MainBoss Advanced 3.0:
Installation and Administration

Table of Contents

Introduction

MainBoss Advanced 3.0 Installation and Administration Guide

This guide explains the process of installing MainBoss Advanced 3.0. The first chapter gives a quick guide that should be sufficient for many users. Later chapters give more details for those with advanced needs.

This guide is aimed at providing a reasonable path to getting MainBoss up and running. The procedures and settings we describe are not the only possibilities; some sites may have good reason to take different approaches (e.g. in setting up permissions over your local area network). However, if you don't follow our suggestions and you contact MainBoss Technical Support for help, we will probably recommend that you change your configuration to match what is described in this guide.

Software and Hardware Requirements

Operating System: MainBoss Advanced will run on any of the following operating systems:

- Windows XP Pro or better (with Service Pack 2 or later):
    - XP Pro should provide satisfactory performance with one or two simultaneous users, provided that both users aren't on external machines
- Windows Server 2003 or later; or
- Windows Vista Business or better

We do not recommend Windows XP Home, Windows Vista Home, or Windows Vista Basic.

.NET Framework: Before beginning the installation process, every computer where MainBoss will run must have the .NET framework installed (version 3.0 or later). Windows Vista comes with a suitable version of .NET already installed. For Windows XP and Windows Server 2003, .NET is available through the standard Windows Update. Choose "Custom" rather than "Express" and look under "Software, Optional". (When you do this, Windows Update may give you a number of updates before you can get to the "Software, Optional" ones.)

Hardware: We recommend that any computer running MainBoss Advanced should have:

- Windows XP and Windows Server 2003: at least a 3 GHz P4 processor with 1 GB of RAM
- Windows Vista: at least dual 2.5 GHz processors with 2 GB of RAM

In both cases, get the fastest RAM available.

Since MainBoss requires SQL Server 2005, you should also check the hardware requirements of the SQL Server package you intend to use. SQL Server will run on a single computer; if you also intend to use this computer to run MainBoss, the computer's hardware and operating system should satisfy the requirements of both SQL Server and MainBoss. Every other computer where you intend to run MainBoss has to satisfy MainBoss's hardware and software requirements, but doesn't have to satisfy SQL Server's.

If you intend to run MainBoss on multiple computers, we strongly recommend that MainBoss users should belong to a domain. A domain lets MainBoss users sign on with the same login name and password on different computers. You can create a domain with Microsoft Server software, or with a Linux network running Samba.

If you're not currently using a domain setting and are thinking about getting one, we recommend Windows Small Business Server. If you'll have fewer than five people using MainBoss simultaneously, you can use the basic version of Small Business Server. If you'll have more people using MainBoss simultaneously, you should get the premium version of Small Business Server, which includes SQL Server 2005 as part of the package. (In order to get SQL Server 2005, you must get Windows Small Business Server 2003, Release 2 or later—earlier releases have earlier versions of SQL Server, which can't be used for MainBoss.)

Microsoft SQL Server 2005

MainBoss Advanced uses Microsoft's SQL Server 2005 to read and write all data. SQL Server 2005 must be installed at your site before you can begin using MainBoss Advanced. Specifically, you must have:

- SQL Server 2005 SP2 (Version 9.00.3042.00 or greater)

MainBoss doesn't work with earlier versions of SQL Server. However, if you need to keep running earlier versions in order to support other software packages, you can run both SQL Server 2005 and earlier versions of SQL Server on the same computer.

If you've installed MainBoss, you can determine what version of SQL Server you have by going to **Help** → **About** in the MainBoss menus.

While this *Installation and Administration Guide* offers suggestions of how to work with SQL Server 2005, SQL Server is a Microsoft product and Thinkage has no control over its behavior. If you have any difficulties with SQL Server, contact Microsoft, not MainBoss support.

SQL Server 2005 is actually a family of compatible software products, with different members of the family designed for different work loads. If you have a small organization, you can use SQL Express, a free version of SQL Server 2005 available from Microsoft. For more information, see *SQL Express Considerations* on page 26.

SQL Express will likely be adequate if you intend to run the SQL Server on Windows XP Windows XP Pro (which supports fewer than five simultaneous users). The same applies to Windows Vista. If you will have more than five simultaneous users, you should run SQL Server on Windows Server 2003 and buy a version of SQL Server whose size is appropriate to your needs.

If you intend to have multiple people using MainBoss, or you intend to access MainBoss from multiple machines, you should install SQL Server Management Studio and the configuration tools at the same time that you install SQL Server 2005. This is discussed in the Appendix.

Installation Locations for MainBoss Software

When setting up MainBoss, it's important to distinguish between where you store the data and where you store the software:

- MainBoss *data* will be stored on a computer where SQL Server 2005 is running. There is only one copy of the data; this is shared by all users at your site.
- MainBoss *software* should be installed on every computer where people wish to work with MainBoss. In other words, you may have many copies of the software. (The software for MainBoss Advanced doesn't have to be installed on the server system if people won't be using MainBoss from that system.)

Problems may arise if you store the MainBoss executable program on one computer and try to run it on another computer. (This is a known problem with the Microsoft .NET framework.) As a result, you must install the MainBoss software on every computer where you intend MainBoss to run.

The same consideration applies to MainBoss's installation procedure. The installation file(s) must be copied to each computer where you intend to install MainBoss. Once you've installed MainBoss, you can delete these copies of the files. (One easy way to get copies of the files is to copy them to a CD, then take the CD to every computer where you want to install the software.)

Installation Overview

In most organizations, a single person will do most of the work for installing and setting up MainBoss. This guide calls that person the *MainBoss Administrator*. We recommend that the MainBoss Administrator performs installation operations in the following order:

1. Install SQL Server 2005 first. This should be on a computer that is accessible to all other computers where you wish to use MainBoss Advanced. We will call this the *Server computer*.

2. Install the MainBoss Advanced software on a single computer first. Ideally, this should be the computer where SQL Server is running; this eliminates any issues that may arise when accessing SQL Server over your local area network.
3. Start MainBoss on the computer where you just installed it, and create a maintenance organization as described in *Creating a Maintenance Organization* on page 4.
4. If you intend to import data from MainBoss Basic (MainBoss 2.9), follow the instructions given in the guide *Migration from MainBoss Basic to MainBoss Advanced*.
5. Enter the license keys you were given when you licensed MainBoss Advanced. For more information, see *Entering License Keys* on page 6.
6. If multiple people will be using MainBoss Advanced, you must add those people to MainBoss's table of authorized users, as described in *Users* on page 8.
7. Once you have MainBoss Advanced working correctly on one computer, proceed to install it on all other computers where you'll wish to use MainBoss.
8. Once the software is installed on a given computer, each person who will be using MainBoss on that computer must start MainBoss and add the MainBoss database to their personal list of maintenance organizations. For more information, see *Installing MainBoss on Other Computers* on page 11.
9. If you have licensed the @Requests module, install @Requests as described in *Installing the @Requests Service* on page 13.

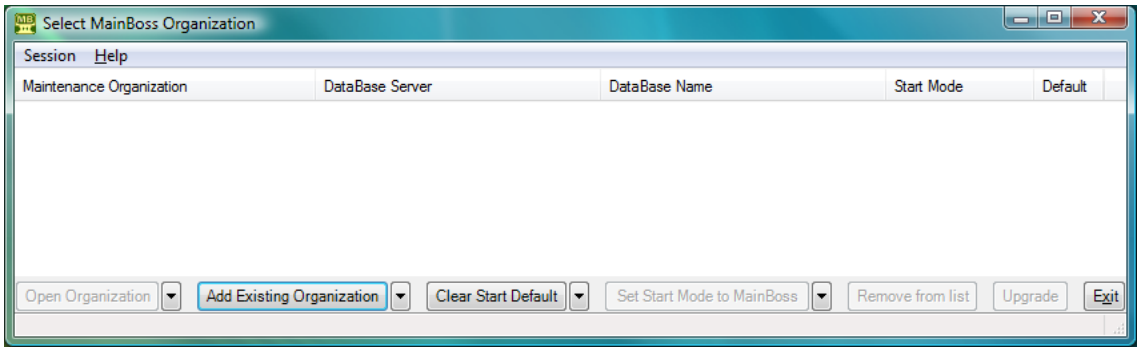
The Windows operating system is full of pitfalls, especially if your IT department has set up a customized environment. For help on dealing with difficulties during the installation process, see *Appendix A: Setting Up SQL Server* on page 24 and *Appendix B: Troubleshooting* on page 29, especially if you'll be using SQL Express.

Installing MainBoss Software

MainBoss Advanced is currently available through download from the MainBoss web site. To install the software, follow the instructions given on the download web page.

Creating a Maintenance Organization

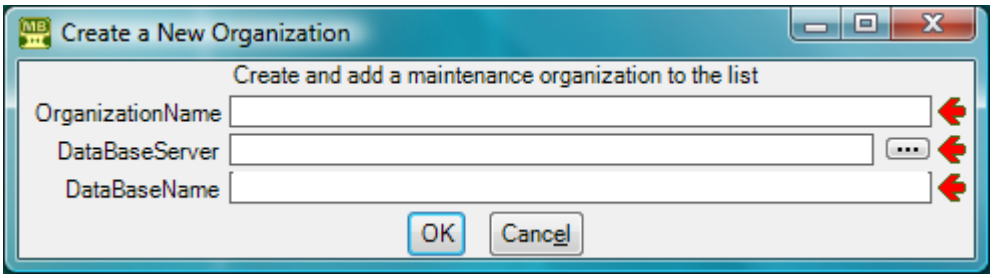
When you first start MainBoss, the software displays a window that lets you create a new maintenance organization (database).



In order to create a MainBoss database, you must be logged into an account that has SQL Server Administrator privileges on the SQL Server that will manage the MainBoss database.

To create a new maintenance organization database:

1. Drop the arrow on **Add Existing Organization** and click **Create New Organization**. This opens a window like this:



2. In "**OrganizationName**", enter a name that MainBoss can display when you're using the database. Each user can have his or her own name for the same database. Your "**OrganizationName**" can contain spaces and special characters if you wish.

3. In "**DataBaseServer**", specify the name of the SQL Server 2005 server that you will use. You can get a list of all available servers by clicking the ". . ." button.

If you click the "..." button after "**DataBaseServer**", MainBoss will attempt to determine what servers are available to you. However, this list may not be complete; certain firewall settings and other technical issues can prevent servers from appearing in the list, even though the servers may be available for use.

4. In "**DataBaseName**", enter the name of the new database. This will be the name used by SQL Server 2005; it must be different from all other databases controlled by the server. We recommend that this name should *not* contain any spaces or special characters—just letters and digits. It can be useful for "**DataBaseName**" to be similar to "**OrganizationName**" so that it's easy to see which organization goes with which database; however, it's not necessary.

5. Click **OK**. MainBoss will proceed to create the database.

Entering License Keys

Once you have installed MainBoss and created a maintenance organization, MainBoss enters a special mode for you to enter license keys for the new organization. You must have a separate set of license keys for each organization you use; for more information, contact your dealer or Thinkage Ltd.

