

**Einrichtung eines SQL Server
als Index-Datenbank für Quarantänen
in AntiVir Exchange**

Inhalt

EINFÜHRUNG.....	3
Vorteil der im Standard verwendeten Jet-DB.....	3
Nachteil bei Verwendung eines SQL-Servers.....	3
SCHAUBEISPIELE	3
FEHLERBEHANDLUNG BEI QUARANTÄNEN.....	4
Typische Fehler bei SQL Server.....	4
Unterstützung von SQL-Server.....	4
Konsequenzen.....	5
EINRICHTEN EINER SQL-SERVER QUARANTÄNE	6
Einrichten des SQL-Servers.....	6
Anlegen der SQL-Datenbank.....	7
Anlegen des SQL – Benutzer.....	8
Anlegen der Tabellen mit Script.....	9
Berechtigungen des angelegten SQL-Users prüfen.....	10
Anzeige der Berechtigungen.....	11
KONFIGURIEREN DER QUARANTÄNE IN ANTI VIR EXCHANGE	12
Konfigurieren der Datenbank - Verbindung.....	12
Ein Beispiel:.....	13
Die Quarantänen.....	14
Auswahl der Quarantäne im Job Advanced spam filtering.....	16
HIER NOCH EIN KLEINER TIPP FÜR DIE DARSTELLUNG SEHR GROBE QUARANTÄNEN:.....	18

Einführung

In AntiVir Exchange 7 kann als Index-Datenbank für die Quarantäne auch ein **lokal** installierter SQL-Server verwendet werden.

In AntiVir Exchange 7 kann als Index-Datenbank für die Quarantänen auch ein lokal installierter SQL-Server verwendet werden. Während die von AntiVir Exchange normalerweise verwendete Jet-DB in AntiVir Exchange ab 80% von 1GB Dateigröße Warnungen versendet (weil über 1 GB die MDB-Dateien unhandlich werden und dann regelmäßig Probleme hervorrufen), kann mit einem SQL Server eine größere Menge an Index-Daten gehalten werden, also entweder mehr Index-Daten pro Email (Body-Auszug, Job-Reports) oder mehr Emails (d.h. ein längerer Zeitraum).

Vorteil der im Standard verwendeten Jet-DB

Die Jet-DBs sind unschlagbar einfach zu administrieren und sehr stabil.

Der Administrator hat normalerweise überhaupt keine Arbeit damit. AntiVir Exchange legt sie bei Bedarf an, räumt sie auf und kann das DB-Schema bei Versionswechsel automatisch erweitern.

Nachteil bei Verwendung eines SQL-Servers

Beim SQL-Server muss der Admin viele dieser Dinge von Hand machen, was Anwendern ohne Vorkenntnisse mit SQL Server schwer fallen kann. Wir empfehlen unseren Kunden immer zunächst eine Lösung auf der Basis der Jet-DBs zu suchen.

Schaubeispiele

- Für ein einfaches Journal aller von extern eingehenden Emails (Addressfilter-Job, der alles in eine Journal-Quarantäne schreibt, ohne Body-Auszug und Job-Report) passen auf einem Gateway 800.000 Emails in den Index. Da lassen sich die Emails (um die 10.000 Emails pro Tag) monatelang aufheben.
- Sehr häufig sind **SPAM-HIGH** Quarantänen das Problem, da die Spam-Reports sehr lang sind und dadurch nur wenige Emails in den Index passen. Also löscht man diese Emails schon nach z.B. einer Woche. Sollte doch mal ein Empfänger eine Email vermissen, kann man sie immer aus dem Journal erneut senden (siehe vorherigen Punkt).
- **SPAM-MEDIUM** Emails (ebenfalls mit langem Spam-Report) muss man dagegen Länger aufbewahren. Hier ist naturgemäß das Risiko höher, dass sie doch jemand braucht. Vielleicht sind auch Summaries mit Links zum Zugriff auf die Emails konfiguriert. Allerdings liegen nur **sehr** wenige Emails im **SPAM-MEDIUM** Bereich. Im Durchschnitt etwa tausendmal weniger als **SPAM-HIGH**, so dass sich hier das Problem großer Index-DBs nicht stellt.

- Allerdings gibt es natürlich Kunden, die SQL Server bereits einsetzen und sich damit sehr gut auskennen. Solche Kunden kommen natürlich auch mit SQL-Quarantänen gut zu recht, denn meist funktioniert ja alles.

Fehlerbehandlung bei Quarantänen

Prinzipiell gibt es eine Einstellung in AntiVir Exchange an jeder Quarantäne, die "**Mission Critical**" heißt. Diese Einstellung beeinflusst die Reaktion der Jobs auf Fehler beim Versuch, eine Email in die Quarantäne zu stellen. Dies ist nicht spezifisch für SQL Server Quarantänen. Bei solchen Quarantänen führt dies aber leicht zu unerwünschte Auswirkungen, da erfahrungsgemäß hier häufiger Fehler auftreten.

Typische Fehler bei SQL Server

- Der SQL Server Dienst ist nicht gestartet, oder ein anderes administratives Problem verhindert den Zugriff auf die Datenbank (Berechtigungen, Firewall, Locks, Timeout).
- Der Kunde verwendet SQL Express, und die Obergrenze für die Dateigröße der Datenbank ist erreicht. Ohne Vorwarnung funktioniert die Datenbank dann einfach nicht mehr.
- Der Kunde hat den SQL Server nicht lokal auf dem Email-Server, sondern auf einer anderen Maschine, und es gibt Probleme mit dem Netzwerk.

Unterstützung von SQL-Server

Antivir Exchange unterstützt grundsätzlich nur **lokal** auf dem Exchange **installierte** SQL Server für die Quarantänen, um zumindest den dritten Fehlerfall auszuschließen.

Technisch möglich ist es natürlich, den SQL Server auf einem anderen Rechner zu betreiben; das *kann* je nach Einsatzzweck akzeptabel sein.

Ohne "**Mission Critical**" Einstellung der Quarantäne (dies ist der Default) wird ein Job einen Fehler der Quarantäne einfach ignorieren.

Es wird ein Hilferuf per Email an den Administrator gesendet und ein Eintrag ins Eventlog geschrieben. Das ist alles.

Die Email ist eben später nicht in der Quarantäne, was im Extremfall bedeutet, dass die Email verloren ist (etwa wenn die Job-Aktion "in Quarantäne stellen, dann Email löschen" ist).

Im Falle einer virulenten Email wäre das nicht gravierend.

Eine "**Mission Critical**" Quarantäne hingegen wird bei Quarantäne-Fehlern einen Fehler im Job auslösen. Der Job bricht dann seine Verarbeitung ab.

Auch im Job gibt es eine Einstellung "**Mission Critical**", die das weitere Vorgehen bestimmt.

Wenn der Job nicht "**Mission Critical**" ist (und das sind im Default die meisten Jobs, außer dem Virenschanner-Job), dann wird sich der Job bei einer Häufung derartiger Fehler irgendwann selbst abschalten.

Wieder gehen Hilferufe per Email an den Administrator, und es gibt Eventlog-Einträge und derartiges. (Der Job schaltet sich übrigens auch selbst wieder ein, aber eine gewisse Zeitspanne muss der Admin ohne den Job leben können.)

Eine nicht erreichbare Quarantäne schaltet dann also den Job ab. Bei einem Viren-Job kann das gefährlich sein.

Wenn der Job jedoch ebenfalls "**Mission Critical**" ist, dann wird der Quarantäne-Fehler im Job die Verarbeitung der Email ganz abbrechen (Die Logik ist: Ein "Mission Critical" Job muss unbedingt gelaufen sein, sonst geht die Email nicht durch).

