr = checkSecurityGroup( securitygroups, mandantField);
      }
   }

   var k = querystr.indexOf("<MATCHES>");
   while (k >= 0)
   {
      for (var l=k; querystr.substring(l,l-1)!= ">" && l>0; l--)
      {
      }
      var field_name=querystr.substring(l,k);
      var f = field_name.indexOf(".");
      while (f >=0)
      {
         field_name=field_name.substring(0,f) +
         field_name.substring(f+1,field_name.length);
         var querystr=querystr.substring(0,l) + field_name +
         querystr.substring(l + field_name.length+1, querystr.length);
         var f = field_name.indexOf(".");
      }
      var k = querystr.indexOf("<MATCHES>", k + 10);
   }
   if (querystr == "()")
   { querystr=""; }
   return querystr;
}

function checkSecurityGroup( groups , field )
{
   var secgroup = new SCFile("scsecuritygroup");
   var sm_query = "";
   for ( var i=0;i<groups.length();i++)
   {
      var rc = secgroup.doSelect("security.id=\""+groups[i]+"\"");
      if (rc == RC_SUCCESS)
      {
            for ( var j=0;j<secgroup.include.length();j++)
            {
               if (secgroup.include[j]!=null)
               {
                  sm_query+= "(" + field + " <MATCHES> " +
                  secgroup.include[j] + ")";
                  if (j<secgroup.include.length() - 2)
                  {
                     sm_query += " <OR> "
                  }
               }
            }
            for ( var j=0;j<secgroup.exclude.length();j++)
            {
               if (secgroup.exclude[j]!=null)
               {
                  sm_query+= " <AND> "
                  sm_query+="( <NOT> " + field + " <MATCHES> " +
                  secgroup.exclude[j] +")";
                  if (j<secgroup.exclude.length() - 2)
                  {
                     sm_query += " <OR> "
                  }
               }
            }
      }
   }
   return sm_query;
}
```

The same script modifications will work for the Interaction Library (whose script name is Interaction_Library_kmsearchsecurity), where the following is changed from

```
   var rc_mandant = mandant.doSelect("filename=\""+"probsummary"+"\"");
```

To

```
   var rc_mandant = mandant.doSelect("filename=\""+"incidents"+"\"");
```

In addition, if Mandanten protection has been implemented on the knownerror table according to the documentation, the script modifications will work for the KnownError Library (script name KnownError_Library_kmsearchsecurity) where you change the first line above to:

```
   var rc_mandant = mandant.doSelect("filename=\""+"knownerror"+"\"");
```

Likewise, if the rootcause table has been set up for Mandanten according to the documentation, the script modifications will work for the Problem Library (script name Problem_Library_kmsearchsecurity) where you change the first line above to:

```
   var rc_mandant = mandant.doSelect("filename=\""+"rootcause"+"\"");
```

The script changes shown above apply Mandanten protection using the scmandant settings. Additionally, a limiting query can be entered in the scaccess table. If this is used, the query in the scaccess record will have to be translated from Service Manager query language to the search engine's query language. To do that, enter the JavaScript as shown below. It adds an additional function to the search security script that gets the name of the security group (groups[i]) and the tablename ("probsummary") as parameters. The function then returns the query string translated from Service Manager's query language to Verity's VQL. The translated query then needs to be connected to the VQL query string created from the scmandant settings with an <AND>:

```
// This script returns a string containing a search query
// that filters the search based on the user's rights and
// permissions.

function getSecurityInfo(user, record)
{
   var querystr = "";
   var operatorFile = new SCFile("operator");
   var rc = operatorFile.doSelect("name=\""+user+"\"");

   if ( operatorFile != null )
   {
      var securitygroups = operatorFile.security_group;
      var mandant = new SCFile("scmandant");
      var rc_mandant =
      mandant.doSelect("filename=\""+"probsummary"+"\"");
      if (rc_mandant == RC_SUCCESS)
      {
         var mandantField=mandant.fieldname;

         querystr = checkSecurityGroup( securitygroups, mandantField);
      }
   }

   var k = querystr.indexOf("<MATCHES>");
   while (k >= 0)
   {
      for (var l=k; querystr.substring(l,l-1)!= ">" && l>0; l--)
      {
      }
      var field_name=querystr.substring(l,k);
      var f = field_name.indexOf(".");
      while (f >=0)
      {
         field_name=field_name.substring(0,f) + field_name.substring(f+1,
         field_name.length);
         var querystr=querystr.substring(0,l) + field_name +
         querystr.substring(l + field_name.length+1, querystr.length);
         var f = field_name.indexOf(".");
      }
      var k = querystr.indexOf("<MATCHES>", k + 10);
   }
   if (querystr == "()")
   { querystr=""; }
   return querystr;
}


function checkSecurityGroup( groups , field )
{
   var secgroup = new SCFile("scsecuritygroup");
   var sm_query = "";
   for ( var i=0;i<groups.length();i++)
   {
      var rc = secgroup.doSelect("security.id=\""+groups[i]+"\"");
      if (rc == RC_SUCCESS)
      {
         for ( var j=0;j<secgroup.include.length();j++)
         {
            if (secgroup.include[j]!=null)
            {
               sm_query+= "(" + field + " <MATCHES> " +
               secgroup.include[j] + ")";
               if (j<secgroup.include.length() - 2)
               {
               sm_query += " <OR> "
               }
            }
         }
         for ( var j=0;j<secgroup.exclude.length();j++)
         {
            if (secgroup.exclude[j]!=null)
            {
               sm_query+= " <AND> "
               sm_query+="( <NOT> " + field + " <MATCHES> " +
               secgroup.exclude[j] +")";
               if (j<secgroup.exclude.length() - 2)
               {
                  sm_query += " <OR> "
               }
            }
         }
         sm_query="(" + sm_query + ")" +
         get_scaccess_query(secgroup.security_id, "probsummary")
      }
   }
   return sm_query;
}