License keys have the format

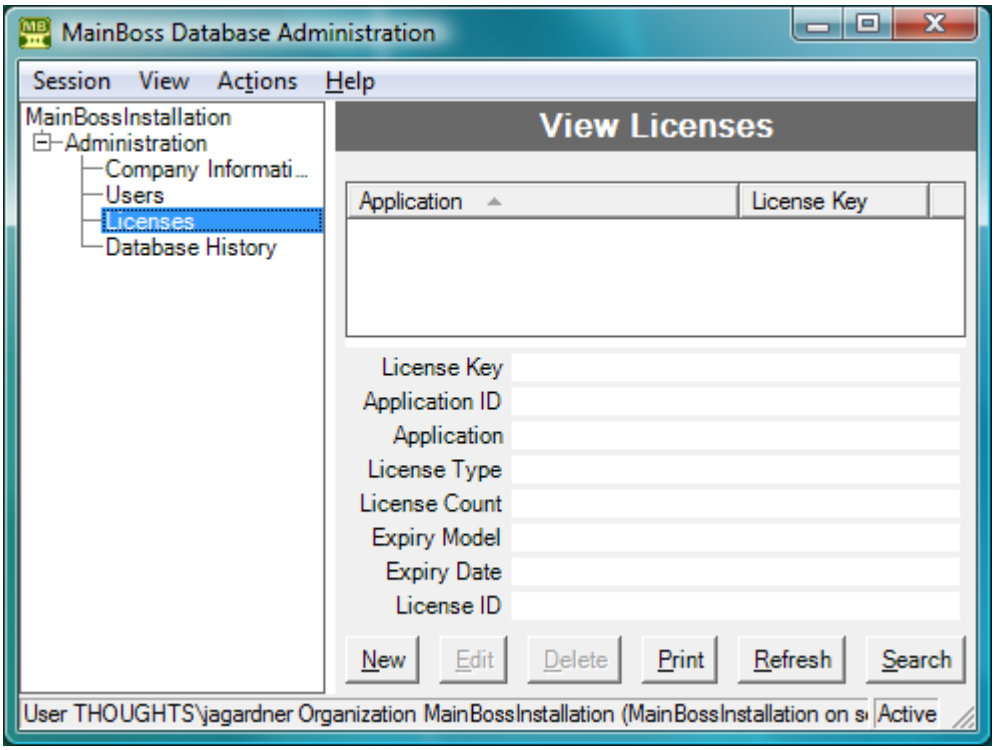
```
AAAAA-AAAAA-AAAAA-AAAAA-AAAAA
```

where each `A` is a letter or number.

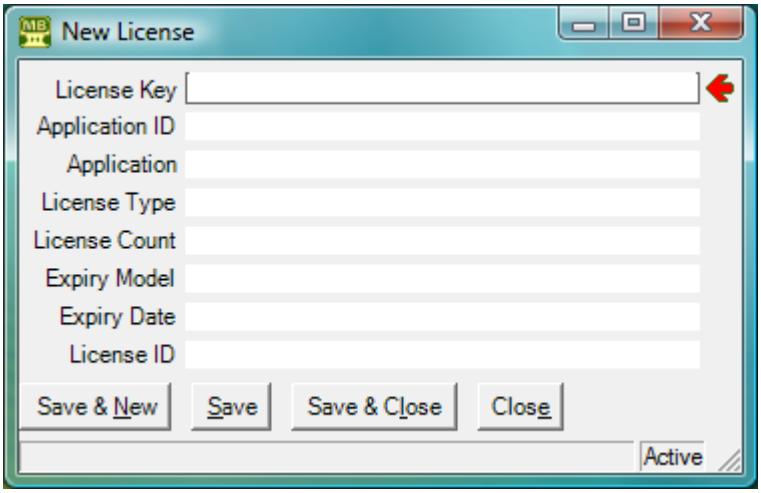
The presence or absence of license keys will change the appearance of many MainBoss windows. For example, if you have not licensed the **Inventory** module, you will not see functions related to inventory in the control panel and in other parts of MainBoss.

To enter your license keys:

1. In the MainBoss control panel, go to **Administration** | **Licenses**:



2. Click **New**. MainBoss opens a window where you can enter a license key:



3. In "**License Key**", enter any of the license keys that you were given. Letters can be typed in either upper or lower case—it doesn't matter which. You must type the hyphens "-" between groups of characters.
4. When you have typed the key correctly, MainBoss will fill in the rest of the window with information extracted from the key. If this doesn't happen, check the key to make sure you have typed it correctly.
5. Click **Save & New** to save what you've just typed in. MainBoss will clear all the fields and let you type in a new key.
6. Repeat Steps 3-5 until you have typed in all the keys. Click **Close** or **Save & Close** to return to the previous window. The window will show all your new keys.

Replacing Keys: In some cases, your database may already have license keys. For example, if you upgrade your MainBoss license (e.g. to authorize more users), you will be given one or more license keys to replace your previous keys. In this case, follow these steps:

1. Go to **Administration** | **Licenses** and click on one of the keys you want to replace.
2. Click **Edit**. MainBoss will open a window where the "**License Key**" field shows the old field.
3. Type the new key into the "**License Key**" field.
4. Click **Save & Close**.
5. Repeat this process for each new key.

Users

In order for users to access a MainBoss maintenance organization, their login names must be recorded in the MainBoss database. When a maintenance organization is first created, the only user who can access the organization is the person who originally created the database.

Important: In order for someone to be able to use MainBoss from several different computers, that person must have the same login name and password on each of the computers. (This is automatically true if you use domains.)

Scopes: In the procedure described below, you record the login names of people authorized to use MainBoss. During this process, you'll be given the opportunity to specify a *scope* for that person's login name.

- If your site uses *trusted domains*, the scope should be the trusted domain to which the user belongs.
- If your site doesn't use trusted domains (or if you aren't sure whether you use them or not), we recommend leaving the "**Scope**" field blank. A blank scope field means that the given login name is a valid account on the computer where SQL Server runs. SQL Server will then accept access from any other computer in your local network, provided that the login name and password on the other computer match the name and password on the Server computer.

  For example, suppose you authorize `jsmith` to use MainBoss and leave "**Scope**" blank. Then the Server computer must accept `jsmith` as a login name. SQL Server will accept access from any other `jsmith` in your local network, provided the password for `jsmith` on non-Server computers is the same as the password for `jsmith` on the Server computer.

Note that the names in MainBoss's **Users** list must be individual login names; they can't be groups. In MainBoss, login names are associated with records in order to create an audit trail. This is why MainBoss itself doesn't accept group authorizations—an audit should record exactly who made a particular change, which means tracking the individual login name, not just a group name.

SQL Server Authorization: In order for someone to use MainBoss, the person's login name must be registered in both the MainBoss database and with SQL Server. Only someone with SQL Server Administration privileges can register new users with SQL Server.

- The easiest scenario for adding new users to MainBoss is if someone with SQL Server Administration privileges does the work. In this case, MainBoss can automatically register each new user in the MainBoss database and with SQL Server.
- If the person who adds new users to MainBoss *doesn't* have SQL Server Administration privileges, then the process needs two separate steps:

- Registering the new users with MainBoss, as described in *Registering New Users with MainBoss* on page 9 (which can be done by anyone who's already registered with MainBoss and SQL Server)
- Registering the new users with SQL Server, as described in *Authorizing Users in SQL Server* on page 38 (which can only be done by someone with SQL Server Administration privileges)

In order to deal with these two scenarios, MainBoss has an option set in **Administration** | **Users**. Go to the **Defaults for User** section.

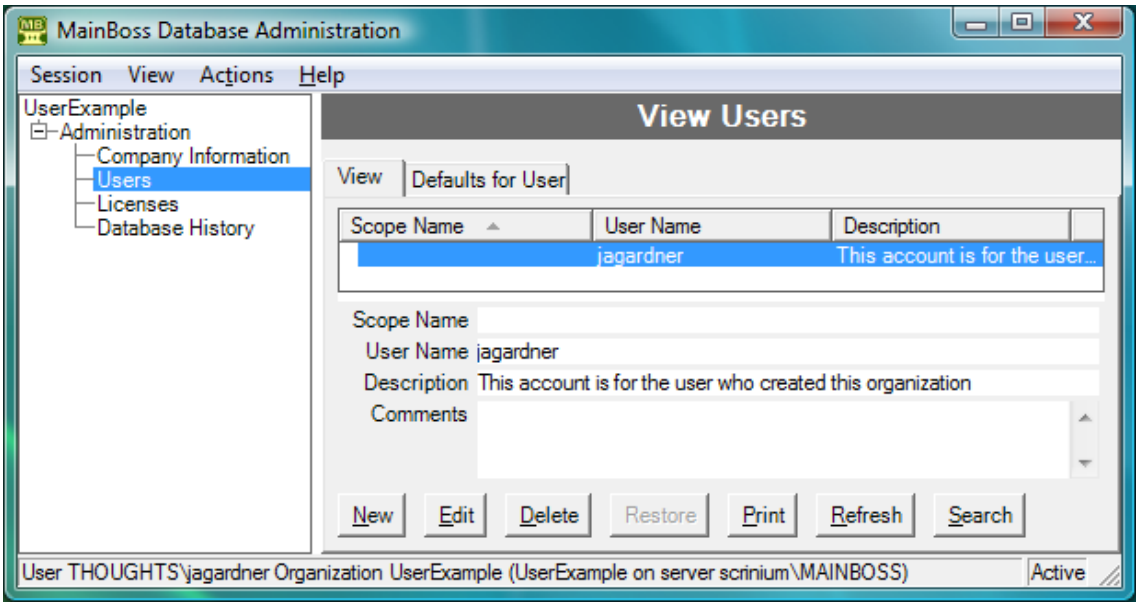
- If the option **MainBoss manages SQL Security** is checkmarked, anyone adding new users must have SQL Server Administration privileges
- If the option is blank, you must use the two-stage approach to adding new users (add the users in MainBoss first, then add them in SQL Server)

(As an alternative two-stage approach, you could create a Windows group named `mainboss` in the domain. A SQL Server Administrator could then set up SQL Server permissions so that anyone in the group could access the MainBoss database. Then, whenever someone new needs to be authorized to use MainBoss, a Windows administrator simply adds the new person to the group. Some IT departments may prefer the "group" approach over authorizing each individual within SQL Server.)