Die Email wird dann in die **Badmail**-Quarantäne gestellt (Nebenbei: **Diese Badmail-Quarantäne kann man nicht in den SQL-Server verlegen**). Solange der SQL-Server also nicht erreichbar ist, wird keine Email mehr durchgelassen - alles landet in der Badmail (und kann dort später wieder eingestellt werden).

Eine solche Einstellung ist sehr extrem.

Konsequenzen

Die Konsequenzen einer nicht erreichbaren Quarantäne-DB liegen also zwischen den Extremen "Emails gehen verloren" und "keine Emails kommen mehr an".

Es ist daher wichtig, dass die Quarantänen funktionieren und die lokalen Jet-DBs sind hierbei klare Favoriten.

Einrichten einer SQL-Server Quarantäne

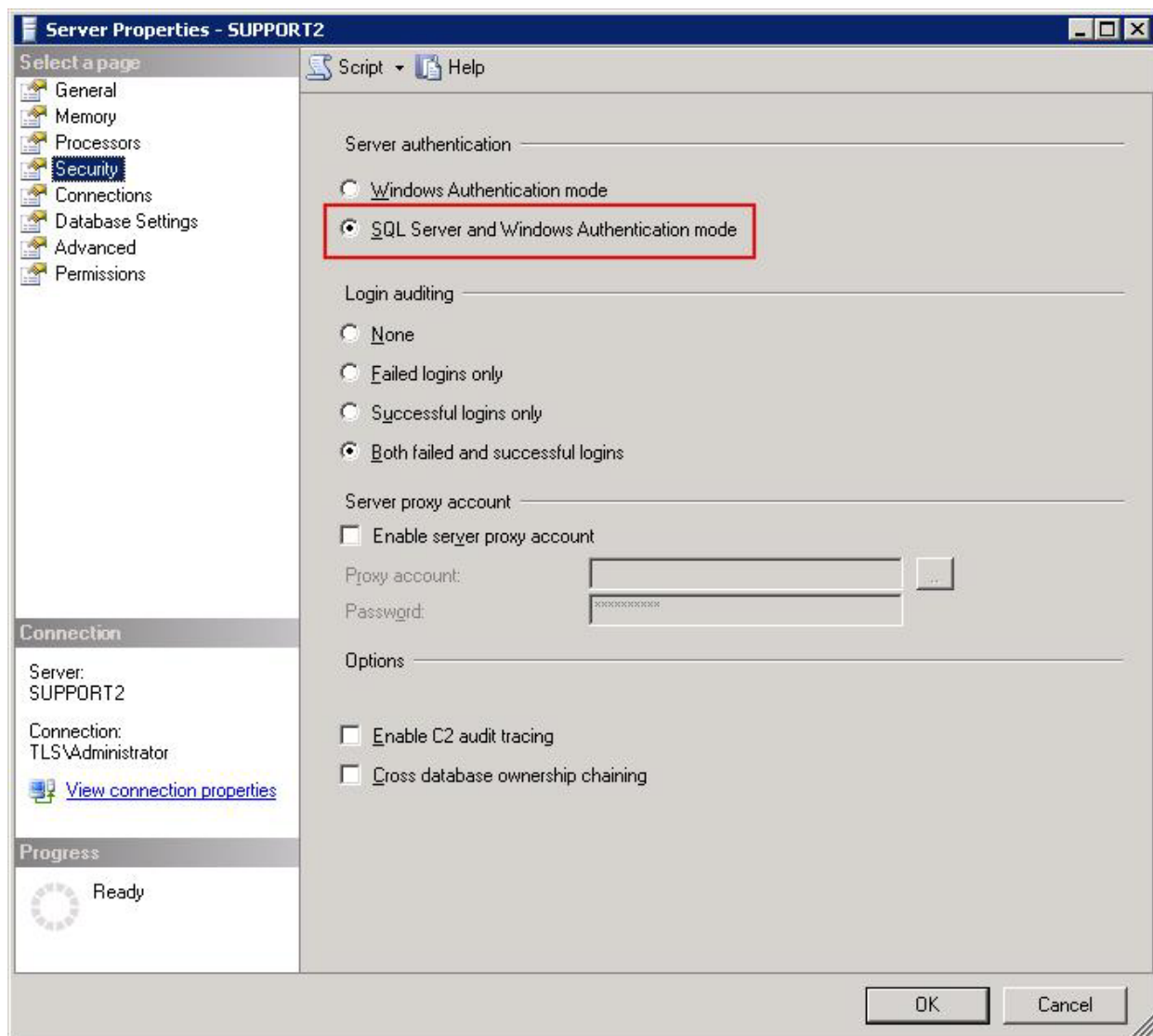
Das Einrichten der Quarantäne-Datenbank erfolgt durch folgende Schritte:

1. Einrichten des benötigten SQL-Users und konfigurieren der Quarantäne-Datenbank
2. Konfigurieren der Quarantäne in AntiVir Exchange

Einrichten des SQL-Servers

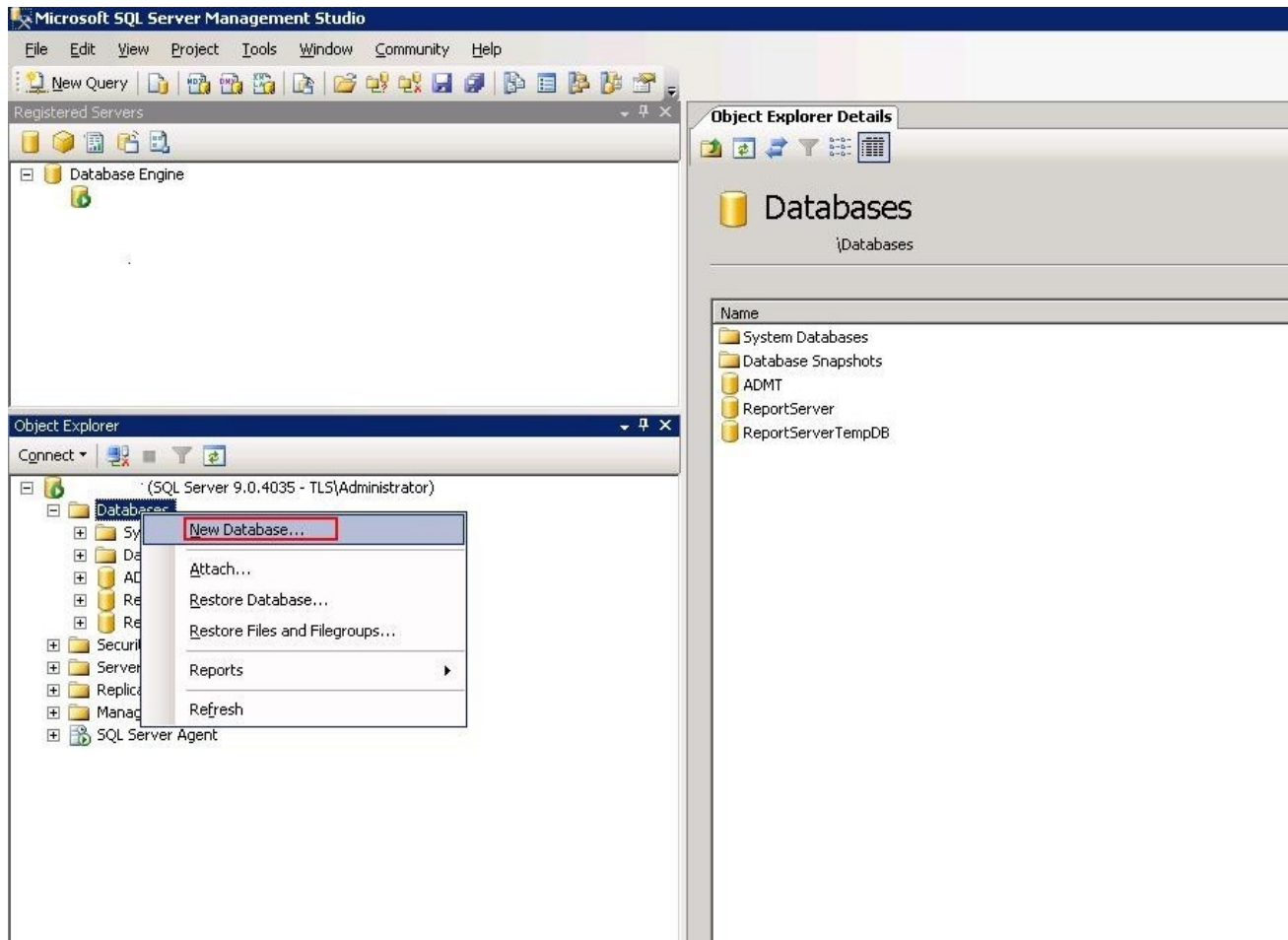
Im SQL Server brauchen wir einen User und eine Datenbank mit den Tabellen.