function get_scaccess_query(group, tablename)
{
   var scaccess=new SCFile("scaccess");
   var rc=scaccess.doSelect("filename=\"" + tablename +"\"" + "and
   security.id=\""+group+"\"");
   if (rc == RC_SUCCESS)
   {
      var new_query=scaccess.restricting_query;
      var replace=" <MATCHES> ";
      var k = new_query.indexOf("=");
      while (k>= 0)
      {
         new_query=new_query.substring(0,k) + replace +
         new_query.substring(k + 1, new_query.length);
         var k = new_query.indexOf("=", k);
      }

      var replace=" <NOT> ";
      var k=new_query.indexOf("~");
      while (k>=0)
      {
         new_query=new_query.substring(0,k) + new_query.substring(k + 1,
         new_query.length);
         for (var l=k; new_query.substring(l,l-1)!= " "; l--)
         { }
         new_query=new_query.substring(0,l) + replace +
         new_query.substring(l, new_query.length);
         k=new_query.indexOf("~", k+1);
      }

      var k=new_query.indexOf("ISIN");
      while (k>=0)
      {
         new_query="(" + new_query.substring(0,k) + " <MATCHES> " +
         new_query.substring(k + 4, new_query.length);
         for (var l=k-1; new_query.substring(l,l-1)!= " "; l--)
         {
            var field_name=new_query.substring(l-1,k);
         }

         var replace_1 = " <OR> " + field_name + " <MATCHES> ";
         var m=new_query.indexOf(",",k);
         while (m>=0)
         {
            new_query=new_query.substring(0,m) + replace_1 +
            new_query.substring(m + 1, new_query.length) + ")";
            m=new_query.indexOf(",",m+1);
         }
         k=new_query.indexOf("ISIN", k+4);
      }
      var k = new_query.indexOf("{");
      while (k>=0)
      {
         new_query=new_query.substring(0,k) + new_query.substring(k + 1,
         new_query.length);
         k = new_query.indexOf("{",k+1);
      }
      var k = new_query.indexOf("}");
      while (k>=0)
      {
         new_query=new_query.substring(0,k) + new_query.substring(k + 1,
         new_query.length);
         k = new_query.indexOf("}",k+1);
      }


      var k = new_query.indexOf("\"");
      while (k>=0)
      {
         new_query=new_query.substring(0,k) + new_query.substring(k + 1,
         new_query.length);
         k = new_query.indexOf("\"",k+1);
      }

      var k = new_query.indexOf(" and ");
      while (k>=0)
      {
         new_query=new_query.substring(0,k) + " <AND> " +
         new_query.substring(k + 5, new_query.length);
         k = new_query.indexOf(" and ",k+5);
      }

      var k = new_query.indexOf(" or ");
      while (k>=0)
      {
         new_query=new_query.substring(0,k) + " <OR> " +
         new_query.substring(k + 4, new_query.length);
         k = new_query.indexOf(" or ",k+4);
      }

      var k = new_query.indexOf(" <AND> ");
      while (k>=0)
      {
         new_query=new_query.substring(0,k) + " ) <AND> ( " +
         new_query.substring(k + 7, new_query.length);
         k = new_query.indexOf(" <AND> ",k+7);
      }


      new_query=" <AND> (" + new_query + "))";
      if (new_query == "()")
      { new_query=""; }
   }
   else
   {
      new_query="";
   }
return new_query;
}
```

**Note**: This script will translate Service Manager queries containing “and”, “or”, “~” (not) and “ISIN” into the Verity Query Language (VQL). Service Manager functions such as `operator()` or `index()` cannot be translated because they do not have an equivalent in VQL.

Modifications to this script can be made as needed, as long as the query part that needs to be added has a counterpart in VQL.

# Summary

To enable Mandanten security in Knowledge Management, you will have to

1. Set up Mandanten protection according to the documentation:
   - The operator needs to belong to one or many security groups.
   - Security groups must have one or many “include” and/or “exclude” values.
   - The `scmandant` file must have a record for the table to protect and define a field in that table as the mandant field.
2. Ensure that all fields used in the `scmandant` and `scaccess` files are defined in the Manage Knowledge Bases record’s Field definitions tab.
3. Modify the search security script for the library that uses Mandanten protection, as described above

**Note**: Make sure to run the full re-index as an operator without Mandanten limitations, since the Mandanten query that enforces security on the originating table will limit the records read during the re-index operation.

# For more information

Please visit the HP Management Software support Web site at:

http://www.hp.com/managementsoftware/support

This Web site provides contact information and details about the products, services, and support that HP Management Software offers.

HP Management Software online software support provides customer self-solve capabilities.  It provides a fast and efficient way to access interactive technical support tools needed to manage your business.  As a valued support customer, you can benefit by being able to:

- Search for knowledge documents of interest
- Submit and track progress on support cases
- Submit enhancement requests online
- Download software patches
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter discussions with other software customers
- Research and register for software training

**Note:** Most of the support areas require that you register as an HP Passport user and sign in.  Many also require an active support contract.

To find more information about support access levels, go to the following URL:

http://www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to the following URL:

http://www.managementsoftware.hp.com/passport-registration.html

02/2008