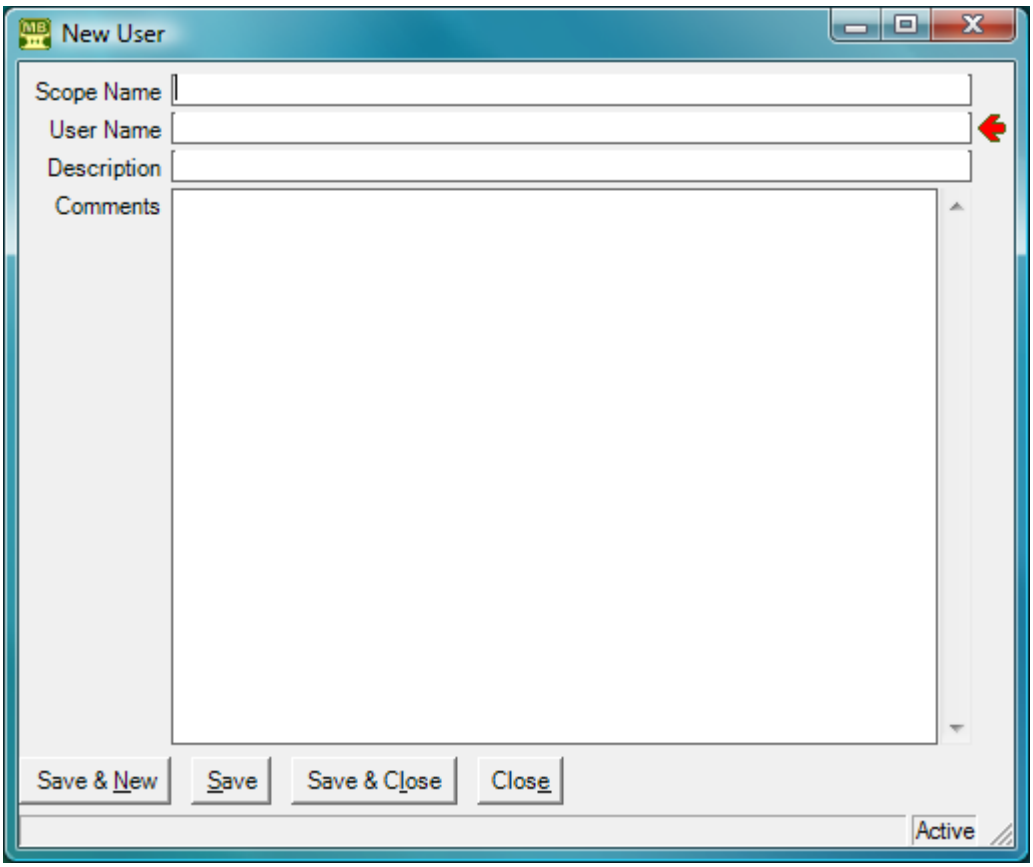
Registering New Users with MainBoss

To authorize someone to access a particular MainBoss database, follow these steps:

1. In the MainBoss control panel, go to **Administration** | **Users**:



2. Click **New**. This opens a window where you can specify other users who can use the database.



3. If your site uses trusted domains, enter the user's domain name in "**Scope**". Otherwise, leave "**Scope**" blank.
4. In "**User Name**", type the person's login name.
5. In "**Description**", type the person's real name (e.g. `Pat Smith`).
6. Click **Save & New**.
7. Repeat Steps 3-6 for each person who'll be using MainBoss. Click **Close** or **Save & Close** to return to the previous window. The window will show the users you have authorized.

Your MainBoss license restricts how many people may use MainBoss *simultaneously*, not how many users you can have in total. Therefore, there's no limit on the number of users you can add to your **Users** table.

SQL Server Security: As noted in a previous section, if the option **MainBoss manages SQL Security** (in **Defaults for User**) is checkmarked, the person who adds new users must have SQL Server Administration privileges on the SQL Server that manages MainBoss. If the person adding new users *doesn't* have this privilege, you must turn off **MainBoss manages SQL Security** (or else you'll get an error). If **MainBoss manages SQL Security** is turned off, new users must be separately authorized within SQL Server. For more information, see *Authorizing Users in SQL Server* on page 38.

Local Users vs. Domain Users: If your site uses domains, it's possible for the same login name to exist both inside and outside of a domain. For example, suppose you have a domain named `OurDomain` that contains a user named `jsmith`. It's possible for the SQL Server computer to have a local login name `jsmith` as well as the domain version (typically written `OurDomain\jsmith`).

If both versions of `jsmith` need to use MainBoss, they must both be in the MainBoss **Users** table. For the domain `jsmith`, set "**Scope**" to the domain name (`OurDomain`). For the non-domain `jsmith`, set "**Scope**" to the name of the computer. (This can be determined from the computer's **Control Panel** by going to **System**.) The non-domain `jsmith` will only be able to use SQL Server if the Server machine also has a local non-domain `jsmith` with the same password, and if the Server computer has been appropriately set up for network file-sharing. Also the non-domain `jsmith` must have a SQL Login recorded with SQL Server, which MainBoss may or may not create automatically (depending on various options). If you need to create a SQL Login by hand, see *Authorizing Users in SQL Server* on page 38.

Installing MainBoss on Other Computers

Once you have set up MainBoss on one computer, and are sure that it is working correctly with SQL Server 2005, you can set up MainBoss on other computers (if you wish).

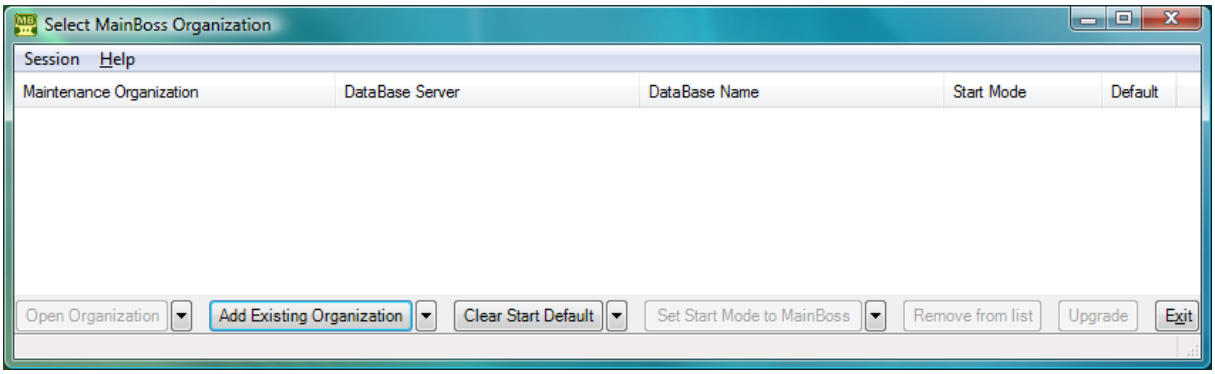
A MainBoss license dictates the number of people who can be using MainBoss simultaneously. For example, suppose you are licensed for 10 simultaneous users. You can install MainBoss on any number of computers, but only 10 people will be allowed to use the program at any one time. (The database keeps track of how many people are currently using MainBoss, and will prevent other people from starting the program if you've reached your licensed maximum.)

To install MainBoss on any computer, follow the steps given in *Installing MainBoss Software* on page 4. Once you've done so, **each person who will use MainBoss on that computer** must set up his or her options for connecting with the MainBoss database.

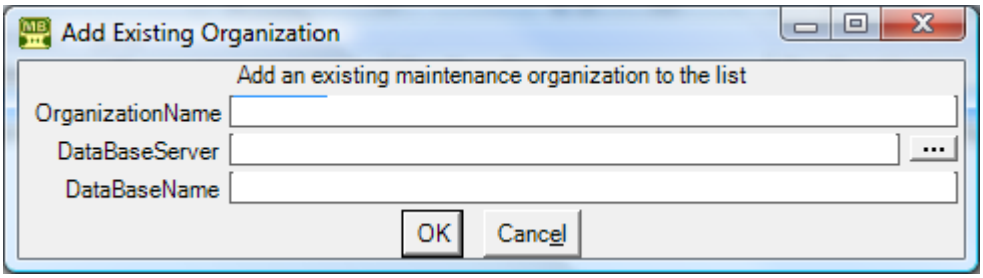
Important: MainBoss options are set up for each individual user. This means that each user must login and set up a connection to the MainBoss database. The process is easy, but the MainBoss Administrator must tell every other user the "**DataBaseServer**" and "**DataBaseName**" needed for the connection. Alternatively, you can have each user log in and then the administrator "leans over the user's shoulder" to set up the database connection.

To set up MainBoss so that it can access an existing maintenance organization:

1. A MainBoss user should login to the computer under his/her usual login name.
2. Start MainBoss on the new machine. It will display a window asking you to specify the maintenance organization:

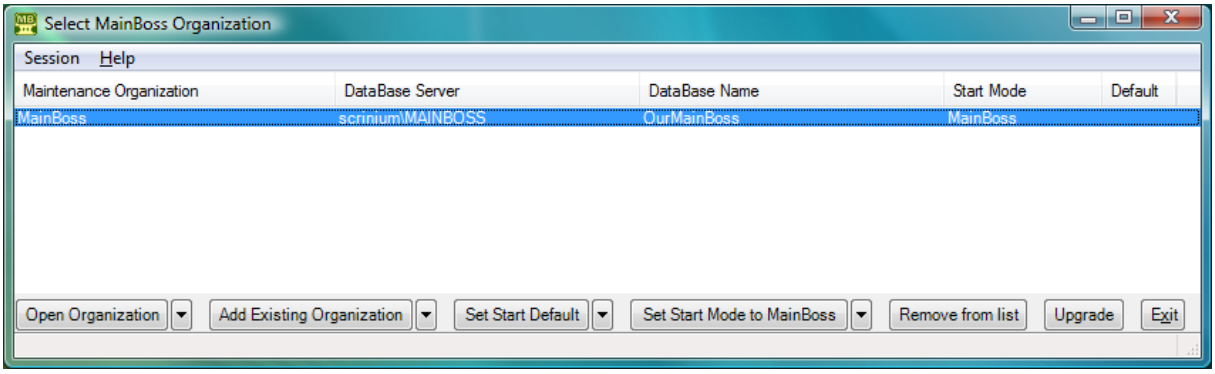


3. Click **Add Existing Organization**. This opens a window like this:



4. In "**OrganizationName**", enter a name that will let you recognize this organization. This does *not* have to be the name used when the database was created; however, using the same name may help avoid confusion.
5. In "**DataBaseServer**" and "**DataBaseName**", enter the same values that you did when you originally created the database (see *Creating a Maintenance Organization* on page 4).

6. Click **OK**. MainBoss will add the database name to the list, as in



7. Click on the line containing the maintenance organization name, then click **Set Start Default**. (This means that MainBoss will use this maintenance organization whenever it starts up.)
8. Click **Open Organization**.

Once you have set up this computer to access the maintenance organization, you don't have to do it again. Every time you start MainBoss on this computer, it will automatically start working with the specified database.

Installing the @Requests Service

This section only applies if you have licensed the @Requests module.

In order to use @Requests, you must activate a Windows service. This service watches for incoming e-mail sent to the address associated with @Requests; the service also sends out acknowledgement messages when appropriate.

The @Requests service can only be activated on a single computer; MainBoss won't let you install it on more than one computer. Ideally, it should be installed on a computer that is easily accessible to the MainBoss Administrator and where the MainBoss Administrator has Windows Administrator privileges. The @Requests service requires Windows Administrator privileges to install and configure. You will also find it easier to access the @Requests event log (i.e. diagnostic messages) if you are logged in to the computer where @Requests is actually running.

Accessing the @Requests event log from a different computer is possible, but there are some preconditions. For more, see *The @Requests Event Log* on page 17.