Der User kann *kein* Windows-User sein. Es muss ein expliziter SQL User sein (SQL Server nennt das **"Mixed Mode"**), da der Antivir Exchange-Dienst nicht unter einem User-Kontext, sondern als Local System läuft und Username und Passwort beim Aufruf übergeben werden.

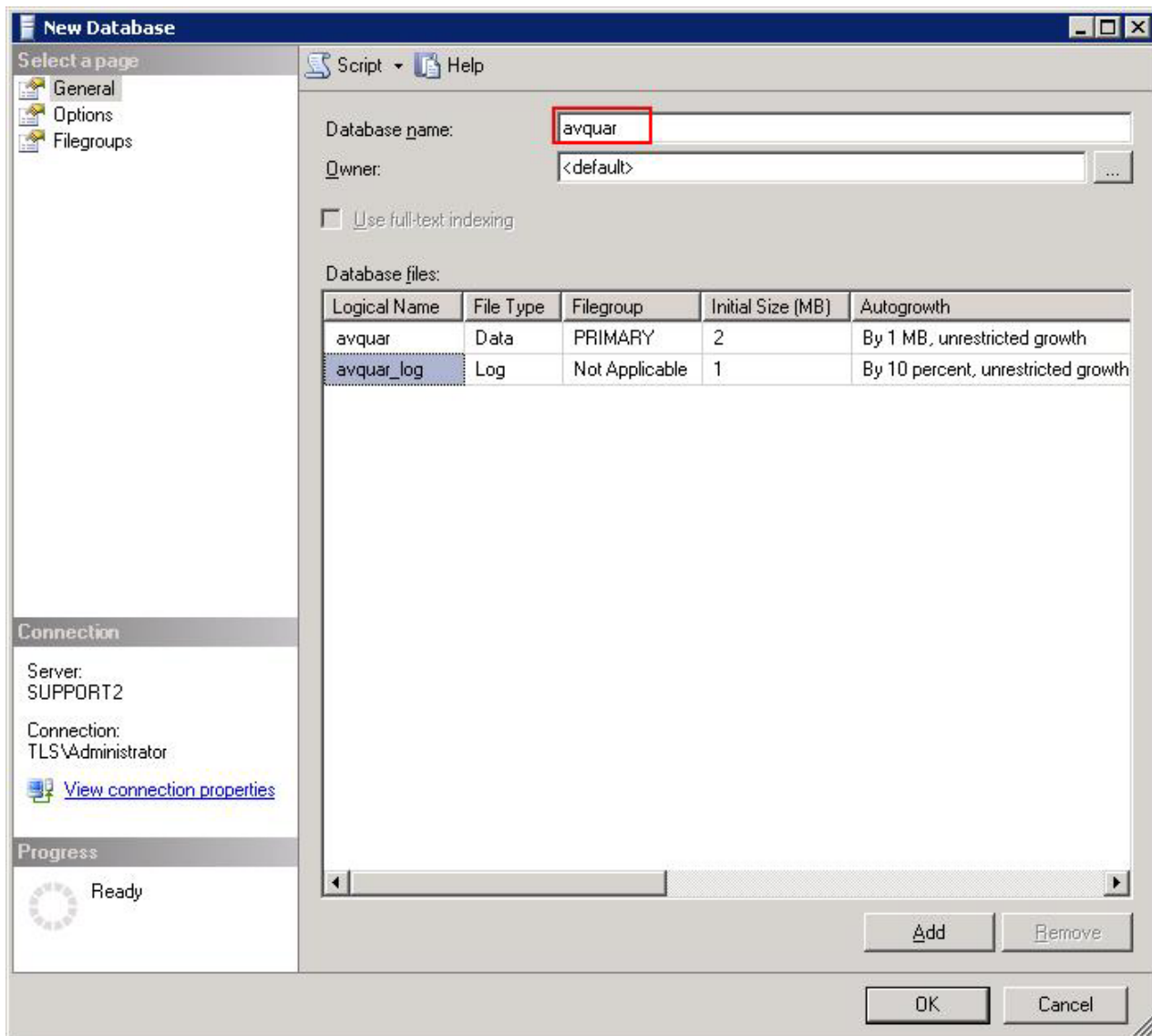


Anlegen der SQL-Datenbank

Die Datenbank legt der SQL Administrator an. Der Datenbankname sollte ein einfacher, kurzer String ohne Leerzeichen o.ä. sein, weil später die Quarantäne entsprechend angelegt werden kann und derselbe String dort als Folder-Name für die Ablage der Quarantäne - Emails verwendet wird (s.u.).



Anlegen des SQL – Benutzer



New Database

Select a page

- General
- Options
- Filegroups

Script Help

Database name: **avquar**

Owner: <default>

Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth
avquar	Data	PRIMARY	2	By 1 MB, unrestricted growth
avquar_log	Log	Not Applicable	1	By 10 percent, unrestricted growth

Connection

Server: SUPPORT2

Connection: TLS\Administrator

[View connection properties](#)