Wherever you install the @Requests service, MainBoss should be correctly installed on the same computer before you try activating the @Requests service.

The @Requests service can only be installed by someone with administrator privileges. This means that the person installing @Requests must be in the computer's `Administrators` group.

To install the @Requests service for a particular maintenance organization, you must go to **Administration** | **@Requests** and click **Manage**. (Since this starts a privileged piece of software, Windows Vista will ask for permission to proceed.) MainBoss opens a window that lets you proceed with the installation process. Click **Install**.

What Name Do You Specify for Running the Service: When you click **Install**, MainBoss will ask you to specify a username and password. The @Requests service will run as if it was a program invoked by this user. The following considerations apply:

- Ideally, you should create a new login name for running the @Requests service, and you should *only* use this name for running @Requests. This makes it easier to track what @Requests is doing and to separate @Requests activities from everything else on the computer.
- Ideally, the login name associated with @Requests should *not* have any privileges (e.g. it should not be an administrator account). The @Requests service doesn't need privileges, and it's best not to give privileges to any program that doesn't need them. Since you must install @Requests from an account with administrator privileges, this means that your login name will be *different* from the one you're logged into when you install @Requests
- The login name should never have its password expire.
- The login name should be entered in the MainBoss users table (**Administration** | **Users**) *before* you try to install @Requests, and you should make sure that someone logged in under that name can actually use MainBoss. (For example, make sure you can create a work order or a request.)
- When you enter the name during the installation of @Requests:
    - If the login name is part of a domain, **you must specify the domain name as part of the username**, e.g. `OURDOMAIN\jsmith`.
    - If the login name is not part of a domain, you must specify the machine name, as in `LOCALHOST\jsmith`.

Configuring @Requests

Once you have installed @Requests, you configure it by going to **Administration** | **@Requests** and clicking **Manage**. (On Vista, you'll be warned that this starts a privileged program, and will be asked for permission to continue.) MainBoss then opens a window showing the current status of @Requests.

In order to configure @Requests, the service must be stopped. This will be true the first time you configure @Requests. If you want to change the configuration later on, you must click **Stop Service** (on the **Start Service** drop-down button) before you can proceed to reconfigure.

The process for configuring @Requests will be described shortly. During this process, you must specify a mailbox where people can send problem reports. You must also specify how @Requests can send and receive messages from that mailbox.

- Messages are sent using a technique called SMTP.
- Messages are received (read) using one of two techniques: POP3 or IMAP. We recommend that you use POP3 unless your site has a reason for preferring IMAP.

  There are more recent versions of these techniques called POP3S and IMAPS. These have extra security features built in, and are generally preferable to the earlier POP3 and IMAP. However, if your e-mail software is relatively old, it may not support POP3S and IMAPS.

For more information on SMTP, POP3 and IMAP, see the documentation for your e-mail software.

To configure @Requests, click **Edit**. MainBoss opens a window that contains the following:

Incoming mail server: The name of the computer that receives your incoming mail. For example, if you're running Microsoft Exchange, this is usually the name of your Exchange server. This machine must allow mail access through POP3 or IMAP4.

Incoming mail server's port: The port used by the mail server. The normal port numbers are:

- For POP3: 110
- For POP3S: 995
- For IMAP: 143
- For IMAPS: 993

Note: Getting the port numbers wrong is the most common source of errors when configuring @Requests.

Username: The name of the e-mail account to which requests will be sent. For example, if users send mail to `workreqs@yourcompany.com`, the **Username** would be `workreqs`.

Password: The password (if any) for the specified **Username**.

Use IMAP4 instead of POP3 protocol: By default, @Requests uses POP3 to obtain incoming requests from the designated e-mail address. If you click this button, @Requests will use IMAP4 instead.

Mailbox (IMAP4 only): The name of the mailbox, if you're using IMAP4.

Attempt Transport Layer Security (TLS) connection: If you checkmark this box, @Requests tries to connect to the mailbox uses a TLS (secure) connection. In other words, MainBoss will attempt to use POP3S or IMAPS instead of POP3/IMAP. **We strongly recommend that you try turning on this option** (although not all e-mail packages support it).

Outgoing SMTP mail server (SMTP): The name of the computer that will handle @Requests' outgoing mail. Normally, this is the same as "**Incoming mail server**", but it doesn't have to be.

Outgoing SMTP port: The port used by the SMTP mail server. This is usually port 25.

Wakeup interval: Dictates how often @Requests will check for new incoming messages and outgoing acknowledgements. By default, this is `0:30` (every 30 minutes).

Automatically create new requestors: If this is checkmarked, @Requests will automatically create new entries in the **Requestors** table (**Coding Definitions** | **Requests** | **Requestors**) whenever an e-mail message is received from someone who isn't currently on the list.

Remember that your **MainBoss Requests** license may place limits on the total number of requestors allowed in the **Requestors** table.

Return E-mail address: The return e-mail address to be placed on the acknowledgements (i.e. the address that will be used if the recipient wants to reply). This should usually be the e-mail address of a person in the maintenance department: someone who can personally handle responses from clients.

Title: The title to be used when creating acknowledgements.

Subject line prefix: The start of the acknowledgement's subject line. The default is `@Re:` so that if a requestor sends an e-mail with the subject

```
Tap is dripping
```

@Requests sends back an acknowledgement with the subject

```
@Re: Tap is dripping
```

Introduction: Text that will be used as the first line of all acknowledgements.

For information on testing whether POP3 and/or IMAP4 are enabled on your system, see *Testing POP3/IMAP4 Permissions for @Requests* on page 43.

Once you have configured the @Requests service, you should start it as described in the next section.

Starting and Stopping the @Requests Service

The window for managing @Requests (obtained by going to **Administration** | **@Requests** and then clicking **Manage**) also contains buttons that will start or stop the @Requests service. To start the service, click **Start Service**.

- If you are logged in to the computer where @Requests is installed, **Start Service** (and **Stop Service**) will work if you have Windows Administrator privileges (which must already be the case, if you can successfully click **Manage**).
- If you are logged in to a different computer, **Start Service** and **Stop Service** only work if the following are true:
    - You have Windows Administrator privileges on the computer where the @Requests service is installed.
    - The computer where @Requests is installed has file-sharing turned on. (For how to turn on file-sharing, see *File-Sharing* on page 30.)

Uninstalling @Requests

You may wish to uninstall the @Requests service (e.g. to change the computer on which it's running). In order to do the uninstall, you must be logged in on the computer where the service is running, on an account with administration privileges.

- Go to **Administration** | **@Requests** and click **Manage**.
- In the resulting window, click **Uninstall**.

If you use some other method to uninstall the software (e.g. using the Windows `sc delete` command), MainBoss's records will say the service is still installed and you won't be able to install the service anywhere else.

The @Requests Event Log

The @Requests event log lists any diagnostic messages that the @Requests service has received. If @Requests is behaving the way you expect, checking the event log may help you determine what's going on.

The easiest way to see the event log is to log in to the computer where the @Requests service is running. Start MainBoss; **Administration** | **@Requests** will list the log messages.

Depending on your Windows configuration, you may need Windows Administrator privileges to see the event log. If you need privileges to see the log but you don't have them, you'll see an error message saying that you don't have permissions to see the log.

Messages are only put into the event log when the @Requests service has some kind of problem. Therefore if you are having difficulties with @Requests, check the event log to see if the messages help determine what's going on. (As always, it may be useful to use Google to search for explanations of the messages.)

Don't confuse the event log with **Administration** | **@Requests** | **E-mail Requests**. The **E-mail Requests** table shows incoming e-mail messages that were turned into requests. These can be read from any computer.

You can get rid of event messages by clicking **Clear Events** in the @Requests management window (obtained by clicking **Manage** in **Administration** | **@Requests**). To clear event messages, you need the same privileges as the ones described in *Starting and Stopping the @Requests Service* on page 17.

Seeing the Event Log from a Different Computer: As noted above, the easiest way to see the event log is to be logged into the computer where the @Requests service is running. However, you can see the event log from a different computer provided the following are true:

- You must have Windows Administrator privileges on the computer where @Requests is running. (You do *not* need Windows Administrator privileges on the computer where you're logged in.)
- The computer where @Requests is installed must have file sharing turned on. For instructions on how to turn on file-sharing for a computer, see *File-Sharing* on page 30.
- The **Remote Registry** service must be running on the computer where @Requests is installed.

To turn on the **Remote Registry** service if it isn't already running, follow these steps:

1. Click on the Windows **Start** button.
2. Right-click **My Computer** (Windows XP and Server 2003) or **Computer** (Vista).
3. In the resulting menu, click **Manage**.
4. In the left-hand panel of the resulting window, expand **Services and Applications**.
5. Click **Services**.
6. In the right-hand panel, find the entry for **Remote Registry**.
7. If the **Remote Registry** service has a status of "Started" and a startup type of "Automatic", it's already set up correctly. You can close the window.
8. Otherwise, right-click on the entry for **Remote Registry**. Click **Properties**.
9. In the resulting window, set **Startup type** to **Automatic**.
10. Click **OK**.
11. Right-click on **Remote Registry** again.

12. Click **Start**.

If all the conditions for displaying the event log are not met, **Administration** | **@Requests** displays an error message.

If you are logged into the computer where the @Requests service is installed, the event log is automatically updated whenever an event occurs. If you are logged into a different computer, the event log is only updated when you click the **Refresh** button.

Administrator Privileges

Normally, you do not need **Administrator** privileges to run MainBoss. However, you do need such privileges in the following cases:

- When you first install the software, if you intend to make it available to everyone on the system
- When you click **Manage** in **Administration** | **@Requests**
- In order to look at the event logs in **Administration** | **@Requests** (if you don't have privileges, the Event Log list will have an error message rather than log messages)

Using Multiple Maintenance Organizations

Some maintenance departments may decide to license multiple maintenance organization databases. For example, a property management company that services multiple independent properties may choose to have a separate database for each property.

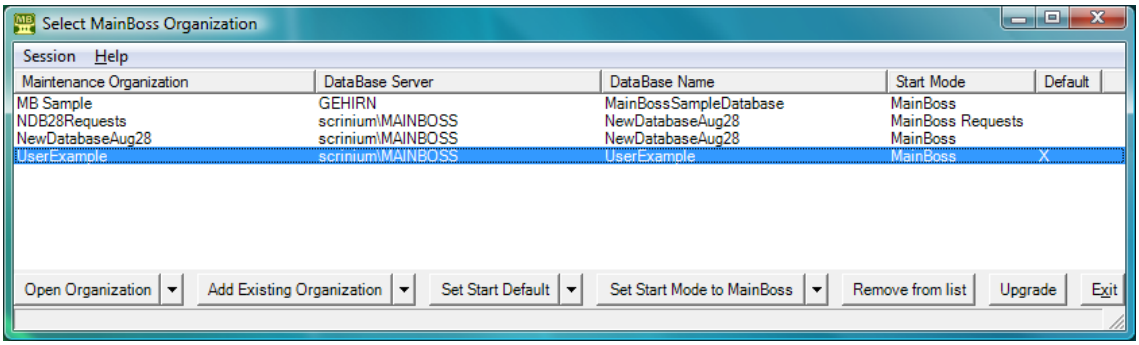
If you have multiple databases, each must have its own separate set of license keys. You may not use the same license keys for multiple databases.