Progress

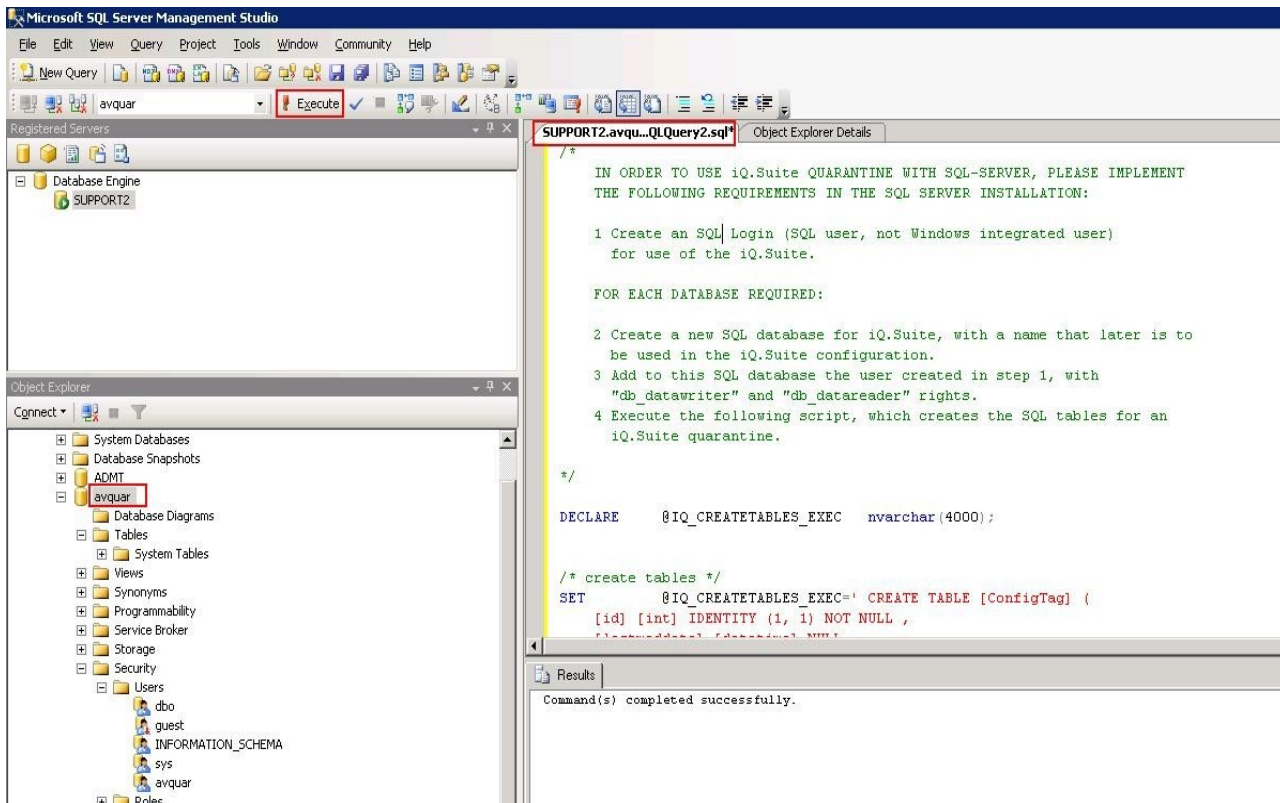
Ready

Add Remove

OK Cancel

Anlegen der Tabellen mit Script

Anschließend gibt es ein Skript **QUARANTINE.sql** im Support-Ordner (Avira/Antivir Exchange/Support), das in der Datenbank die benötigten Tabellen, Verknüpfungen und Stored Procedures anlegt und in dem die Details beschrieben sind. Ein SQL Administrator weiß, was damit zu tun ist. Das Skript kann man in die Management-Oberfläche des SQL Server kopieren und dort starten.



The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Registered Servers' pane shows a server named 'avquar'. The 'Object Explorer' pane shows the 'avquar' database selected. The main window displays a SQL script with the following content:

```

/*
IN ORDER TO USE iQ.Suite QUARANTINE WITH SQL-SERVER, PLEASE IMPLEMENT
THE FOLLOWING REQUIREMENTS IN THE SQL SERVER INSTALLATION:

1 Create an SQL Login (SQL user, not Windows integrated user)
for use of the iQ.Suite.

FOR EACH DATABASE REQUIRED:

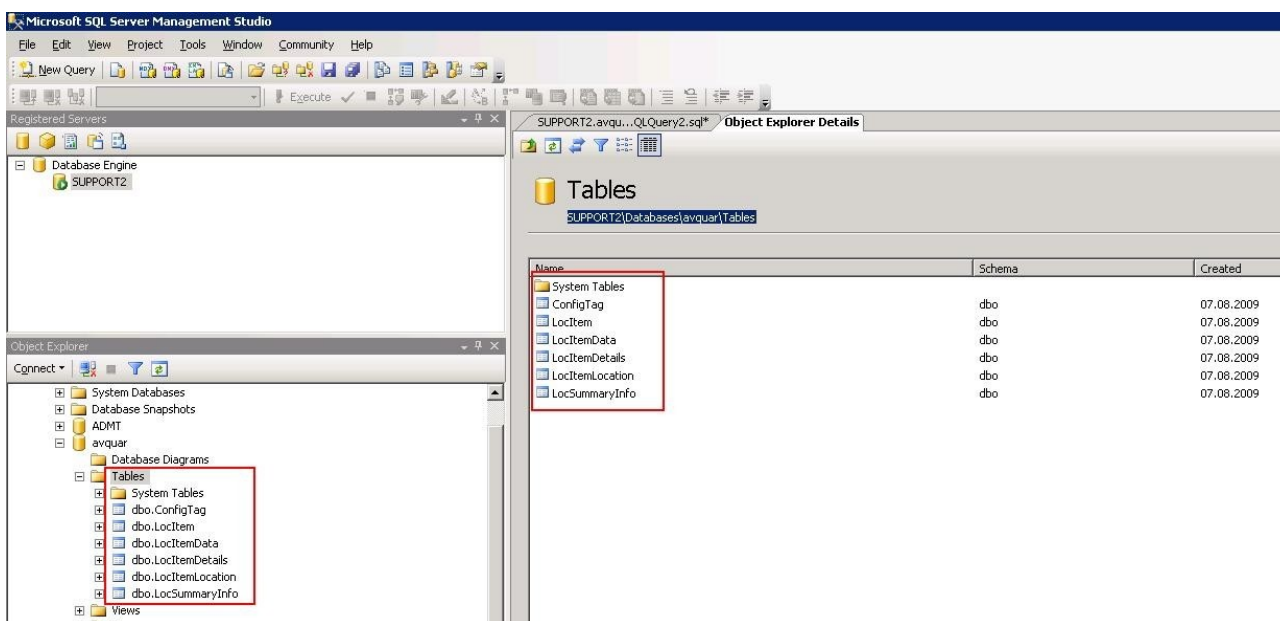
2 Create a new SQL database for iQ.Suite, with a name that later is to
be used in the iQ.Suite configuration.
3 Add to this SQL database the user created in step 1, with
"db_datawriter" and "db_datareader" rights.
4 Execute the following script, which creates the SQL tables for an
iQ.Suite quarantine.
*/

DECLARE @IQ_CREATETABLES_EXEC nvarchar(4000);

/* create tables */
SET @IQ_CREATETABLES_EXEC=' CREATE TABLE [ConfigTag] (
[id] [int] IDENTITY (1, 1) NOT NULL ,
[TableName] [varchar] (255) NOT NULL
);
';

```

The Results pane at the bottom shows the message: "Command(s) completed successfully."

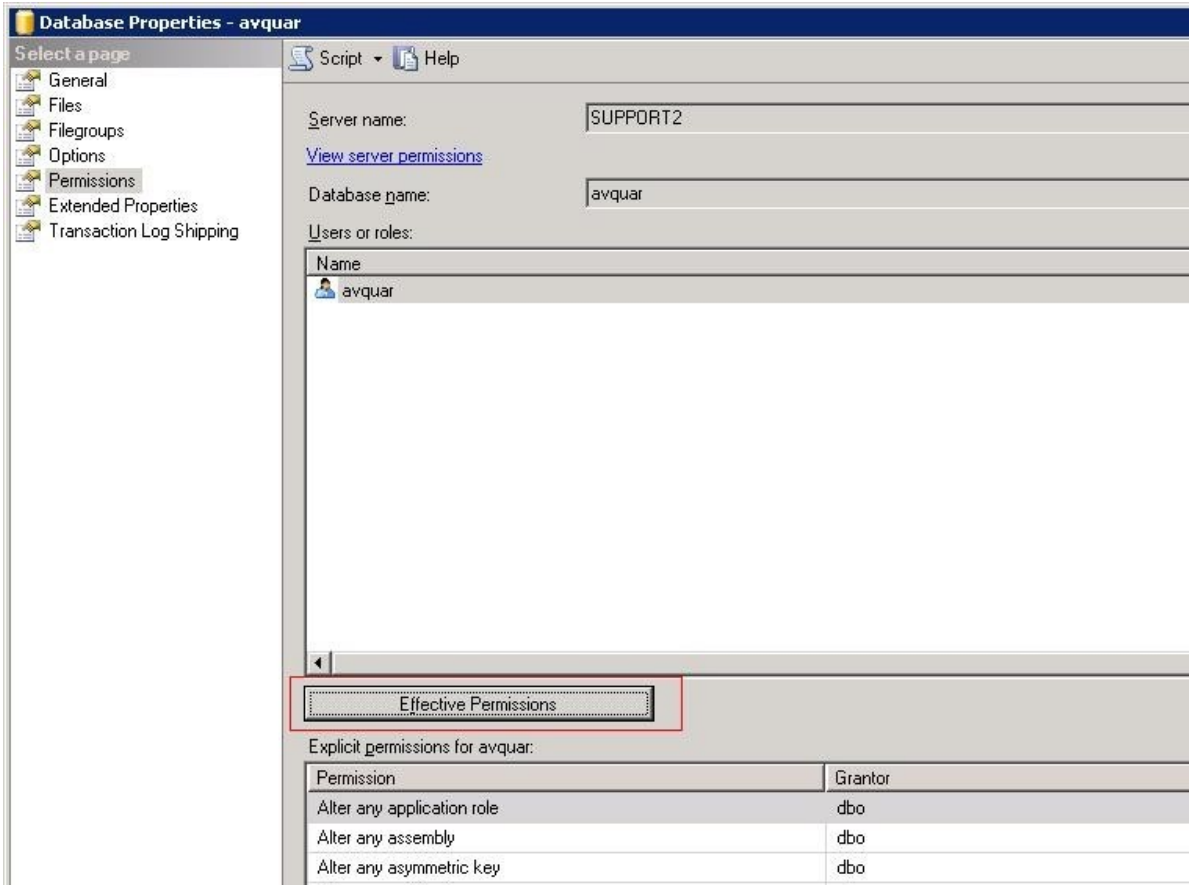


The screenshot shows the Microsoft SQL Server Management Studio interface. The 'Registered Servers' pane shows a server named 'SUPPORT2'. The 'Object Explorer' pane shows the 'avquar' database selected. The main window displays the 'Tables' view for the 'avquar' database. The table list is as follows:

Name	Schema	Created
System Tables		
ConfigTag	dbo	07.08.2009
LocItem	dbo	07.08.2009
LocItemData	dbo	07.08.2009
LocItemDetails	dbo	07.08.2009
LocItemLocation	dbo	07.08.2009
LocSummaryInfo	dbo	07.08.2009

Berechtigungen des angelegten SQL-Users prüfen

Der SQL User muss Einträge in den Datenbank-Tabellen einfügen, ändern und löschen können. AntiVir Exchange wird keine Schema-Änderungen an den Tabellen vornehmen, daher benötigt der User diese Rechte im Moment auch nicht. Wenn wir einmal Änderungen am Schema vornehmen müssen, muss das der SQL Admin dann von Hand während des Updates von AntiVir Exchange machen.

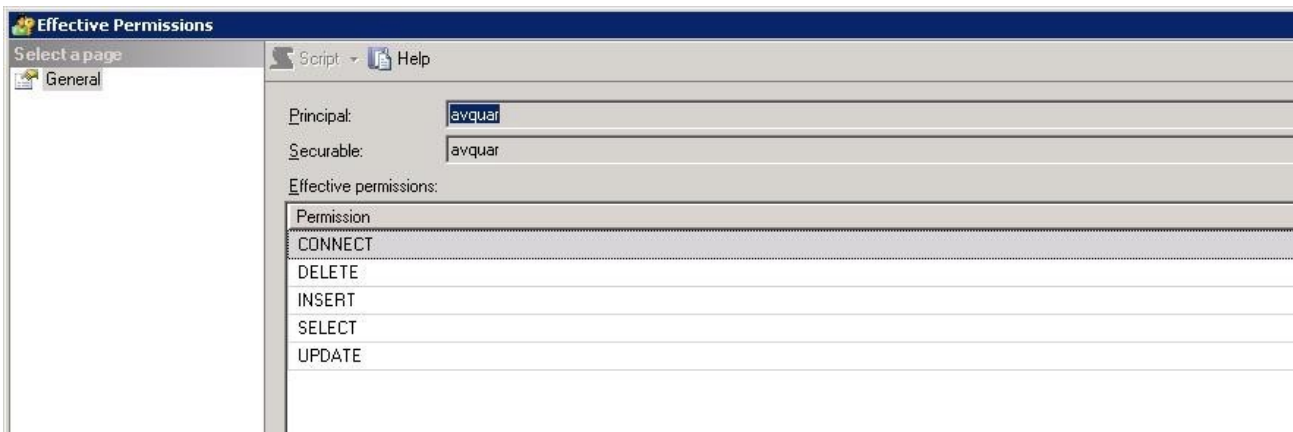


The screenshot shows the 'Database Properties - avquar' window. The 'Permissions' page is selected in the left-hand tree. The main area shows the following details:

- Server name: SUPPORT2
- Database name: avquar
- Users or roles: avquar

At the bottom, the 'Effective Permissions' section is highlighted with a red box. Below it, the 'Explicit permissions for avquar:' table is visible:

Permission	Grantor
Alter any application role	dbo
Alter any assembly	dbo
Alter any asymmetric key	dbo



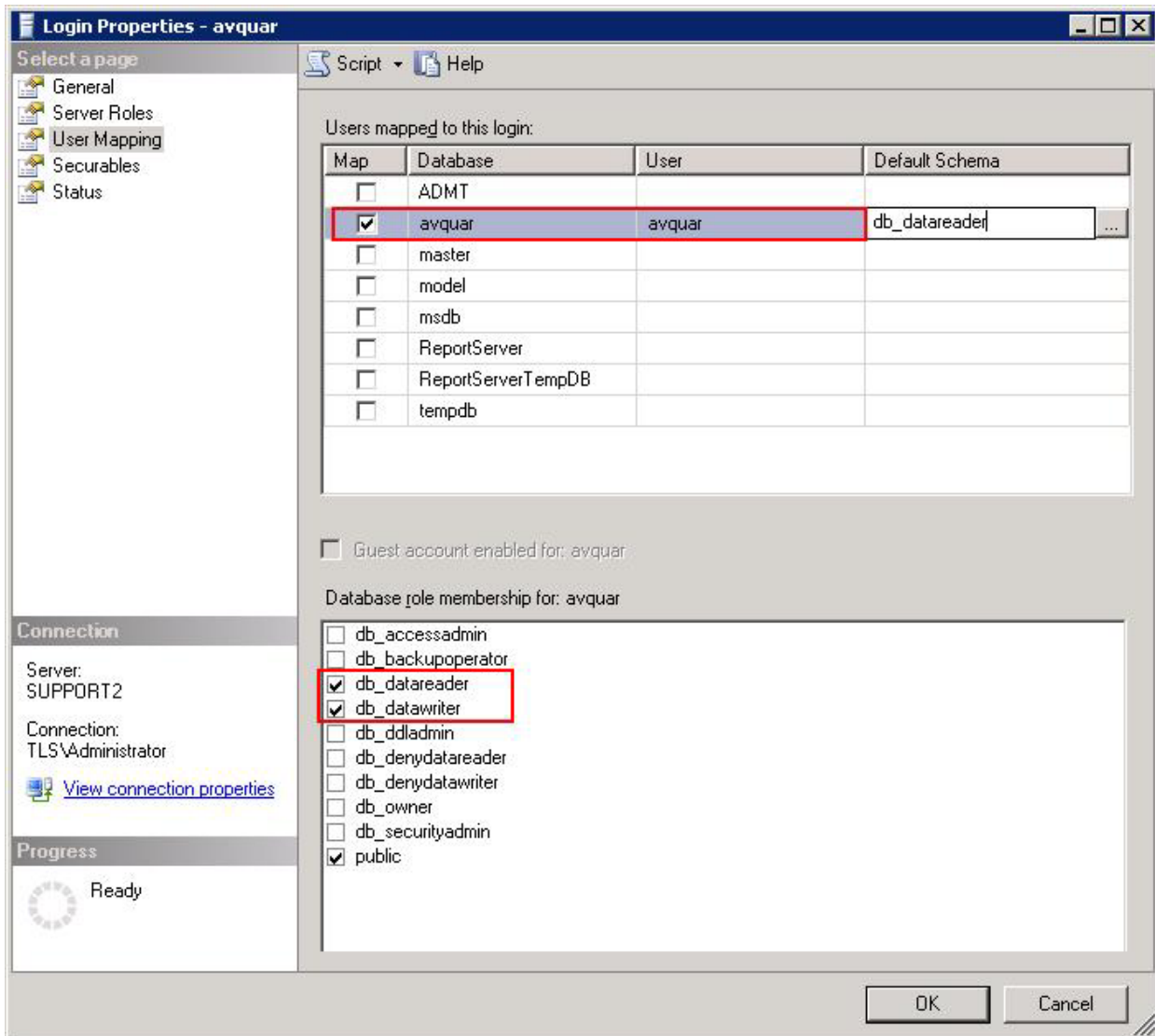
The screenshot shows the 'Effective Permissions' window. The 'General' page is selected. The details are as follows:

- Principal: avquar
- Securable: avquar

The 'Effective permissions:' table lists the following permissions:

Permission
CONNECT
DELETE
INSERT
SELECT
UPDATE

Anzeige der Berechtigungen



Login Properties - avquar

Select a page: General, Server Roles, **User Mapping**, Securables, Status

Script Help

Users mapped to this login:

Map	Database	User	Default Schema
<input type="checkbox"/>	ADMT		
<input checked="" type="checkbox"/>	avquar	avquar	db_datareader ...
<input type="checkbox"/>	master		
<input type="checkbox"/>	model		
<input type="checkbox"/>	msdb		
<input type="checkbox"/>	ReportServer		
<input type="checkbox"/>	ReportServerTempDB		
<input type="checkbox"/>	tempdb		

Guest account enabled for: avquar

Database role membership for: avquar

- db_accessadmin
- db_backupoperator
- db_datareader
- db_datawriter
- db_ddladmin
- db_denydatareader
- db_denydatawriter
- db_owner
- db_securityadmin
- public

Connection: Server: SUPPORT2, Connection: TLS\Administrator, [View connection properties](#)

Progress: Ready

OK Cancel

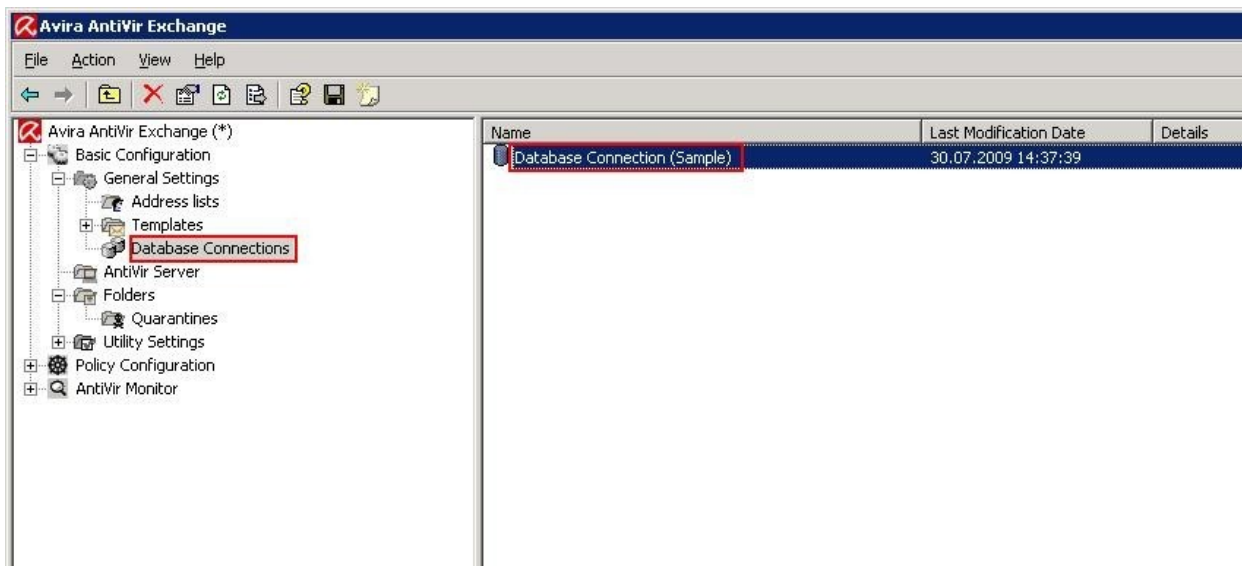
Konfigurieren der Quarantäne in AntiVir Exchange

In AntiVir Exchange gibt es zwei Stellen, an denen Einstellungen zu einer SQL Quarantäne Vorgenommen werden müssen:

- die Datenbank-Verbindung
- und die Quarantäne selbst.

Konfigurieren der Datenbank - Verbindung

Die Datenbank-Verbindung besteht aus dem ADO Connection String, dem in vorherigen Schritt angelegten SQL User mit seinem Passwort, und einer Timeout-Einstellung.



Zu beachten ist hier nur der **ADO Connection String**. Er definiert den Zugriff auf die Datenquelle mit ADO.

Der eingetragene **Default** ist:

```
Provider=SQLOLEDB;Initial Catalog=[DBCatalog];Data Source=[Server];User  
ID=[ADOUser];Password=[ADOPwd]
```

und funktioniert für **lokal** installierte SQL Server.

Ein Beispiel:

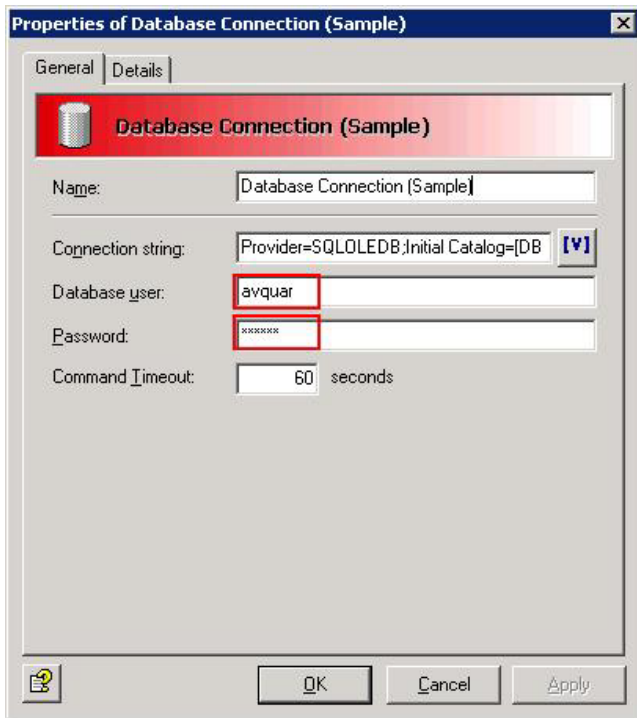
```
Provider=SQLOLEDB;DataSource=SUPPORT2\AVQUAR;Trusted_Connection=No;Initial  
Catalog=[DBCatalog]; UserID=[ADOUser];Password=[ADOPwd];Connect Timeout=120;
```

Da der Server SUPPORT2 sehr langsam ist und der Default-Timeout zum Herstellen einer Verbindung nicht immer ausreicht, haben wir den Wert dort etwas erhöht. Am Anfang kann jedoch mit den „Default“ Werten gearbeitet werden. Sollten sich während des Betriebs Störungen bei der Erreichbarkeit der Datenbank ergeben, so kann der Wert langsam erhöht werden.

Die Variablen **[ADOUser]** und **[ADOPwd]** verweisen auf die entsprechenden Einstellungen auf der gleichen Seite. Dies verhindert, dass das Passwort in der Konfiguration im Klartext gespeichert wird. Prinzipiell kann aber der User und das Passwort auch direkt in den ADO-String geschrieben werden. Die Variable **[DBCatalog]** gibt die zu verwendende Datenbank an.