1. You create each new database following the same procedures described in *Creating a Maintenance Organization* on page 4.
2. You then record the database's license keys as described in *Entering License Keys* on page 6.
3. You authorize appropriate users (as described in *Users* on page 8).
4. On every computer where someone will wish to access a particular database, you follow the steps in *Installing MainBoss on Other Computers* on page 11 so that the databases are known on each appropriate computer.
5. If you are using @Requests, each separate database must have its own e-mail address for receiving mailed-in requests. You must also install a separate @Requests service for each database where @Requests will be used. For more information, see *Installing the @Requests Service* on page 13.

You will end up with a list of maintenance organizations you may access. One organization is designated as the default; whenever you start MainBoss on a particular computer, that computer will start with its default organization. To switch to another maintenance organization, go to MainBoss's **Session** menu and select **Change Maintenance Organization**. MainBoss will display a list of available organizations.

To switch to another maintenance organization:

1. Go to MainBoss's **Session** menu and select **Change Maintenance Organization**. MainBoss will display a list of available organizations:



2. Select the organization you want, then click **Open Organization**. MainBoss will open the organization and you can begin working with it.

If you wish to change the default, select the desired database in the above window and click **Set Start Default**. This designates the chosen database as the default. From this point on, MainBoss on this computer will start with the selected default. (Note that each user on each computer can have a different default.)

Start Modes

MainBoss can start in a number of different modes:

MainBoss: Full MainBoss mode with all available modules operational. The maximum number of simultaneous users in full MainBoss mode is limited by your license. Note that if the same person starts two separate sessions of MainBoss, that counts as two simultaneous user sessions. However, inside a single MainBoss session, you can have as many windows open as you want; the only thing that counts is the number of MainBoss start-ups, not the number of windows that are open simultaneously once MainBoss starts.

MainBoss Requests: Requests-only mode. The control panel will only display entries relevant to dealing with requests. You can always open MainBoss in this mode, provided you have a **MainBoss Requests** license key and you have been granted

permissions to access the database. Opening MainBoss in this mode does *not* count against the maximum number of users allowed to connect with the database; therefore, it is recommended for people who only intend to create requests (e.g. help-desk personnel).

Administration: Only allows access to the **Administration** part of the control panel. You can always start MainBoss in this mode, even if you currently have no valid license keys. In fact, when you create a blank organization, you *must* start MainBoss in this mode in order to enter your license keys. Once you've done so, you can quit and start MainBoss again in its normal mode.

View Sessions: Only shows who else is using MainBoss. You can always start MainBoss in this mode, even if MainBoss has reached the maximum number of users for which it's licensed. In this way, you can see who's using MainBoss and can perhaps ask one of them to quit so that you can use MainBoss yourself.

You can specify a start mode in the **Select MainBoss Organization** window. You can set a default mode for any organization by using the **Set Start Mode** drop-down button or by choosing a mode from the **Open Organization** drop-down button.

As an example of the use of start-up modes, your help-desk personnel might set their starting mode so that they always access the database in requests-only mode unless they manually open it in some other mode. The value of this is that requests-only mode doesn't count toward the maximum number of simultaneous users allowed by your **MainBoss** license.

If you sometimes want to open a database in full mode and sometimes in requests-only mode, you can give the same database multiple organization names. For example, suppose your database has the "**DataBaseName**" `MyMainBoss`. You could use **Add Existing Organization** to create two different lines in the organization list:

- One line might have the "**OrganizationName**" `MyMainBoss-Full`, with a start mode of **MainBoss**.
- A second line might have the "**OrganizationName**" `MyMainBoss-Requests`, with a start mode of **MainBoss Requests**.

If you click **Open Organization** on `MyMainBoss-Full`, MainBoss opens in full **MainBoss** mode. If you click **Open Organization** on `MyMainBoss-Requests`, MainBoss opens in requests-only mode.

You can also use command line options to specify start mode and maintenance organization you wish to access. This makes it easy to create shortcuts and desktop icons that each open a different organization or open the same organization in different modes. For more on command line options, see the next section.

Command Line Options

If you will frequently access different maintenance organization, you might want to create a set of desktop icons, each of which starts up MainBoss with a different organization. To create such icons, you need to use MainBoss command line options. These options are described in full detail in the online help, but this guide offers a quick overview.

When MainBoss starts, it usually opens your default database. To tell it to open a different database, use the command line:

`mainboss /on:`*organizationname*

For example,

`mainboss /on:OurOrganization`

opens the database that has the organization name `OurOrganization`.

Start Modes: The command line also lets you specify a particular *mode* in which you want MainBoss to operate. For example, you can start MainBoss in "requests only" mode—this lets the user view and enter requests, but nothing else. Help-desk personnel might run MainBoss in this mode so they can record problem reports but not use other MainBoss features.

To start in requests mode, use the command line

`mainboss /mode:requests`

You can also add options specifying an organization name, as shown previously.

Using Command Line Options: The easiest way to use command line options is to create a desktop icon that specifies the options. You can do this by following these steps:

1. In the standard Windows **Programs** menu, right-click on the entry for **MainBoss 3.0**. In the resulting menu, click **Copy**.
2. Go to the desktop. Right-click on the desktop, then click **Paste Shortcut**.
3. Give the new desktop icon a good descriptive name (right-click on the icon and click **Rename** in the resulting menu).
4. Right-click on the new desktop icon. In the resulting menu, click **Properties**. Windows will open a Properties window for the icon.
5. The "**Target**" field contains the name of the MainBoss executable file in double quotes. After the closing double quote on the file name, add a space and whatever options you wish to specify.
6. Click **OK**.

The icon that you just created will start up MainBoss with the command line options you specified. You can have multiple MainBoss desktop icons, each with a different command line. (Each such icon has to be given a unique name.)

Backups

We strongly recommend that you backup your database on a regular basis. Most commercial backup software packages can tell SQL Server to create a backup of a database, after which the backup package backs up the backup. In this way, backing up your MainBoss database is automatically incorporated into your normal backup procedures.

If you use commercial backup software, check that it can, in fact, deal with SQL Server database. With some backup products, dealing with SQL Server is an "extra" that you have to pay for on top of standard backup operations.

If you do not have commercial software that deals with SQL Server, you can do backups manually. For more information, see *Backups with SQL Server* on page 25.

Appendix A: Setting Up SQL Server

Note: All web addresses given in this appendix are subject to change without notice. If Microsoft does change a web address, go to `microsoft.com` and search for the appropriate page.

MainBoss Advanced needs to use Microsoft SQL Server 2005 SP2 (Version 9.00.3042.00 or greater). If you already have this software installed at your site, MainBoss can use your existing installation. If you do not already have the software, you can purchase SQL Server from Microsoft or download the free version of SQL Server 2005, called SQL Express.

Note: Before you look into obtaining any version of SQL Server, check to see if your company already has one installed. Many organizations may already have SQL Server in connection with some other software package.

There are several important options that should be specified when you install SQL Server. Therefore, make sure to follow the installation instructions given later in this appendix.

Server Hardware Requirements: The machine where SQL Server is running should meet the hardware requirements specified by Microsoft for the version of SQL Server you will be using. If you will be running MainBoss Advanced on the same computer, the machine should also meet the hardware requirements for MainBoss Advanced (stated in the main body of this guide).

Authentication within SQL Server: SQL Server offers two types of user authorization: SQL Authentication or Integrated Authentication. MainBoss requires Integrated Authentication. Mixed mode (accepting both authentication types) will work with MainBoss, but mixed mode is offered only for backward compatibility and Microsoft has deprecated it.

Instances

You can have multiple independent copies of SQL Server running simultaneously on the same computer. Each of these is called an *instance* of SQL Server. Each instance maintains its own separate set of database files.

If you already have SQL Server running at your site, it presumably manages a set of databases. Your IT department may prefer to keep your MainBoss database(s) separate from databases for other software packages. You can do that by creating a new instance of the SQL Server. This can be set up so that it only deals with MainBoss databases.

Different instances of SQL Server are distinguished by having different *instance names*. This is just an identifying name. One instance can have its name blank, in which case it's called the *default instance*. If someone creates a SQL Server database on a computer without specifying an instance name, the database is created and managed by the default instance.

By default, the installation procedures for full versions of SQL Server set up an instance with its name blank. However, for SQL Express, the default installation procedure specifies the instance name `SQLEXPRESS`.

SQL Server Installation

The default installation procedures for full versions of SQL Server are designed to let other computers connect with the Server computer. Therefore, their default options will generally work well with MainBoss.

However, the default installation procedure for SQL Express assumes that it will only be used on a single computer, without users on other computers trying to connect with SQL Server. If you intend to have users on other machines connect with SQL Express, you must change a number of default options during the installation. You must also set several configuration options *after* SQL Express is installed, as discussed later in this section. For more information, see *SQL Express Considerations* on page 26.

Overall, we believe customers will find it easier to use one of the full versions of SQL Server than the free SQL Express—full versions are simpler to install and to maintain (e.g. backing up the database). However, SQL Express is available at no charge and can be used successfully with MainBoss, especially by small maintenance departments.

SQL Server on Windows Vista: If you will be installing SQL Server on Vista, we recommend that you read Microsoft's background material provided at

`http://msdn2.microsoft.com/en-us/library/ms143719.aspx`

In particular, make sure you comply with the specified hardware and software requirements.

Backups with SQL Server

If you have the full SQL Server, the easiest way to implement backups is to use the automated facilities called *maintenance plans*. To set up a maintenance plan, follow these steps:

1. Start SQL Server Management Studio and connect to the SQL Server instance that manages the MainBoss database.
2. In the left-hand panel, expand **Management**.
3. Click **Maintenance Plans**.
4. In the right-hand panel, right-click. In the resulting menu, click **Maintenance Plan Wizard**.
5. The Maintenance Plan Wizard will walk you through the process of setting up automatic backup procedures.

A formal backup using SQL Server Management Studio is better than just saving the "raw" disk file(s) containing the database, since a "raw" save may leave a little bit of data unusable. However, a "raw" save every day is better than a formal save once a week—losing a week's worth of work is worse than losing a transaction or two.

Some commercial backup software packages can work with SQL Server so that SQL Server backups are included when the rest of your computer is backed up. For more information, see the documentation for your backup software.

SQL Express Considerations

SQL Express is a free version of SQL Server 2005. It is available at

http://msdn.microsoft.com/vstudio/express/sql/register/default.aspx

SQL Express will provide acceptable performance for small sites. If you use this approach, you should install both SQL Express and the Management Studio. Larger sites should buy one of the larger versions of SQL Server 2005. For a comparison of the different versions of SQL Server, see

http://www.microsoft.com/sql/prodinfo/features/compare-features.mspx

We should note that SQL Express lacks some features of paid-for versions of SQL Server. For example, SQL Express does not have automatic "maintenance plans", Microsoft's facilities that simplifies automatic backups and integrity checks. However, `expressmaint` is a free software package that provides some of the functionality that's missing in SQL Express. For more information, see

http://www.sqldbatips.com/showarticle.asp?ID=27

The `expressmaint` package is not a Thinkage or Microsoft product. Neither MainBoss Support nor Microsoft will provide any support for this package.

The rest of this appendix describes considerations for installing and working with SQL Express.

Installing SQL Express

The following description assumes that you do not have any instance of SQL Server installed on your computer (either SQL Server 2005 or earlier versions). To determine whether you do have such an installation, go to the Windows Control Panel and check "Add/Remove Programs" (Windows XP and Windows Server 2003) or "Programs and Features" (Windows Vista).

Microsoft recommends that SQL Express be installed with a limited permission set and/or as a limited user. For information on this process, see

http://msdn2.microsoft.com/en-us/library/ms143504.aspx

If you ever intend to have multiple users, we offer the following suggestions as options to choose during installation. Your site might decide to make different choices, but our suggestions should help typical users find their way through the many possibilities.

Unless otherwise stated below, we recommend that you accept the default options in each phase of the installation.

- In the **Registration Information** phase of the installation, remove the checkmark from the **Hide advanced configuration options** box.
- In the **Feature Selection** phase of the installation, expand all the entries in the list of possible installation features. Select **Data Files**, **Shared Tools**, **Connectivity Components** and **Management Studio** for full installation ("Entire feature will be installed on local hard drive").
- In the **Instance Name** phase of the installation, specify a suitable name for this instance of the server. We recommend using `MAINBOSS` to make the server's purpose obvious.
- In the **Configuration Options** phase, checkmark **Add user to the SQL Server Administrator role**.
- In the **Error and Usage Report Settings** phase, choose whatever options you like.
- Proceed with the installation.

The options above constitute a standard configuration for use with MainBoss Advanced. They match Microsoft's recommendations for SQL Server 2005 and for future releases of SQL Server software. If you make different configuration choices, you may find you have to make changes in future; also, if you run into trouble, MainBoss Technical Support may recommend that you reset your SQL Server options as listed above.

SQL Express Installation Notes: After installing SQL Express, you must enable TCP/IP connections. This process is described in this guide, in the section *SQL Server Configuration* on page 34.

If you intend to connect with SQL Express from other computers, you must also adjust the Server computer's firewall to allow remote users to connect with the SQL Server and the SQL Server Browser. This process is explained in *Adjusting the Firewall* on page 35.

Finally, you may have to adjust your local security settings to allow access to shared files. This process is explained in *Local Security Settings* on page 37.

Backups with SQL Express

Maintenance plans aren't available with SQL Express. For information on automatic backups with SQL Server Express, see

http://www.sqldbatips.com/showarticle.asp?ID=27

You can also do backups manually by following these steps:

1. Start SQL Server Management Studio Express and connect to the SQL Server instance that manages the MainBoss database.
2. In the left-hand panel, expand **Databases**.
3. Right-click on the name of the MainBoss database you want to back up. In the resulting menu, click on **Tasks** and then **Back Up**. Management Studio displays a window where you can do the backup.
4. Click **OK**. Management Studio proceeds with the backup.

The above process waits for a moment when MainBoss isn't performing operations on the database, then it temporarily suspends operations while the backup is taking place. This ensures that the database is in a "stable" state—not partway through making changes.

The Management Studio backup operation creates a single file containing the entire contents of the database. This file has a format that facilitates restoration if it ever becomes necessary.

Once the backup file is created, you should make sure it's copied to some other storage location (e.g. a disk on some other computer, a CD, or some other type of backup medium). A backup file stored on the same disk as the database itself won't help you if the disk crashes.

Appendix B: Troubleshooting

This appendix offers some help for determining whether SQL Server and other software are set up properly for use with MainBoss. The appendix also describes tests for checking whether a particular user can access the SQL Server databases. Finally, the appendix provides some tips on how to approach "non-standard" MainBoss configurations if your IT department insists on deviating from MainBoss's recommended set-up.

General Advice

There are many things that might go wrong when trying to use SQL Server, and Microsoft's diagnostic facilities may not be helpful. If an error condition occurs with SQL Server, MainBoss can only display the diagnostic message that SQL Server or Windows provides, and this might not tell you much.

If you get a diagnostic message that doesn't tell you enough, we recommend that you use Google to search `microsoft.com` for references to the *Details* section of message. (If the Details section is blank, use the main part of the message.) For example, if the Details section of the message is `"Some text"`, you would type the following into Google's search field:

```
site:microsoft.com "Some text"
```

The `site:microsoft.com` tells Google that you only want responses from Microsoft's web site. Putting quotes around `"Some text"` means that you search for that exact string.

Note that if the error text contains specific references to names of computers, users, etc., you must remove them before you do the search and only put quote marks around each separate section. Also, if the message is very long, you can just extract important sections. For example, if you get the message

```
Login failed for user 'JSMITH'. The user is not associated
with a trusted SQL Server connection.
Unable to create the base database session
```

you could write up the Google search as

```
site:microsoft.com "Login failed for user" "trusted SQL Server connection"
```

Note: This appendix uses "the Server machine" to refer to the computer where SQL Server 2005 is running and where the MainBoss database is stored. We assume that the MainBoss database already exists, created in accordance with the instructions in the main body of this guide. "The Remote machine" is a different computer from which you want to access the MainBoss database; this computer should be on your local area network (LAN).

Several of the sections in this appendix describe tests you can try if something isn't working. Different tests are certainly possible, but the ones we describe provide a productive route for

checking potential sources of difficulty. All of these tests use standard Microsoft software. If the tests fail somewhere along the way, you can hope that the software will provide an error message that explains what went wrong. You can then fix the problem and try again. Once the test works with Microsoft software, you can try again with MainBoss itself.

The tests described in this section should be carried out in the order given. The purpose of each test is to detect problems in your set-up of Windows, your network, or SQL Server and to determine where the problem lies. It's important to eliminate the possibility of such problems before contacting MainBoss Support—don't mistake a problem in Windows for a problem in MainBoss.

If you have problems with Windows, your network, or SQL Server, contact Microsoft or your software service provider.

File-Sharing

Before other computers can connect with the Server machine, the Server's firewall must be set up to allow file-sharing.

If you have a commercial firewall product (e.g. one that comes as part of an antivirus package), consult the software's documentation for how to allow file-sharing. If you're using the standard Windows firewall, follow these steps:

1. On Windows Vista:
    - In the Control Panel, click **Windows Firewall**.
    - In the resulting window, click **Change settings**.
2. On Windows XP:
    - In the Control Panel, click **Windows Firewall**.
3. A window opens where you can make firewall adjustments. Go to the **Exceptions** section of this window.
4. In the list, go to `File and Printer Sharing`. If the accompanying checkbox isn't already marked, checkmark it.
5. Click **OK** to exit.

This step is required for all the other troubleshooting discussed later in this appendix.

Testing for the Server's Machine Presence

To check connections between the Server and Remote computers, the most basic test is a *ping*:

1. On the Remote computer, open a Command Prompt window by clicking the standard Windows **Start** button, clicking **All Programs**, expanding **Accessories** and clicking **Command Prompt**.
2. Type `ping`, followed by a space, followed by the name of the Server computer. For example, if the Server is called `OurServer`, you'd type

   ```
   ping OurServer
   ```

Wait for results. If the connection can't be made, you'll get an error message or a "time-out" message (i.e. the `ping` command waited a significant length of time but never got an answer). If the connection can be made, you'll get a message saying that the Server computer replied and providing other statistics.

If `ping` fails, check the documentation on your network software to see what's gone wrong.

Testing That a User Can Connect with the Server Machine

This test determines if a particular user on a particular Remote machine can connect with the Server machine, and that the user has appropriate permissions to access the machine.

The user's login name and password must be the same on the Remote machine as on the Server machine. This will automatically be true if you're using domains. If you aren't using domains, have the user attempt to login to the Server machine using his/her login name and password. If this doesn't work, the login name and/or password must be changed so that they're the same on all relevant systems.

> **Note:** If you aren't using domains, users should be warned that if they change their passwords on one computer, they should change the password on all other computers too. This is especially important on the Server machine.

It's important to note that SQL Server has its own security permissions that are separate from Windows permissions. For example, a disk may have general read/write permissions, allowing any user to read or write any file on the disk; still, SQL Server may only permit access to a limited set of users based on login name and password. If you can't connect to SQL Server, here's a way to make sure that a particular user on a Remote machine has login and password set up correctly.

Before you do this, you have to make sure that you don't have any existing connections to the Server computer. To do that, follow these steps:

1. On the Remote computer (where you'll be doing this test), open a Command Prompt window by clicking the standard Windows **Start** button, clicking **All Programs**, expanding **Accessories** and clicking **Command Prompt**.
2. Type `net use` (this displays any existing connections).
3. If there are no connections listed to the Server computer, you don't have to do anything; just close the Command Prompt window.

4. Otherwise, you have to delete all existing connections to the Server computer.

    - First, write down the information shown about connections to the Server computer. You'll need this information later in order to re-establish the connections.
    - If a connection has a letter drive shown in the Local column (e.g. `Z:`), type a line of the form

      ```
      net use Z: /delete
      ```

      where you replace `Z:` with the letter shown.
    - If a connection doesn't have anything in the Local column, it will have a name in the Remote column of the form `\\machinename\folder`. Type

      ```
      net use \\machinename\folder /delete
      ```

      where you replace `\\machinename\folder` with the actual names shown.
    - Repeat the above for every connection to the Server computer. You do *not* have to delete connections to any other computer.
    - When finished, it's useful to leave the Command Prompt window open, since you can use it again later.

Note: The connections that you delete may be needed in order for other software to work. If this is so, you must re-establish the connections after you've done the test described in this section. We'll discuss re-connections below.

Once you've got rid of the connections, you can do the following test:

1. On the Server computer, use Windows Explorer to share a test folder with the rest of the network:

    - In Windows Explorer, create a new folder anywhere on the Server computer's disk drive(s).
    - In Explorer's **Tools** menu, click **Folder Options**.
    - In the **View** section of the resulting window, go down to the bottom of the list of settings and turn off **Use Sharing Wizard** (on Vista) or **Use simple file sharing** (on XP).
    - Click **OK**.
    - Right-click on the folder, then click **Share**.
    - In the **Sharing** section of the resulting window, click **Advanced Sharing**.
    - In the resulting window, checkmark **Share this folder**.
    - Click **Permissions**.
    - The resulting window shows who can access the new folder. Remove all existing permissions (if any) by clicking each name in the list, then clicking **Remove**.

- Once you've removed all existing permissions, click **Add**.
- Under **Enter the object names to select**, enter the login name of the person on the remote computer whose connectivity you're testing. (This should also be a login name on the Server computer.)
- Click **OK**.
- When you return to the previous window, checkmark **Full Control** under the **Allow** column.
- Keep clicking **OK** until all windows close.

2. On the Remote computer, have the user log in using his/her login name and password.
3. On the Remote computer, use Windows Explorer to connect to the shared folder.
4. On the Remote computer, attempt to create a text file in the shared folder. Add text to the file (e.g. using Notepad) and save the file.
5. On the Remote computer, attempt to delete the file you just created.

If you can do all this and you are never asked to enter a login name and password during the process, the user's network permissions are set up correctly. If you can't do one of these steps or if you have to type in a login name and password before one of the steps works, you don't have the right permissions. See *Local Security Settings* on page 37 for suggestions of how to deal with the problem.

Once you've verified that the Remote user can pass these tests, the person who created the original (shared) folder on the Server machine can delete the folder.

Again, we emphasize that setting up a domain will avoid a lot of headaches. Microsoft recommends the use of domains with SQL Server.

If you had to remove any connections before testing the file-sharing, you can reconnect them as follows:

1. On the Remote computer, open a Command Prompt window (or return to the one you already had open).
2. If the original connection had a letter drive (e.g. `Z:`), type the following:

   ```
   net use Z: \\machinename\folder
   ```

   where you replace `Z:` with the letter drive it had previously and replace `\\machinename\folder` with the previous value. You may be asked to enter a login name and password.
3. If the original connection had no letter drive, type

   ```
   net use \\machinename\folder
   ```

   where you replace `\\machinename\folder` with the previous value. Again, you may be asked to enter a login name and password.

4. Repeat the above for every previous connection to the Server computer.

SQL Server Configuration

As noted in *SQL Server Installation* on page 25, the default installation procedures for SQL Server set things up so that SQL Server allows access to users on other computers. Unfortunately, the default installation for SQL Express does *not* set up for this kind of remote access. Furthermore, even full versions of SQL Server may have been configured by someone else to prevent remote access. Therefore:

- The operation described in this section is necessary if you've just installed SQL Express
- The operation may also be necessary if you have a full version of SQL Server that isn't properly configured for the type of remote connections (called TCP/IP) that MainBoss needs

Either way, you must enable TCP/IP connections if they are currently disabled. To do so, follow these steps:

1. In the Windows **Start** menu, locate **Microsoft SQL Server 2005**. Under this entry, locate **Configuration Tools** and then **SQL Server Configuration Manager**. This opens a window where you can configure your SQL Server.

> If you can't find an entry for **SQL Server Configuration Manager**, you'll have to add the software. See "Maintenance Installs" below.

2. In the left-hand side of the window, expand **SQL Server 2005 Network Configuration**.
3. In the resulting expansion, click **Protocols for MAINBOSS**. (This will have a different name if you chose a different instance name.)
4. If MainBoss will be used on a network, with or without a domain, right-click on **TCP/IP** in the right-hand side of the window and click **Enable** (if this isn't already enabled). You do not have to do this if you'll only be using MainBoss on a single computer.

   A message will appear saying that your changes will only take effect once you stop and restart the service. The steps below do exactly that.

> The list of protocols also offers the possibility of enabling SQL Server access through named pipes. However, Microsoft recommends that you don't use named pipes across a network.

5. In the left-hand side of the window, click **SQL Server 2005 Services**. You should see an entry for **SQL Server (MAINBOSS)**. (If you used a different instance name than **MAINBOSS**, that name will be displayed instead.) Right-click on this entry, then click **STOP**. (This will actually stop both instances displayed in the right-hand side of the window.)
6. Right-click on **SQL Server (MAINBOSS)** and click **START**.

7. Right-click on **SQL Server Browser** and click **Properties**. This opens a window where you can set the browser's properties.
8. In the **Service** section of the properties window, click **Start Mode**.
9. Drop down the associated arrow (at the end of the line) and click **Automatic**.
10. Click the window's **Apply** button.
11. In the **Log On** section of the same properties window, click **Start** to start the browser again.
12. Click **OK** to close the window.

Maintenance Installs: When your SQL Server was installed, it's possible that SQL Server Configuration Manager was omitted—its installation is optional. If you're missing the configuration manager, you'll have to do a *maintenance install* to add the software.

To do this, start the SQL Server or SQL Express installation again. Tell the installation procedure that you want to change the installation, then follow the suggestions given in *Installing SQL Express* on page 26.

Adjusting the Firewall

In order for other computers to access SQL Server on the Server machine, you must configure the Server machine's firewall to allow such access. (If SQL Server is already in use at your site, the Server machine is probably configured appropriately all ready.)

The following considerations apply:

- If you're running an anti-virus program, it has probably installed its own firewall. Check the program's documentation for how to grant access to SQL Server.
- If you're using the Windows XP or Windows Server 2003 firewall, check the MainBoss web site for a program that will create a rule to allow access through the firewall. You can determine the pathname as follows:

1. Start the SQL Server Configuration Manager on the computer where SQL Server is running.
2. In the left-hand panel, click **SQL Server 2005 Services**.
3. In the right-hand panel, right-click the entry for **SQL Server (MAINBOSS)** or whatever entry there is for SQL Server itself.
4. In the resulting menu, click **Properties**.
5. In the properties window, go to the **Service** section.
6. Double-click on the **Binary Path** entry. Windows will display the name of the SQL Server executable file; grant access through the firewall to this file, as described in the next section.

7. Repeat steps 3-6 for **SQL Server Browser** in the right-hand panel.

Granting Access to a Program through the Firewall

If you have commercial firewall software (typically part of an antivirus package), follow the software's instructions to let SQL Server and the SQL Server browser through the firewall (as discussed in the previous section). If you use the built-in Microsoft firewall, follow these steps.

1. On Windows Vista:
    - In the Control Panel, click **Windows Firewall**.
    - In the resulting window, click **Change settings**.
2. On Windows XP:
    - In the Control Panel, click **Windows Firewall**.
3. A window opens where you can make firewall adjustments. Go to the **Exceptions** section of this window.
4. Check in the list to see if `sqlbrowser.exe` and `sqlservr.exe` are already there and checkmarked. If so, you're finished—click **OK** to exit.
5. If the names are there but not checkmarked, checkmark them then click **OK** to exit.
6. If one or both names are missing, click **Add Program**.
7. In the resulting window, click **Browse**.
8. In the resulting window, find the name of the SQL Server executable file (see the previous section of this guide for how to find this name). Click the file name, then click **Open**.
9. You'll return to the previous window. Click **OK**. The file is added to the list.
10. Repeat steps 6-9 for the SQL Server browser. Click **OK** when you're done.

Scopes: Each program in the Windows Firewall list has an associated *scope* (unrelated to scopes in MainBoss). The scope may be any of the following:

- **Any computer**, in which case the firewall lets the program be used by any computer anywhere (including those on the Internet).
- **My network**, in which case the firewall only lets the program be used by computers in your local area network.
- **Custom list**, in which case the firewall only lets the program be used by a specified list of computers (e.g. ones identified by IP number).

Most sites will wish to restrict MainBoss access to users on your local area network (**My Network**), not to users anywhere on the Internet. If you open up access to all Internet users, you must be sure that you've protected yourself with appropriate security measures.

Your usage requirements may change with time. For example, you may originally restrict MainBoss access to **My network**. Later on, you might decide that you want to be able to connect from outside sites (e.g. when working at home or at a remote job site), in which case you'll have to change your configuration.

To see the scope associated with a program (and possibly to change it), follow these steps:

1. Follow steps 1-3 listed earlier in this section.
2. In the **Exceptions** section of the firewall settings window, click the entry for `sqlservr.exe`.
3. Click **Properties**.
4. In the resulting window, click **Change Scope**.
5. In the resulting window, check the scope. If it isn't what you want, change the setting.
6. Click **OK** twice to return to the firewall settings window.
7. Repeat steps 2-6 for `sqlbrowser.exe`.

Local Security Settings

If you are not using domains and you intend to use MainBoss on multiple machines, you may have to change the security settings on the computer where SQL Server is running. **This is unnecessary if you're using domains.**

In many cases, the security settings will already be correct by default. However, some versions of Windows XP (and possibly Vista) do not have the correct settings. Also, if you upgrade your version of Windows, the upgrade process retains your old settings, even if they're different from the usual defaults of your new operating system.

To set up the correct settings, follow these steps on the computer where SQL Server is running:

1. In the Windows Start menu, click **Control Panel**.
2. In the resulting menu, click **Administrative Tools**.
3. In the administrative tools menu, click **Local Security Policy**. This opens a window where you can change the security settings.
4. In the left-hand panel of the window, expand **Security Settings**, then expand **Local Policies**, then click **Security Options**. MainBoss opens a list of security options you can set.

5. Double-click on the entry for **Network access: Sharing and security model for local accounts**. This opens a window where you can set the option.
6. In the drop-down list, choose the entry for **Classic - local users authenticate as themselves**.
7. Keep clicking **OK** until you've closed all the windows.

The alternative to **Classic** is that anyone logging into the computer will end up as a user called `Guest` (provided that `Guest` is enabled which is not true on many systems). Such a user would not be able to use MainBoss unless you authorized `Guest` as a MainBoss user...but then *anyone* could use MainBoss since `Guest` usually requires no password.

Authorizing Users in SQL Server

If you have checkmarked the option **MainBoss manages SQL Security** in the **Defaults for User** section of **Administration** | **Users**, then MainBoss automatically gives new users appropriate permissions to access the MainBoss database through SQL Server. If you have turned off this option, you must authorize users manually. You might also have to authorize a user manually if someone else (or another program) has de-authorized the user for some reason.

> In order to follow the steps described in this section, you must have SQL Server Administrator permissions on the SQL Server that manages MainBoss.

The following steps describe how to check whether a user has appropriate permissions for accessing the MainBoss database through SQL Server. The steps also describe what to do if permissions for a user have somehow been removed:

1. On the computer where SQL Server is running, start SQL Server Management Studio.
2. When Management Studio asks to specify the "**Server Name**", give the name of the instance of SQL Server that holds the MainBoss database. (This is the same server name you specified when you created the database.)
3. Click **Connect**. Management Studio will open a window showing the configuration of the given instance of SQL Server.
4. In the left-hand panel, expand the entry for **Security**.
5. Under **Security**, click **Logins**. Management Studio will display a list of authorized users. This list may contain user groups as well as the login names of individual users.
6. If the user you want to authorize isn't currently in the list:
    - Right-click **Logins**.
    - In the resulting menu, click **New Login...** Management Studio will open a window where you can authorize a new user.

- In "**Login name**", enter the login name of the person you want to authorize. This must be a valid login name on the computer where SQL Server is running.
- Click **OK**.

7. Once the user's name is on the list of logins, you must check that the user has correct permissions for accessing the MainBoss database. Right-click the name of the user in the right-hand panel, then click **Properties**. MainBoss opens a window providing information about the user.
8. In the left-hand panel of the "Login Properties" window, click **User Mapping**.
9. In the right-hand panel, checkmark the entry for the MainBoss database (if it isn't already checkmarked). Management Studio will highlight the line and put the user's login name in the "User" column.
10. The entry in the "Default schema" column should be "`dbo`". If it isn't:
    - Click the drop-down button (...) at the end of this line. Management Studio opens a window where you can specify a default schema.
    - Under "Enter the object names to select", type "`dbo`".
    - Click **OK**. The original line should now give the name of the MainBoss database, the user's name, and "`dbo`".
11. Under "Users mapped to this login", make sure the name of the MainBoss database is highlighted.
12. Under "Database role membership for:", checkmark **MainBoss** (if it isn't already).
13. In the left-hand panel, click **Status**.
14. Click **Grant** and click **Enabled** (if they aren't already).
15. Click **OK**.
16. Repeat Steps 6-15 for every user you want to authorize.

Remember that the name you specify must be a valid login name for the current computer. When logging in from other computers, users must have the same name and password as on the computer where SQL Server is running. (This will always be true if you use domains, since the whole point of domains is to let users have the same name and password on multiple machines.)

Spelling: If a particular user can't access the MainBoss database, always check that the user's name is spelled correctly in the various places it should appear, e.g. MainBoss's **Users** table and SQL Server's **Logins** list.

Groups: SQL Server lets you grant permissions to user groups as well as individuals. Therefore, you might choose to use the above procedure to grant permissions to a "MainBoss User" group. Once you do that, you can add new users to the group and they automatically

receive SQL Server permissions on the MainBoss database. However, the **Users** table inside MainBoss *only* allows individual login names, not groups.

Note that you might be tempted to set up SQL Server permissions so that *anyone* can access the MainBoss database. However, MainBoss itself will only work for people explicitly authorized in the MainBoss **Users** table. If you grant SQL Server permissions to all users, you end up with the undesirable situation where some people may be prevented from using MainBoss itself, but could still change the database "by hand" (e.g. with Microsoft Access) if they wanted to. If someone can access the database, it's best to make sure that they do it by using MainBoss, not some other piece of software.

Deleting Users: When you add a user to **Administration** | **Users**, MainBoss grants that user permission to connect with SQL Server (if you've turned on the **MainBoss manages SQL Security** option). However, when you delete a user from the **Users** table, MainBoss does *not* delete the user's permission to connect with SQL Server—the user may need "Connect SQL" permission in order to use SQL Server with another software package.

If you wish to completely remove a user's access to SQL Server, remove the user's name from SQL Server's **Logins** list, using SQL Server Management Studio.

Testing Whether a User Has Access to SQL Server and the MainBoss Database

In order to access the MainBoss database from a Remote machine, a user must have appropriate permissions with SQL Server. The following procedure can be used to determine if a user has these permissions. (In order for this to work, the user who performs the test must have administrator privileges, i.e. be part of the `Administrators` group.)

1. Login to the Remote machine in question under the login name you want to test.
2. Click the usual Windows Start button, then click **Control Panel**.
3. In the resulting menu, click **Administrative Tools**.
4. In the Administrative Tools menu, click **Data Sources (ODBC)**. (On Vista, you may be asked for permission to continue. Click **Continue**.)
5. In the resulting window (ODBC Data Source Administrator), go to the **System DSN** section.
6. Click **Add**.
7. In the list, click **SQL Server** (typically near the bottom of the list), then click **Finish**.
8. Windows opens a box where you can create a new data source for SQL Server. For **Name**, specify anything (e.g. `TEST`). In **Server**, pick the SQL Server where MainBoss resides.
9. Click **Next**, then click **Next** again.
10. Checkmark **Change the default database to:**. In the resulting drop-down list, pick the name of the MainBoss database on the SQL Server system.

11. Click **Next**, then click **Finish**.

12. Click **Test Data Source**.

You should end up with a window saying that the test completed successfully.

Once you get the message that the test completed successfully, you can click **OK** twice. This gets you back to the ODBC System Administrator window. At this point, you can **Remove** the data source that you just added. This doesn't do anything to the MainBoss database, it just removes the entry from the list. Click **OK** to finish up.

Remember, if you get an error during this process, use Google to look up the error message: you may get useful tips about what went wrong.

Testing Database Access with Microsoft Word

You can test your access to the MainBoss database by using Microsoft Word. We suggest this simply because many computers have Word installed; you can also use Microsoft Access or Microsoft Excel in a manner similar to what we'll describe, but since Word is most common, we'll use it.

Any user can perform this test, but the Word software must be installed on your system.

1. Login to the Remote machine in question under the login name you want to test.
2. Start Microsoft Word.
3. If you have Office 2003:
    - In Word's **View** menu, click **Toolbars** and checkmark **Database** (if it isn't already checkmarked). This opens the database toolbar.
    - In the database toolbar, click the icon for **Insert Database**.
    - Word opens a window with the button **Get Data**. Click this button.
4. If you have Office 2007:
    - In the menu bar, click **Mailings**.
    - Click **Select Recipients**.
    - Click **Use Existing List**.
5. Word opens a window for you to specify a data source. Click the **New Source** button near the bottom of the window.
6. Word opens a wizard to walk you through the data access process. Under **What kind of data source do you want to connect to?** click **Microsoft SQL Server**.
7. Click **Next**.

8. Under **Server Name**, enter the name of the SQL Server that holds the MainBoss database. Click **Next**.
9. Select the name of the MainBoss database from the drop-down list.
10. The wizard displays a list of tables in the MainBoss database. Since this is just a test, you can select any table; **License** is a good choice since it always contains a few lines of data, but not too much. Select this table.
11. Click **Next**, then click **Finish**.
12. Word returns to the Data Source window, with the file name filled in. Click **Open**.
13. Word returns to the original "Get Data" window. Click **Insert Data**.
14. Click **OK**.

If all goes well, Word will insert a table containing your license keys and other data. If this process works, the user has all the correct permissions to access the MainBoss database. If not, error messages from Word may help you determine what went wrong. You may also find the following article useful:

http://www.microsoft.com/technet/prodtechnol/sql/bestpractice/CISQL2005ASCS.mspx

Testing Database Access with SQL Server Management Studio Express

If you don't have Microsoft Word on a particular system, you can test database access with SQL Server Management Studio Express, available free for download from Microsoft:

https://www.microsoft.com/downloads/details.aspx?familyid=C243A5AE-4BD1-4E3D-94B8-5A0F62BF7796&displaylang=en

Follow these steps:

1. Have a privileged user install SQL Server Management Studio Express on the Remote computer where you wish to test access.
2. Under your own login name, start Management Studio Express.
3. When the software asks you to specify a server name, enter the name of the Server computer, followed by a backslash, followed by the SQL Server instance name where MainBoss is running, e.g. `MYSERVER\MAINBOSS`.
4. Click **Connect**.
5. In the left-hand panel, expand the entry for **Databases**.
6. Expand the entry for the MainBoss database.
7. Expand the entry for **Tables**.
8. Expand the entry for **dbo._DAccessCode**.
9. Expand the entry for **Columns**.

If you can see entries for **Code** and other data fields, SQL Server is allowing you to look at the database. (You don't need to see the actual database values.)

Testing POP3/IMAP4 Permissions for @Requests

The **@Requests** module needs to obtain mail messages from its mailbox using either the POP3 or IMAP4 protocols (techniques for accessing a mailbox). Some sites configure their systems to prevent POP3 and/or IMAP4 from being used. Therefore, it's useful to test whether these protocols are usable on the computer where you've installed the @Requests service (see *Installing the @Requests Service* on page 13). The easiest way to do this is to try to create a POP3 or IMAP4 mailbox.

Different mail software products have different ways of doing this. If you have Microsoft Outlook or Outlook Express, you can do the following:

1. In Outlook's **Tools** menu, click **E-Mail Accounts**.
2. In the resulting wizard, click **Add a new e-mail account**.
3. Click **Next**.
4. In the resulting window, click **Add**.
5. Click either **POP3** or **IMAP** (whichever you're using for @Requests).
6. Click **Next**.
7. If you have configured @Requests to use POP3S or IMAPS (which is recommended), checkmark **Log on using Secure Password Authentication (SPA)**.
8. Continue to fill in information as the wizard requires. Make sure that you use the same port numbers that you specified when configuring @Requests.
9. When you've finished creating the account, send mail to it and see if the mail arrives.

As noted, some sites have set up their e-mail systems to prevent POP3 and/or IMAP4, since these protocols can create security holes if used in the wrong context. This has led some sites to ban the protocols entirely, even though they are safe when used in the right context. If your site has banned the protocols, the information in

`http://www.mainboss.com/english/products/pop3.shtml`

may help convince your IT department that the protocols can be allowed for MainBoss.

On Vista, if you don't have Outlook, you can use the built-in Windows Mail software. The process for adding a new account isn't exactly like the procedure above, but it's similar.

Testing Access to the @Requests Event Log

In order to see the @Requests event log, various conditions have to be met (as explained in *The @Requests Event Log* on page 17). You can test these conditions as follows:

1. Click on the Windows **Start** button.
2. Right-click **My Computer** (Windows XP and Server 2003) or **Computer** (Vista).
3. In the resulting menu, click **Manage**.
4. In the left-hand panel of the resulting window, right-click **Computer Management (Local)**.
5. In the resulting menu, click **Connect to another computer**.
6. In the resulting window, fill in "**Another computer**" with the name of the computer where @Requests is installed.
7. Click **OK**.

If this works (i.e. if you don't get an error message and you can see useful information in the next screen), then you have appropriate permissions to see the @Requests event log. The @Requests computer and the computer where you are logged in are both configured to allow the event log to be seen.

Testing that You Can Start and Stop the @Requests Service

As discussed in *Installing the @Requests Service* on page 13, you can start and stop the @Requests service from a computer other than the one where the @Requests service is installed, provided you have appropriate permissions and the @Requests computer is appropriately configured. As a quick test for checking whether these conditions are met, follow these steps:

1. Open a Command Prompt window by clicking the standard Windows **Start** button, clicking **All Programs**, expanding **Accessories** and clicking **Command Prompt**.
2. Type `sc`, followed by " `\\`", followed by the name of the computer where @Requests is running. After this type a space followed by `query`. For example, if the @Requests computer is called `AtRequests`, you would type

   ```
   sc \\AtRequests query
   ```

If this works correctly, you'll get a list of all the services running on the @Requests computer. This means you can start and stop services on that computer. Otherwise, you'll get an error message, indicating that conditions for starting and stopping services haven't been met.

General Troubleshooting

In addition to all the specific troubleshooting points listed in the appendix, there are a number that always apply to Windows programs.

- Check for viruses. Viruses can cause almost any kind of strange behavior.
- Check your disks for hardware problems. The most thorough way of doing this is using the `CHKDSK` command. In a command prompt window, type

  ```
  chkdsk /r /f
  ```

  Then reboot your computer. The boot process will do a lengthy but thorough check to find disk errors and recover from them.
- Check your event logs. You do this starting with the **Control Panel**. Go to **Administrative Tools**, then select **Event Viewer**. Check especially for errors and warnings—these may indicate problems with your computer (e.g. hardware errors or viruses).

  If you expand **Window Logs** in the left-hand panel and then click **System**, the center panel will list recent "events" for your computer. Any marked with red "error" signs are particularly serious and may indicate critical problems with your hardware or software. Remember, use Google to look up any error messages.

INDEX