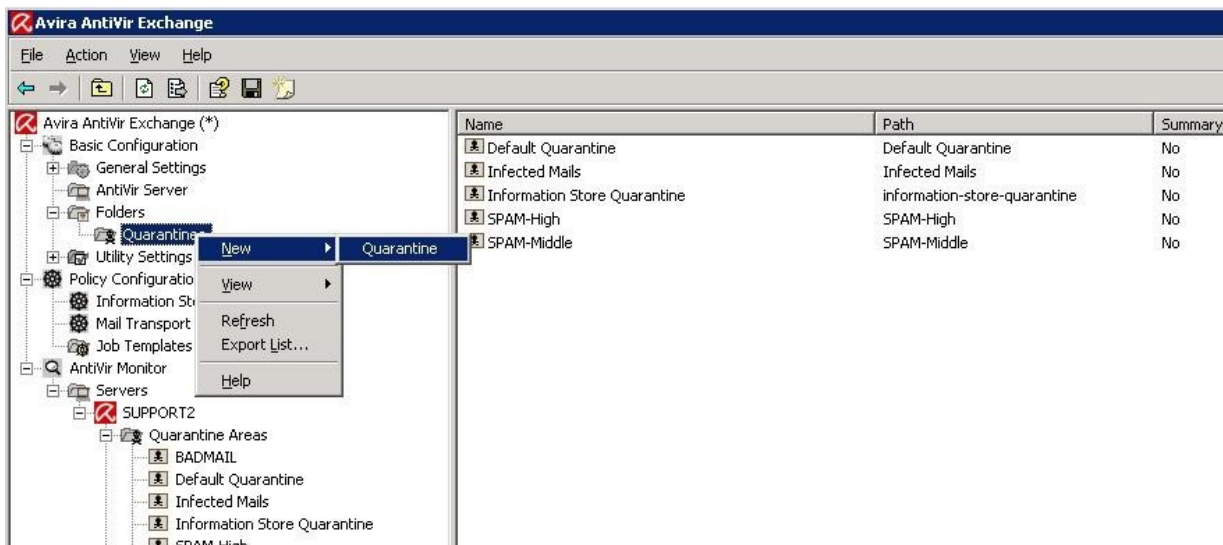
AntiVir Exchange schreibt dort den jeweiligen Wert hinein, wenn die Quarantäne angelegt ist, siehe weiter unten.

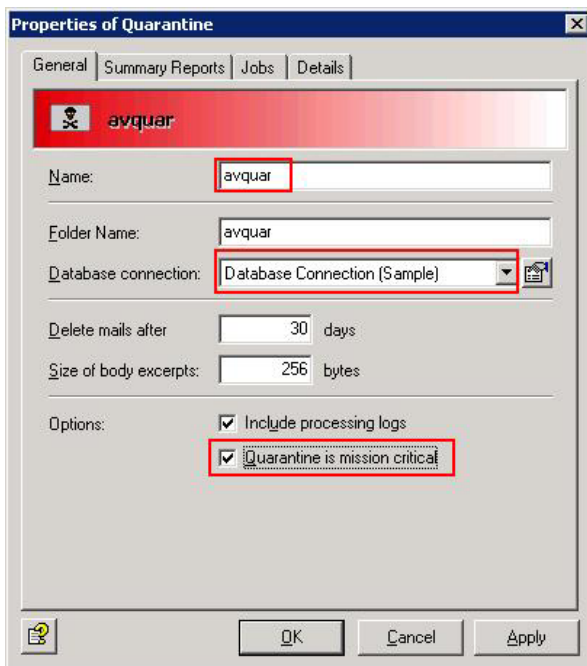
Die Variable erlaubt es, dieselbe Datenbankverbindung für mehrere Quarantänen zu benutzen. Die Variable **[Server]** schließlich wird durch den lokalen Servernamen ersetzt. Wie man im zweiten Beispiel oben sieht, kann man dort aber auch die SQL-Instanz spezieller angeben.



Die Quarantänen

Um eine SQL-Quarantäne anzulegen, muss man immer eine *neue* Quarantäne erstellen. Man kann nicht bestehende (Jet-DB-)Quarantänen nachträglich in eine SQL-Quarantäne umwandeln (aber man kann eine neue SQL-Quarantäne anlegen und dann die abgelegten Emails aus der alten Quarantäne per Drag&Drop in die neu angelegte SQL-Quarantäne kopieren).





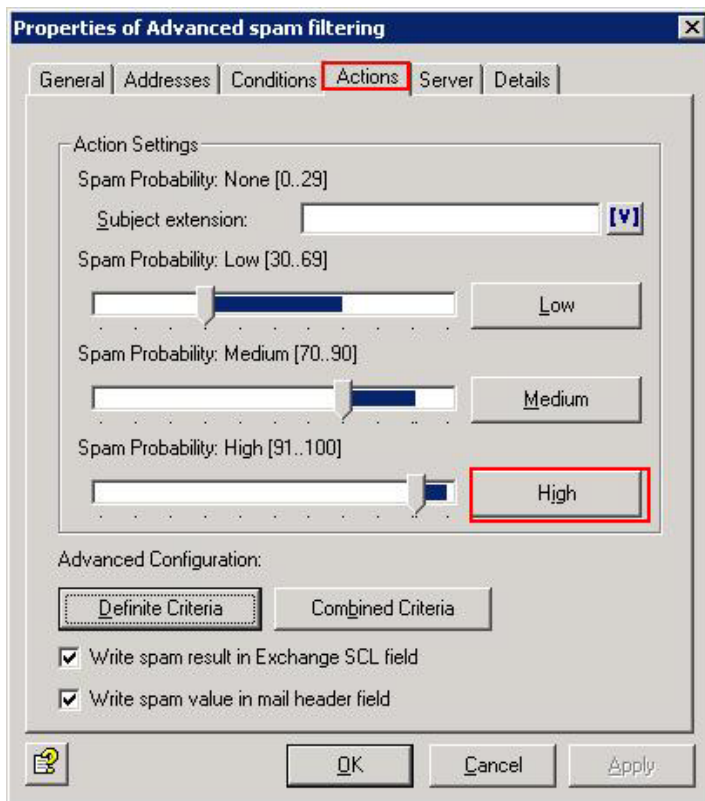
Es gibt dabei nur einen "Trick" - der gewählte Ordner-Name der neuen Quarantäne wird bei SQL-Quarantänen gleichzeitig für die Variable **[DBCatalog]** verwendet. Man trägt also als Ordner-Name den Datenbank-Namen ein (hier "SQL_SPAM_HIGH") und wählt darunter die oben eingerichtete Datenbank-Verbindung aus. Der "**Name**" darüber ist nur ein Display-String und kann beliebig gewählt werden.

Nachdem man hier **OK** gedrückt hat, lässt sich der Folder Name nicht mehr ändern, also muss man beim Eintragen etwas **aufpassen**.

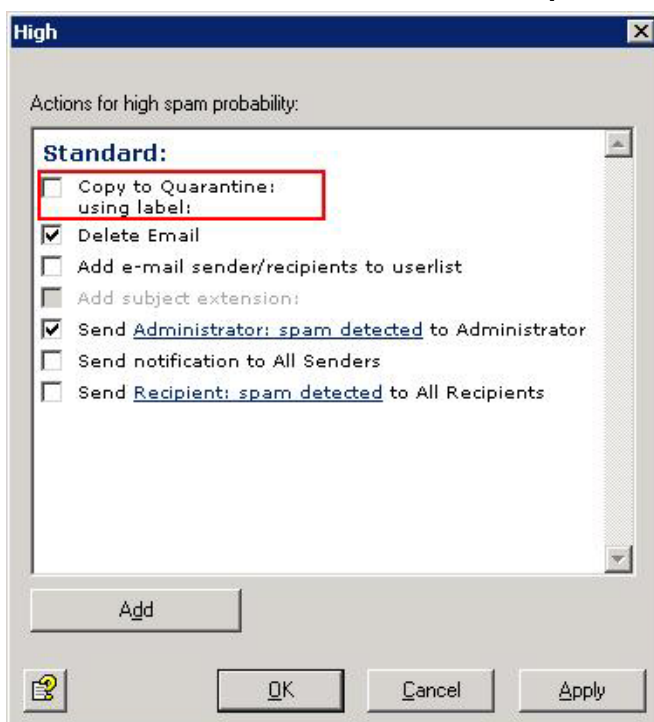
Natürlich kann man den Datenbanknamen auch direkt in den ADO Connection String eintragen, ohne die **[DBCatalog]**-Variable zu verwenden. Dann muss man allerdings für jede Quarantäne eine eigene Datenbankverbindung konfigurieren.

Jetzt kann man in einem Job die neue Quarantäne verwenden, und dann im Monitor sehen ob sie funktioniert.

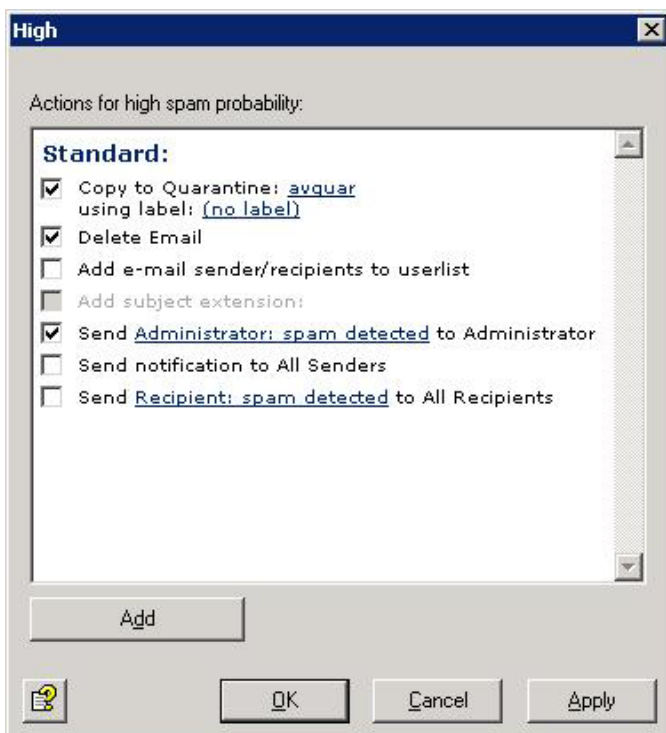
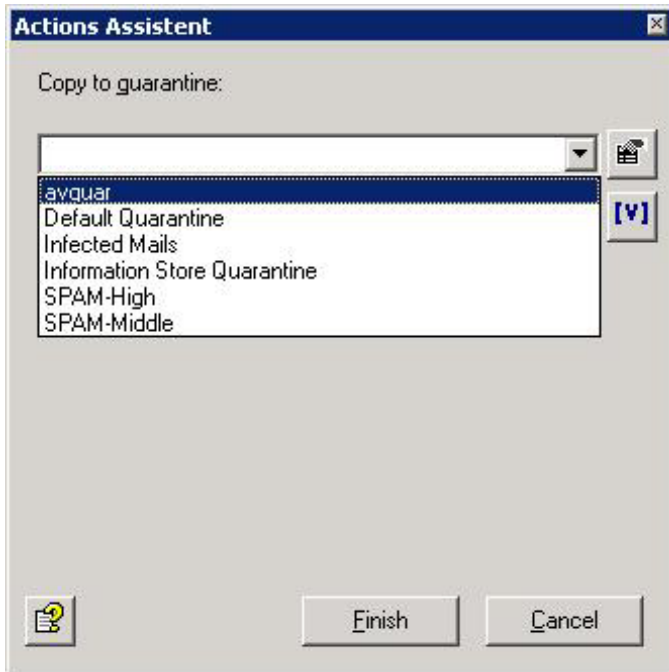
Auswahl der Quarantäne im Job Advanced spam filtering



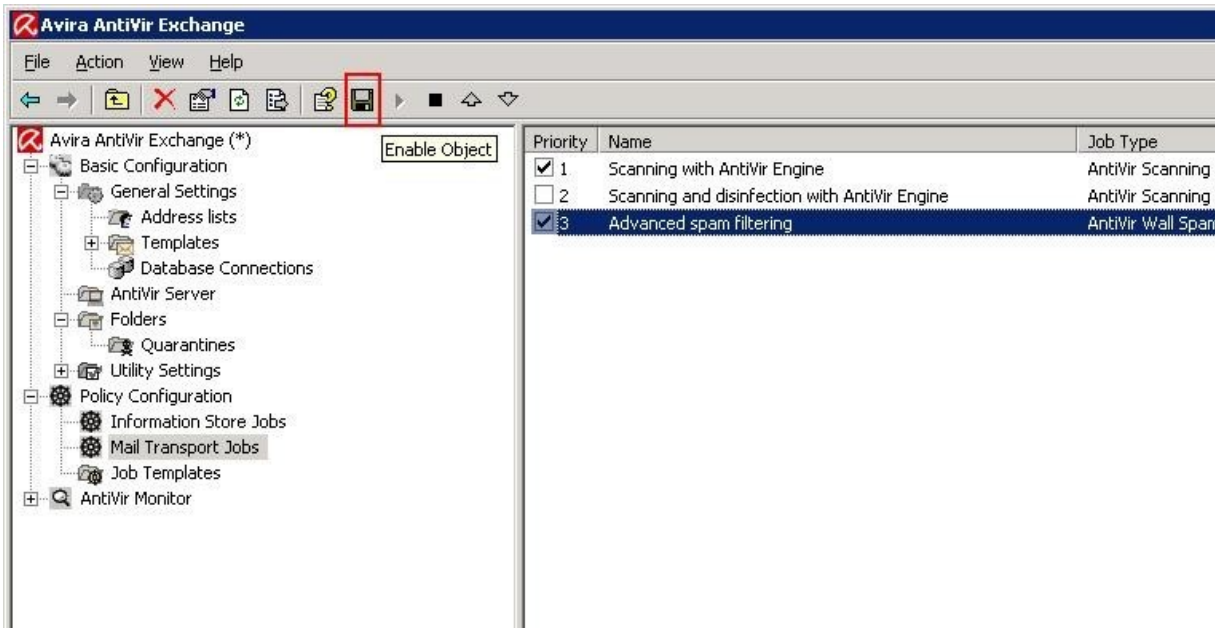
Da wir eigentlich ja nur die „SPAM-HIGH“ Quarantäne über den jetzt eingerichteten SQL-Index verwenden möchten, muss dies natürlich noch in dem dazugehörigen Job konfiguriert werden. Dazu verwenden wir den Job „**Advanced spam filtering**“ und die „Action“ -> High.



Wir können hier nun die „Action“ wählen, wohin unsere „SPAM-HIGH“ Quarantäne verschoben wird. Hier wählen wir nun einfach die schon angelegte Quarantäne „avquar“ aus, welche von unserem SQL-Server bedient wird.



Zum Schluss sollten wir natürlich nicht vergessen, die vorgenommenen Änderungen zu speichern.



Fertig.

Hier noch ein kleiner Tipp für die Darstellung sehr große Quarantänen:

Auf langsamen Systemen kann es sehr lange dauern, bis nach einem Doppelklick auf eine sehr große Quarantäne die Einträge angezeigt werden.

Dann kann es helfen, *vor* dem Doppelklick mit der rechten Maustaste auf die gewünschte Quarantäne den Filter auf "Heute" zu aktivieren: