

MainBoss Advanced 3.2, Update 2 Installation and Administration

Copyright © 2009, Thinkage Ltd.

Permission granted to reproduce all or part of this document
provided that Thinkage's copyright notice is included in the reproduction.
No money may be charged for such reproductions except
to recover all or part of the cost of reproduction.

Table of Contents

MainBoss Advanced Installation and Administration Guide	1
Software and Hardware Requirements.....	1
Microsoft SQL Server	2
Microsoft Report Viewer.....	3
Installation Locations for MainBoss Software.....	3
ClickOnce Deployment Introduction.....	4
Manual Installation.....	5
Manual Installation Overview	5
Manually Installing MainBoss Software	6
ClickOnce Installation.....	6
ClickOnce Deployment Details	7
Creating a Maintenance Organization.....	9
Entering License Keys.....	10
Users.....	12
Registering New Users with MainBoss.....	14
Security Roles	16
Testing Security Roles.....	18
Roles for Administering MainBoss	18
Installing MainBoss on Other Computers.....	18
Installing the @Requests Service.....	20
Configuring @Requests.....	22
Starting and Stopping the @Requests Service	25
Uninstalling @Requests	26
The @Requests Event Log.....	26
Windows Administrator Privileges	28
Transferring a Database to a New Computer	28
The @Requests Service and MainBoss Upgrades.....	30
Using Multiple Maintenance Organizations	30
Start Modes.....	31
Command Line Options.....	32
Backups	34
Backups within MainBoss	34
Restoring a Backup Within MainBoss	37
The MBUtility Command	39
Getting Help for MBUtility	40
Listing MainBoss Organizations	40
Built-In Data Schemas.....	41
Exporting Data from Your Database	41

An Example of Exporting to Excel 2007	43
Importing Data into Your Database.....	44
Adding Organizations.....	45
Backing Up Your Database with MBUtility.....	46
Adding an Administrator to a Database	47
Appendix A: Setting Up SQL Server	49
Instances	50
SQL Server Installation	50
Backups with SQL Server	51
Using SQL Server to Restore a Database from Backup	52
SQL Express Considerations.....	52
Installing SQL Express	53
Backups with SQL Express	54
Upgrading from SQL Server 2005 to SQL Server 2008	54
SQL Server Maintenance	55
Appendix B: Troubleshooting.....	57
General Advice.....	57
File-Sharing.....	58
Testing for the Server's Machine Presence.....	58
Testing That a User Can Connect with the Server Machine	59
Comparing the DNS Name and Active Directory Name	62
SQL Server Configuration	62
Adjusting the Firewall.....	65
Granting Access to a Program through the Firewall.....	66
Local Security Settings.....	67
Authorizing Users in SQL Server	68
Testing Whether a User Has Access to SQL Server and the MainBoss Database...70	
Testing Database Access with Microsoft Word.....	71
Testing Database Access with SQL Server Management Studio Express	72
Testing POP3/IMAP4 Permissions for @Requests	73
Testing Access to the @Requests Event Log.....	74
Testing that You Can Start and Stop the @Requests Service.....	75
General Troubleshooting.....	75
Appendix C: MainBossRemote.....	76
MainBossRemote System Requirements	76
License Keys	76
Initial Set-Up.....	77
Security Considerations.....	77
Application Pools.....	79
Web Sites vs. Applications	81
Adding and Configuring an Application.....	81

Adding and Configuring a Web Site	82
Authentication Settings	83
Application Settings	83
Software Initialization	84
Accessing SQL Server on a Different Computer	84
MainBossRemote Troubleshooting.....	85
Cookie Usage	86

Introduction

MainBoss Advanced Installation and Administration Guide

This guide explains the process of installing MainBoss Advanced. 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 3 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 (with Service Pack 2 or later); or
- Windows Vista Business or better (with Service Pack 1 or later); or
- Windows Server 2008 or later; or
- Windows 7 Pro or better.

Both 32-bit or 64-bit versions of these systems are acceptable.

We do not recommend Windows XP Home, Windows Vista Home, Windows Vista Basic, or Windows 7 Home.

.NET Framework and Report Viewer: Every computer where MainBoss will run must have the .NET framework installed (version 3.5 SP1 or later). If you attempt to install MainBoss on a computer which doesn't have an appropriate of .NET, the MainBoss installation procedure will automatically attempt to download and install the latest .NET software from Microsoft's web site.

Every computer that runs MainBoss will also need Microsoft Visual Studio 2008 Report Viewer. For information on downloading this software from Microsoft, see *Microsoft Report Viewer on page 3* in this guide.

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, Windows Server 2008, and Windows 7: 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, 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.

Small Business Server: 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 Small Business Server Premium, which includes SQL Server 2008 as part of the package.

Microsoft SQL Server

MainBoss Advanced uses Microsoft's SQL Server 2005 or 2008 to read and write all data. SQL Server 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), or
- SQL Server 2008 SP1

MainBoss doesn't work with earlier versions of SQL Server.

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, 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 and 2008 are actually families 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 available from Microsoft. For more information, see *SQL Express Considerations on page 52*.

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 or Server 2008 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. This is discussed in the Appendix.

Microsoft Report Viewer

MainBoss prepares reports using Microsoft's Report Viewer. When you install MainBoss, the installation procedure automatically installs an English-language version of the software. You can replace this with versions in other languages by going to

<http://www.microsoft.com/downloads/details.aspx?FamilyID=cc96c246-61e5-4d9e-bb5f-416d75a1b9ef&displaylang=en>

Choose from the list of available languages, then download and install the software. For non-English support, you must also download the .NET 3.5 SP1 Language Pack from

<http://www.microsoft.com/downloads/details.aspx?FamilyID=8489ed13-b831-4855-96f7-dd35e4c02a20&DisplayLang=en>

Again, choose from the list of available languages, then download and install the software.

Note: SQL Server and Report Viewer are independent of MainBoss, including language support. In particular, reporting facilities will be in the language of your report viewer, which may or may not be the same language used by MainBoss. Similarly, diagnostic messages generated by SQL Server may or may not be in the same language as MainBoss.

Since Report Viewer is Microsoft software, Thinkage takes no responsibility for any problems you might encounter with the Report Viewer software package.

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 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. However, a number of administrative operations are more straightforward if MainBoss *is* installed on the server 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 either:

- Install the MainBoss software on every computer where you intend MainBoss to run; or
- Use ClickOnce deployment, as discussed in the next section

ClickOnce Deployment Introduction

This version of MainBoss supports *ClickOnce Deployment*. This is a Microsoft technique which eliminates the need to install MainBoss updates on user's computers. Essentially, you create a web page which has two links: one that invokes an initial setup program, and one that invokes MainBoss itself.

- When a user runs the setup program, it checks the user's computer to make sure that the user has all necessary prerequisite Microsoft software (including .NET 3.5 and Report Viewer). If one or more prerequisites are missing, the set-up program can download the appropriate software from Microsoft and install it; however, this requires Windows Administrator privileges, so your IT department may have to do this step.
- When the prerequisites are in place, the set-up program installs MainBoss on the user's computer. Once the software has been installed, the set-up program starts MainBoss running.
- From that point on, users will start MainBoss either using the entry in the Windows Start menu or using the other link on the ClickOnce web page: the link that invokes MainBoss directly.
- When you receive a new update of MainBoss, you make the new software available to the web page.
 - If the update doesn't have any new prerequisites, users will automatically get the updated software when they click the link that invokes MainBoss.
 - If the update does have new prerequisites, they'll receive the message "System update required" when they try to run MainBoss. The user then needs to run the set-up program again before running MainBoss itself.

With this approach, you don't have to manually install MainBoss on your users' computers. Users just go through the ClickOnce deployment web page, which automatically installs new software when it becomes available.

For technical reasons, you must still install MainBoss manually on the computer that will run the @Requests service—you cannot use the ClickOnce approach. For more details, see [Installing the @Requests Service on page 20](#).

Full technical details on ClickOnce Deployment are given in [ClickOnce Installation on page 6](#).

Manual Installation

The sections that follow deal with manual installation, as opposed to ClickOnce Deployment.

Even if you use ClickOnce deployment, you must manually install the MainBoss software on the computer that will run the @Requests service (if you've licensed the @Requests module).

Manual Installation Overview

To install and configure MainBoss, some operations must be done by someone who has Windows Administration privileges (typically someone from the IT department). Other operations must be done by someone with SQL Server Administration privileges. Finally, some operations must be done within MainBoss by someone with the MainBoss Administration security role. (In some organizations, this will be the same person.)

Below we list the recommended order of operations for installation and initial set-up, as well as the privileges needed for each step (if any).

1. Install SQL Server 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*. [Windows Administration privileges]
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. [Windows Administration privileges]
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 9](#). [SQL Server Administration privileges; anyone who creates a MainBoss database is automatically granted the MainBoss Administration security role on that database.]
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](#). [SQL Server Administration privileges]

5. Enter the license keys you were given when you licensed MainBoss Advanced. For more information, see *Entering License Keys on page 10*. [MainBoss Administration security role]
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 12*. [MainBoss Administration security role]
7. Once you have MainBoss Advanced working correctly on one computer, either set up ClickOnce deployment or install the MainBoss software on all other computers where you'll wish to use MainBoss. [Windows Administration privileges]
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 18*. [Any user can add a MainBoss database to his/her list of known databases. However, a user's access to a database is controlled by the user's security roles.]
9. If you have licensed the **@Requests Service** module, install @Requests as described in *Installing the @Requests Service on page 20*. [Requires both Windows Administration privilege and the MainBoss Administration security role]

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 49* and *Appendix B: Troubleshooting on page 57*, especially if you'll be using SQL Express.

Manually Installing MainBoss Software

MainBoss Advanced is currently available on DVD or through download from the MainBoss web site. Even if you have the DVD, we recommend that you check the web site to make sure you get the most up-to-date version of the software. To install the software, follow the instructions given on the download web page.

Once you've installed the MainBoss software on a user's system, the software can be started through the standard Windows program menu. Typically, however, you should create a desktop icon that starts the program.

ClickOnce Installation

ClickOnce deployment is a facility created by Microsoft to make it easier for IT departments to ensure that all users are using the same version of a software package, without forcing IT personnel to install the software separately on many computers.

ClickOnce is simple to use but complicated to set up. It requires an IT department with a high degree of technical sophistication. For example, implementing ClickOnce requires the ability to set up a web site containing various special files and a knowledge of how to apply a digital signature to an executable file. If your IT department does not have such expertise, use manual installation instead (as described in *Manual Installation on page 5*).

For an introduction to ClickOnce, see

<http://msdn.microsoft.com/en-us/vbasic/ms789088.aspx>

ClickOnce Deployment Details

Setting up ClickOnce deployment follows these steps:

1. Install SQL Server 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. Create a virtual directory on your organization's web site to contain all the files that MainBoss needs for ClickOnce deployment. We will call this the MainBoss ClickOnce directory.
3. Copy all the necessary files into the MainBoss ClickOnce directory. If you are working from a MainBoss DVD, these files will be in a folder named ClickOnce on the DVD. If you are downloading the software from the MainBoss web site, follow the directions on the web site.
4. One of the files you just copied is named `setup.exe`. This file must be modified to refer to the URL of the MainBoss ClickOnce directory. To do this, open a command window, `cd` to the directory, and execute the command line

```
setup -url=URL
```

where *URL* is the URL of the MainBoss ClickOnce directory. For example, if the URL is `http://xyz.com/mainboss/`, you would execute the command

```
setup -url=http://xyz.com/mainboss/
```

5. Once you have modified the `setup.exe` program to refer to the correct URL, you must digitally sign the `setup.exe` executable file to give it a trusted digital signature. Your IT department should know how to do this.
6. In the MainBoss ClickOnce directory, create a web page with two links: one to the `setup.exe` program and another to `mainboss.application`. Users should click on the `setup.exe` link to initially install the MainBoss software; from then on, they should click on the `mainboss.application` link whenever they want to start MainBoss.

The files supplied by Thinkage contain a `default.htm` file which you can modify to create the needed web page. (When you modify this web page, you should specify the

database server and database name for your MainBoss database.) Alternatively, you can write your own web page from scratch.

When a user runs `setup.exe` for the first time, the program checks to see if the user's computer contains all the prerequisites for MainBoss, including:

- Windows Installer 3.1
- .NET Framework 3.5 SP1
- Microsoft Visual Studio 2008 Report Viewer

If one or more of these is missing, `setup.exe` attempts to install them. In order for the installation to work, `setup.exe` will need the user name and password of someone with Windows Administrator privileges. This means that a Windows Administrator may have to go around to each user's computer to make sure that the prerequisite software is installed.

Once `setup.exe` has verified that the prerequisites are in place, the program will invoke MainBoss. This does *not* require Administrator privileges.

If a user attempts to run MainBoss through the ClickOnce web page without having the appropriate prerequisites, MainBoss will display the message "System update required".

We recommend that you start up MainBoss on one computer and proceed to set up the database as explained in [Creating a Maintenance Organization on page 9](#) and in the sections that follow.

Important: If you have licensed the **@Requests Service** module, the computer where the service will be running cannot use ClickOnce deployment—on that computer, MainBoss must be manually installed.

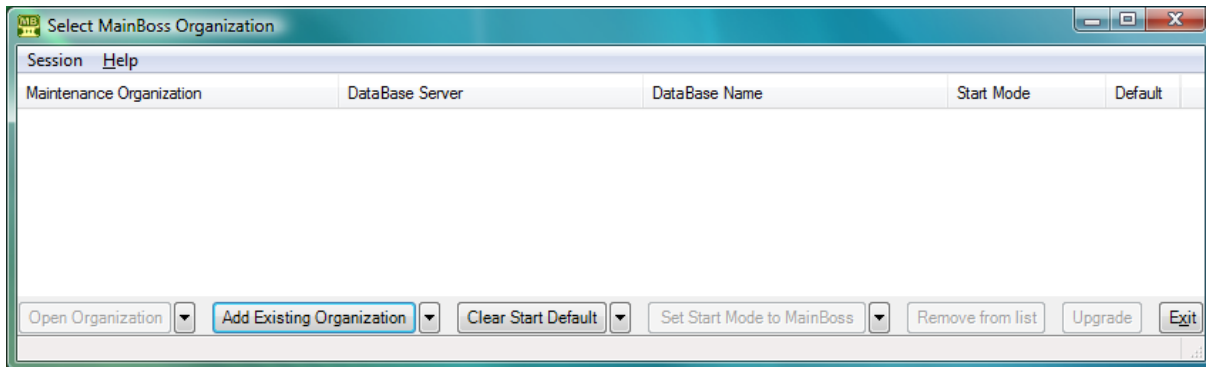
Caveats about ClickOnce: If several people use ClickOnce to run MainBoss on the same computer, the MainBoss software will be installed once for each such user. For example, if three people use the same computer at different times and all of them run MainBoss, the MainBoss software will be installed separately for each person.

If you have a large number of users, network bandwidth may suffer on the day a new release of MainBoss comes out. ClickOnce will download the new software for each user who starts MainBoss; therefore, if you have many users who all start work at 9:00 a.m. on the day after the release becomes available, ClickOnce will try to download a copy of the update for each person who tries to start up MainBoss.

People who invoke MainBoss through ClickOnce will not be able to specify command line arguments to the MainBoss software. They will also not be able to run the MBUtility program (discussed later in this guide).

Creating a Maintenance Organization

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

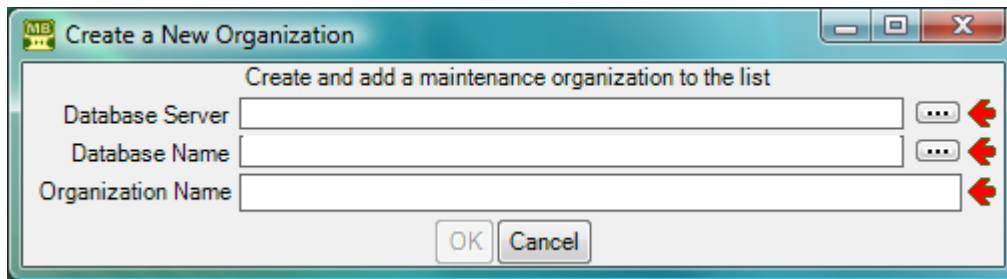


In order to create a MainBoss database, you must be logged into an account that has SQL Server Administrator privileges on the SQL Server that will manage the MainBoss database.

Important: MainBoss’s language support mostly adapts to the personal preferences of each individual user, as expressed in the user’s “Regional Settings”. However, a few key terms are generated and stored in the database at the time the database is created. If possible, these terms are translated into the preferred language of the person who creates the database; for example, if the database creator has set “Regional Settings” to English, the key terms will be stored in English. Once this happens, the terms are difficult to change. Therefore, it’s important for the person who creates the database to set his or her “Regional Settings” to whatever language will best suit the needs of the organization. (Note that you can override your existing “Regional Settings” using the `/CultureInfo` option on the MainBoss command line. For more, see [Command Line Options on page 32.](#))

Note that changing Regional Settings may also change the currency symbol (e.g. from “\$” for dollars to “£” for pounds). However, MainBoss cannot handle multiple currencies; it assumes that all money values are in the same currency. For example, if you enter 33 in a “**Unit Cost**” field, MainBoss may display the result as \$33.00 or £33.00 depending on your Regional Settings...but internally, MainBoss simply record 33.00. Therefore, it is important that all MainBoss users use the same currency for all money values.

- **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 “**Database Server**”, specify the name of the SQL Server that you will use. You can get a list of all available servers by clicking the “...” button.

If you click the “...” button after “**Database Server**”, MainBoss will attempt to determine what servers are available to you. However, this list may be incomplete; certain firewall settings and SQL Server configurations can prevent servers from appearing in the list, even though the servers may be available for use.

3. In “**Database Name**”, enter the name of the new database. This will be the name used by SQL Server; 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. (The default database name is `MainBoss`; this name is recommended, unless your organization has multiple MainBoss databases.)

If you click the “...” button after “**Database Name**”, MainBoss will attempt to display all databases already managed by the specified “**Database Server**”. You cannot choose any of these names (because the databases already exist), but the list may help you see what names are not already in use. (Depending on your permissions, you may not be allowed to see the list.)

4. In “**Organization Name**”, enter the name of your company or organization. This will be used to create a **Location** record for your organization. Your “**Organization Name**” can contain spaces and special characters if you wish.
5. Click **OK**. MainBoss will proceed to create the database.

Entering License Keys

When you license MainBoss, you will be given a set of license keys for the software. Once you have installed MainBoss and created a new maintenance organization, MainBoss enters a special mode where you can enter these keys. 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

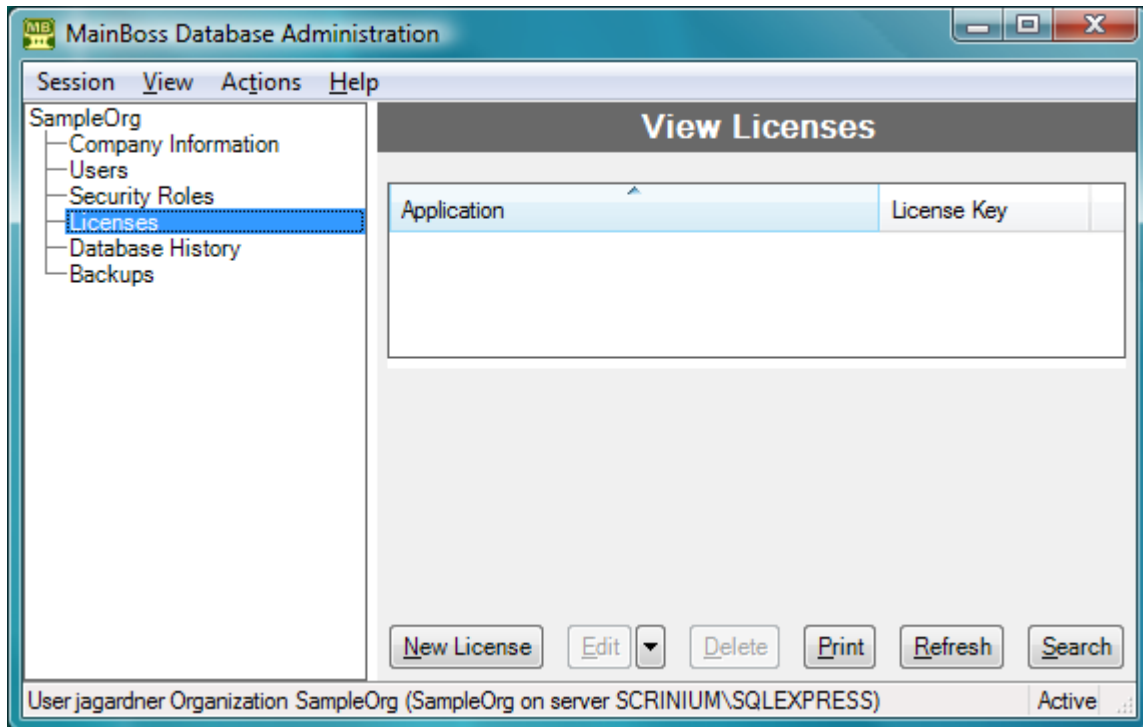
AAAAA-AAAAA-AAAAA-AAAAA-AAAAA

where each A is a letter or number.

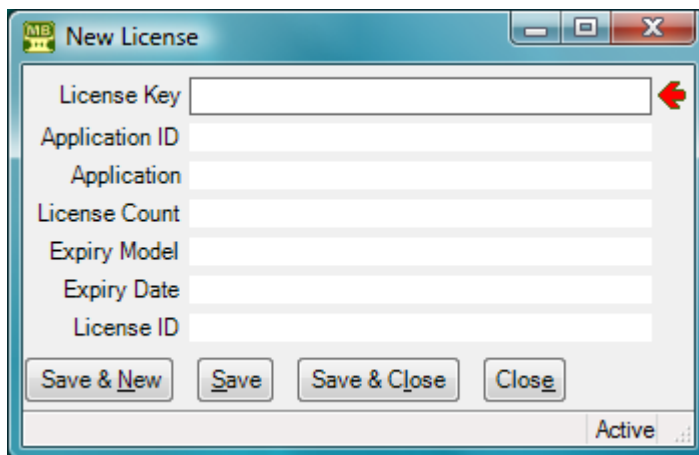
The presence or absence of license keys will change the appearance of many MainBoss windows. For example, if you have not licensed the **Inventory** module, you will not see functions related to inventory in the control panel and in other parts of MainBoss.

□ **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.

Security Roles: When you create a new user record, it has no associated security roles. This means that new users have no permission to do anything in MainBoss, unless you specifically assign them roles. You must therefore assign each user at least one security role if you expect the person to use MainBoss in any way. For more on security roles, see *Security Roles on page 16*.

Note: The person who creates a MainBoss database is automatically given a set of security roles which grant full access to all MainBoss operations.

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 14* (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 68* (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.

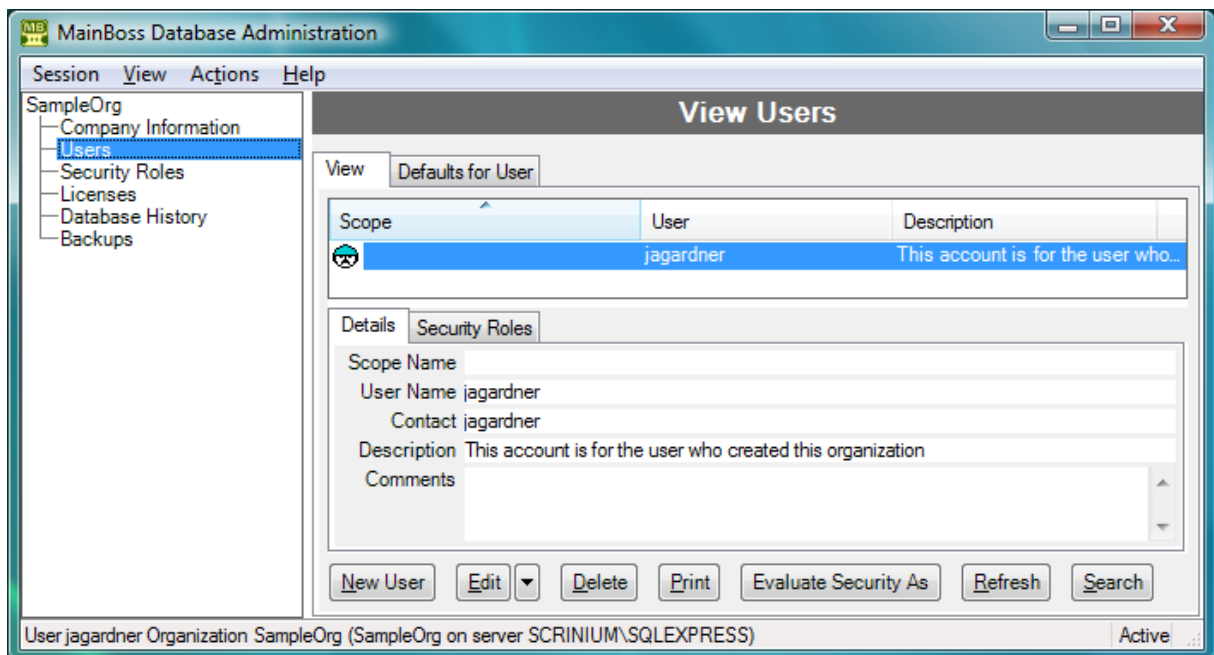
- If the option **MainBoss manages SQL Security** is checkmarked, anyone adding new users must have SQL Server Administration privileges
- If the option is blank, you must use the two-stage approach to adding new users (add the users in MainBoss first, then add them in SQL Server)

(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. On the "**Contact**" line, click the second drop-down arrow. From the resulting menu, click **New Contact**.
6. Fill in contact information for the user whose record you are creating. When you're done, click **Save & Close** to save the contact information and close the window.
7. Back in the user record, click **Save & New**.
8. Repeat Steps 3-7 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 entries there may be in the **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 68](#).

In order to add new users within MainBoss, your entry in the MainBoss **Users** table must have [User has Administrator permissions](#) checkmarked.

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 68](#).

Security Roles

In order to use MainBoss, users need to be assigned one or more security roles. The available roles are listed in [Administration | Security Roles](#). The comments provided in each role record indicate what permissions that role provides.

Security roles affect what a user sees: if a user doesn't have permission to see a particular type of information, that information will be blanked out or missing entirely from MainBoss windows. Security roles also affect what a user can do; for example, if a user has permission to view work orders (`WorkOrderView`) but not to close them, the [Close Work Order](#) button will be disabled for that user.

Important: When information is blank because you don't have appropriate permissions, move the cursor to the blank; MainBoss will tell you what security roles will give you permission to see that information. Similarly, if a button is disabled, move the cursor to the button; MainBoss will tell you what security roles will allow you to use that button. Note, however, that in some cases, information will disappear entirely (e.g. sections of records just won't be displayed); in this case, you may not even know that something is missing.

To assign security roles to a user, you use the [Security Roles](#) section of a user record (**Administration | Users**). In order to decide what roles an individual user should be granted, you must consider what that user is expected to do in his/her job. For example, consider someone working on a help-desk and receiving problem reports from clients outside the maintenance department. What does that person need to do?

- He or she will need to create work requests describing problems; that requires `Request` permission.
- Help-desk personnel also need a small amount of information on work orders (so that they can tell a client, "Your work has been scheduled for Thursday"). This suggests the `WorkOrderSummary` security role.
- The `UnitSummary` role might also be useful. It gives help-desk personnel enough information about units to make sure they understand which unit the client is talking about.

Other roles may also be appropriate, depending on what help-desk personnel are expected to do, and on your organization's policies.

The *Configuration Guide* offers suggestions on which security roles are appropriate for various duties within a maintenance organization.

Related Security Roles: Security roles tend to come in related groupings; for example, `WorkOrder`, `WorkOrderView`, `WorkOrderFulfillment` and `WorkOrderSummary` are all roles related to work orders.

In such a grouping, some roles automatically include others. For example:

- `WorkOrder` automatically includes all the other roles in the grouping.
- `WorkOrderView` automatically includes `WorkOrderSummary`.

For this reason, there's no point in giving someone both `WorkOrder` and `WorkOrderView` (for example), because `WorkOrder` automatically includes all the permissions of `WorkOrderView`.

In general, a `View` role automatically includes the related `Summary` role and a role with a simple name automatically includes roles of the same name with extra words (e.g. `PurchaseOrder` automatically includes `PurchaseOrderView`, `PurchaseOrderSummary`, and `PurchaseOrderReceive`).

Testing Security Roles

It can be difficult to figure out the effects of various security roles, especially when a user has several different roles combined. To test what a particular user can and can't do, you can use the [Evaluate Security As](#) button in **Administration | Users**. (You must have the Administration security role to use [Evaluate Security As](#).)

If you select a user in the **Users** table, then click [Evaluate Security As](#), your MainBoss session changes to seem as if *you* are the selected user. If certain types of data are invisible to that user, they'll become invisible to you too; if certain buttons are disabled, they'll be disabled for you too.

In this way, [Evaluate Security As](#) lets you see all the consequences of a user's security roles. Once you've examined what the user can and can't do, you can return to your own security roles by going to the **Session** menu and clicking **Reset to user YOURNAME security**.

Roles for Administering MainBoss

If you are administering MainBoss, you should have the Administration security role. You are automatically granted this role (and others) if you're the one who creates the MainBoss database.

The Administration role gives you permission to assign security roles to any user, including yourself.

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, you can set up MainBoss on other computers (if you wish).

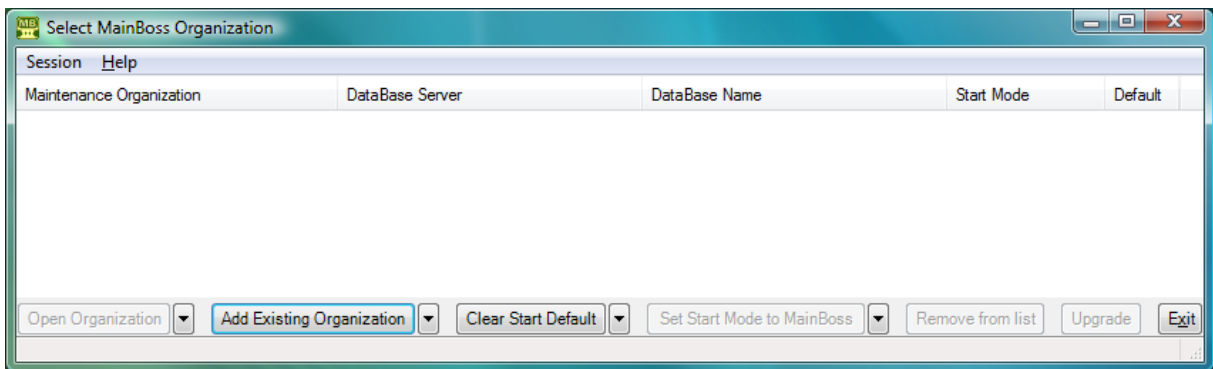
A MainBoss license dictates the number of people who may be authorized to use MainBoss. More specifically, it dictates the number of users who may be entered in the **Users** table (**Administration | Users**).

To install MainBoss manually on any computer, follow the steps given in *Manually Installing MainBoss Software on page 6*. 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. The alternative is to use ClickOnce deployment, as described in *ClickOnce Deployment Introduction on page 4*.

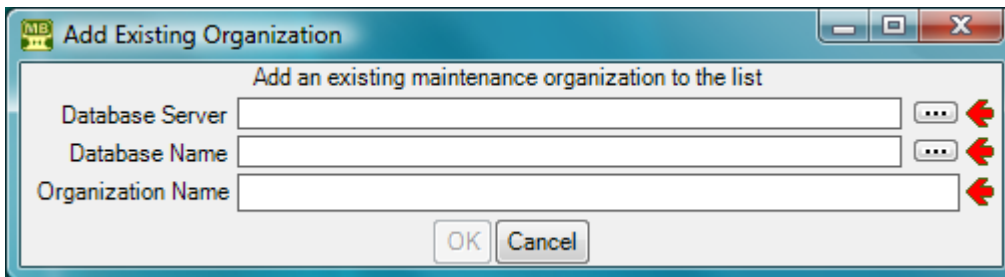
Important: MainBoss options are set up for each individual user. This means that each user must login and set up a connection to the MainBoss database. The process is easy, but the MainBoss Administrator must tell every other user the “**Database Server**” and “**Database Name**” needed for the connection. Alternatively, you can have each user log in and then the administrator takes over 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:



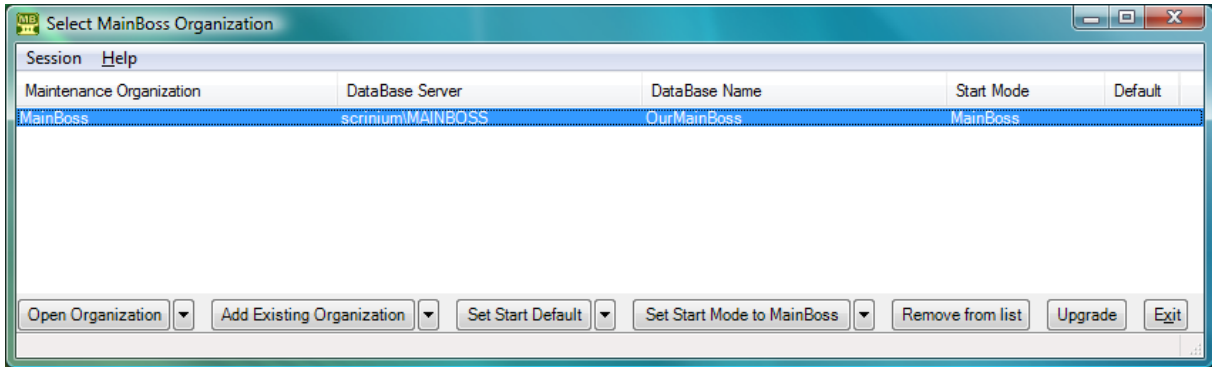
3. Click **Add Existing Organization**. This opens a window like this:



4. Set “**Database Server**” and “**Database Name**” to the same values specified when the database was created—see *Creating a Maintenance Organization on page 9*. The easiest way to set these fields is to click the “...” buttons associated with each field. These display lists of appropriate values. However, under some conditions (related to firewall settings), the “...” lists may not supply the values you want; in that case, you must type in the values by hand.
5. If you specify “**Database Name**” using the “...” button, MainBoss will fill in “**Organization Name**” with the company name specified in the database (from **Administration | Company Information**). If you wish, you can change “**Organization Name**” to your own private name for the organization; this will not affect any other users.

(You might change the “**Organization Name**” if you have a number of organizations you must keep track of, and you want to make sure their names are distinctive.)

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



7. Click **Open Organization**.

Once you have set up this computer to access the maintenance organization, you don't have to do it again.

Installing the @Requests Service

This section only applies if you have licensed the **@Requests Service** 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. You need Windows Administrator privileges to install and configure the @Requests service. You will find it easier to access the @Requests event log (i.e. diagnostic messages) if you are logged in to the computer where the @Requests service 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 26*.

If you turn off the computer where the @Requests service is running, incoming messages will stop being processed. When you turn the computer back on, @Requests will restart; at that point, it should begin processing the backlog of messages that built up while the computer was turned off.

Wherever you install the @Requests service, MainBoss should be **manually** installed on the same computer before you try activating the @Requests service. **If you start MainBoss with**

ClickOnce deployment, you will not be able to install, configure or manage 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 and Server 2008 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 special privileges; in particular, 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.

Note, however, that the login name for the @Requests service must have **Log on as a service** permission (as discussed below).

- 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.)

The entry for the login name in the **Users** table should have the following security roles: Requests, UnitView, Contact, WorkOrderView.

- 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. (To get the name of the computer, right-click

Computer or My Computer in the Windows Start menu, then click **Properties**. The resulting window includes the computer's name.)

The @Requests service will use the "Regional Settings" of the user name under which it runs. This means, for example, that diagnostic messages will be issued in the language associated with that user name (if MainBoss supports that language).

The login name used for running the service must have "service login" allowed by your system's security policies. To give a username this permission:

1. In the Windows Control Panel, click **Administrative Tools**.
2. If this computer controls domain security, you will see an entry for **Domain Security Policy**; click that. Otherwise, click **Local Security Policy**. Either way, Windows opens a window where you can set the policies.
3. In the left-hand section of the window, expand **Local Policies**, and click **User Rights Assignment**.
4. In the right-hand section of the window, double-click **Log on as a service**.
5. In the resulting window, click **Add User or Group**.
6. Enter the username you intend to use, then click **OK**.

Note: Domain security policies can be set up to prevent changes in local security policies. This means that you may not be permitted to change your local policies; if so, talk to the person who administers domain policies in order to resolve the situation.

Password changes: If you ever change the password associated with the service's login name, you must change the information associated with the service so that it uses the new password. To do so, the easiest approach is to use MainBoss to stop the @Requests service, uninstall it, and then install it again with the same login name and the new password. You can also change the password in the properties associated with the service in the Computer Management window.

Configuring @Requests

Once you have installed @Requests, you configure it by going to **Administration | @Requests** and clicking **Manage**. (On Vista and Server 2008, 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 IMAP4. We recommend that you use POP3 unless your site has a reason for preferring IMAP4. (For MainBoss's purposes, POP3 is more efficient than IMAP4.)

There are several versions of both POP3 and IMAP4. We do not recommend plain POP3 or IMAP4; these techniques transmit passwords in unencrypted form, making them insecure. POP3S and IMAP4S are secure enhancements of POP3 and IMAP4 that use encryption. TLS (Transport Level Security) offers a different approach to POP3 and IMAP4, also using encryption to obtain secure communications.

Which version should you use? It depends on your e-mail software. Different e-mail packages accept different techniques. If you don't know what your software accepts, we recommend trial and error in this order: TLS within POP3; POP3S; TLS within IMAP4; IMAP4S; POP3; IMAP4. As noted previously, POP3 and IMAP4 are not recommended—only use them if your e-mail software is too old to accept any of the secure techniques.

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

To configure @Requests, click [Edit](#) in the @Requests management window. 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 type: Choose one of the options listed.

Incoming mail server port: The port used by the mail server. When you select an option under "**Incoming mail server type**", MainBoss automatically sets the port to the standard default:

- For POP3 and TLS within POP3: 110
- For POP3S: 995
- For IMAP4 and TLS within IMAP4: 143
- For IMAP4S: 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**.

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

Outgoing SMTP mail server: 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.

Use SSL encryption for outgoing SMTP connection: If this box is checkmarked, @Requests will use SSL encryption when sending out mail (to requestors).

Outgoing SMTP server authentication: The options in this area determine what kind of authentication (if any) @Requests will use when sending out mail.

Anonymous: If this option is selected, @Requests will not use authentication (mail is sent without going through any identification process).

Use the default network credential: If this option is selected, @Requests uses whatever authentication is provided by the network as the default.

Using the specified SMTP domain, username and password: Choosing this option indicates that your SMTP server requires a domain, username, and password in order to send out mail. The fields below specify the required information:

Outgoing SMTP Domain: The domain that @Requests should specify when sending out mail.

Outgoing SMTP Username: The username that @Requests should specify when sending out mail.

Outgoing SMTP Password: The password for the given username.

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 be the

e-mail address of a person in the maintenance department who can personally handle responses from clients.

Important: The “**Return E-mail address**” should *not* be the same as the mailbox to which users mail requests. If it is, you’ll run into trouble if a requestor has an auto-reply set up, such as “Hi, I’ve gone on vacation and won’t be answering my e-mail.” What happens is this: if @Requests sends an acknowledgement to the user, the auto-reply replies to the @Requests mailbox; @Requests thinks this is a new request, so sends an acknowledgement to the requestor; the requestor’s auto-reply sends another reply to @Requests; @Requests sends another acknowledgement; and so on, until someone runs out of disk space. The way to avoid this is to set “**Return E-mail address**” to refer to a human being (who can be smart about dealing with auto-replies).

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 73*.

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 58.*)

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, stop the service by clicking **Stop Service**.
- Click **Uninstall**.

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 25.*

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 58](#).
- ❑ The **Remote Registry** service must be running on the computer where @Requests is installed. (On some systems, this service is turned on by default; in others, it's turned off.)

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

1. Login to an account with Windows Administration privileges on the computer where the @Requests service is installed.
2. Click on the Windows **Start** button.
3. Right-click **My Computer** (Windows XP and Server 2003) or **Computer** (Vista and Server 2008).
4. In the resulting menu, click **Manage**.
5. In the left-hand panel of the resulting window, expand **Services and Applications**.
6. Click **Services**.
7. In the right-hand panel, find the entry for **Remote Registry**.
8. 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.
9. Otherwise, right-click on the entry for **Remote Registry**. Click **Properties**.
10. In the resulting window, set **Startup type** to **Automatic**.
11. Click **OK**.
12. Right-click on **Remote Registry** again.
13. Click **Start**.

Note: Once you've done the above, you should check the firewall settings for **Remote Administration** to make sure that its use is restricted to your local network, not the whole Internet.

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.

Windows Administrator Privileges

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

- When you manually install MainBoss on a computer
- 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)

The first time you use ClickOnce deployment on a particular computer, you may need Administrator privileges if the computer doesn't have the prerequisite software (.NET 3.5 and Report Viewer)—the privileges are needed to install the prerequisites.

Transferring a Database to a New Computer

When you install the @Requests service on a computer, MainBoss stores various configuration information in the database. Amongst other things, this information tells MainBoss that the service is installed.

Now suppose you want to move your MainBoss database to a different computer, or you move your SQL Server to a different computer. Ideally, you should do this by following these steps:

1. Have all users on all computers exit the MainBoss program.
2. On the original server computer, login to an account with Windows Administrator privileges and start MainBoss.
3. **Uninstall the @Requests service**, as described in *Uninstalling @Requests on page 26*.
4. Backup the database using **Administration** | **Backups** (as described in *Backups within MainBoss on page 34*).
5. Exit MainBoss.
6. Copy the backup file to the new server computer.
7. Use **Create New Organization from Backup** to create a new MainBoss database from the backup file (as described in *Restoring a Backup Within MainBoss on page 37*).
8. Install the @Requests service on the new server computer (as described in *Installing the @Requests Service on page 20*).

This is the recommended approach for transferring a database to a new server or when you transfer SQL Server to a different computer. In some cases, however, you may be doing the transfer because the old server computer is unusable (e.g. due to hardware failure). In this case, you don't have a chance to uninstall the @Requests service first, and you'll be working from an old backup. The old backup database contains records stating that the service is still

installed, even though the service hasn't been installed on the new computer. To correct this situation, follow these steps:

1. Make a copy of the most recent backup of your database from the old server computer.
2. Use [Create New Organization from Backup](#) to create a new MainBoss database from the backup file (as described in *Restoring a Backup Within MainBoss* on page 37).
3. Using an account with Windows Administrator privileges, start up MainBoss and open the organization associated with the transferred database.
4. In **Administration** | **@Requests**, click the [Manage](#) button.
5. In the subsequent window, drop the arrow associated with the [Install](#) button and click [Clear Service Association](#).

[Clear Service Association](#) tells MainBoss to delete @Requests configuration information stored in the database. This will allow you to start from scratch, installing the @Requests service on the new computer and configuring it appropriate.

[Clear Service Association](#) also attempts to use `sc delete` to delete the @Requests service—the service whose name was just cleared from the database. (`sc delete` is a privileged command that deletes a service; for more information, see your Windows documentation.) This `sc delete` operation should be unnecessary—you should only be using [Clear Service Association](#) when the computer that was running the service is no longer in operation. However, deleting the service is intended to avoid losing incoming requests if you click [Clear Service Association](#) accidentally.

Important: When transferring the @Requests service from one computer to another, uninstall the service on the old computer first, or else make sure that the old computer is permanently taken out of service. If the old computer is running the @Requests service at the same time as the new computer, the old computer will “steal” e-mailed requests from the designated mailbox but will do nothing to process those requests. In other words, you’ll lose requests.

[Clear Service Association](#) is *only* intended for the circumstances described above. Do not use it in other circumstances.

If the computer that was running the old @Requests service ever comes back into operation, you may need to delete the old service using `sc delete`. If you do not, the old service will start up every time Windows does. However, it will then check the MainBoss database and see that the database is no longer configured to use the old service. The old service will then issue an error message and terminate itself. Therefore, the old service won't interfere with MainBoss operations, but you should still get rid of it with `sc delete`.

The @Requests Service and MainBoss Upgrades

Before installing a new update of MainBoss, we strongly recommend that you use the old version of the software to stop and uninstall the @Requests service before you install the new software. If you fail to do so, the old service won't work with the new software—the old service will see that it's out-of-date and terminate itself with an error message. In this case, you'll have to use the new software to stop and uninstall the old service, then install a new service that will work with the new software.

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.

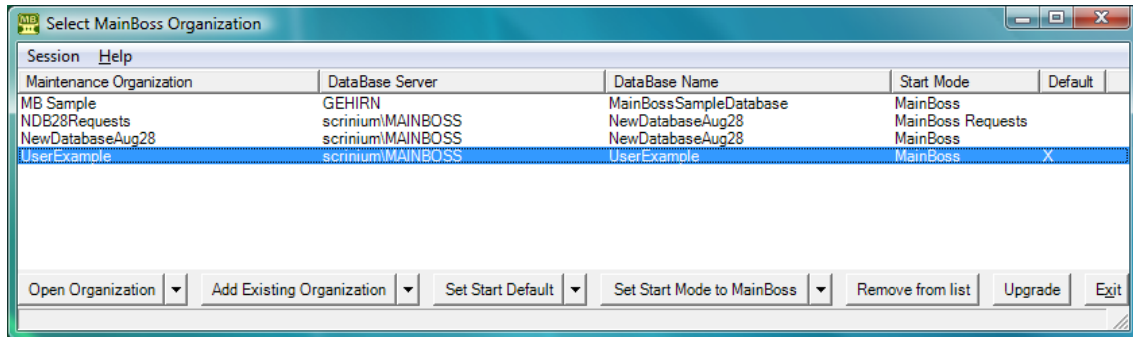
If you have multiple databases, each must have its own separate set of license keys. You may not use the same license keys for multiple databases.

1. You create each new database following the same procedures described in *Creating a Maintenance Organization on page 9*.
2. You then record the database's license keys as described in *Entering License Keys on page 10*.
3. You authorize appropriate users (as described in *Users on page 12*).
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 18* 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 20*.

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. Your license limits the maximum number of names that may be entered in the **Users** table.

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.

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.

Administration functions can only be used if you have the Administrator security role.

View Sessions: Only shows who else is using MainBoss.

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.

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 “**Organization Name**” MyMainBoss-Full, with a start mode of **MainBoss**.
- A second line might have the “**Organization Name**” 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.

You cannot specify command line options with ClickOnce deployment. If you think you’ll need to use command line options on a particular computer, use manual installation on that computer.

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.

Culture Information: The `/CultureInfo` option lets you specify Windows “culture” setting to be used when you execute MainBoss. A *culture* is a collection of information about language and other locale-based information, e.g. what monetary symbols are used in a particular place.

By default, MainBoss takes its culture information from your Windows Regional Settings; it then attempts to operate in the language given by those settings (e.g. MainBoss attempts to run in French if that’s what your settings dictate). However, if you specify a `/CultureInfo` argument on the command line, the argument overrides the Regional Settings. This is particularly useful if you’re creating a batch command file or shell script that you want to have work in a specific language, no matter what the user’s Regional Settings might be.

The argument of `/CultureInfo` is the official Windows “CultureInfo” string name for the desired culture. For example, if you want generic Spanish, you’d specify `/CultureInfo:es-ES`; for U.S. English, you’d use `/CultureInfo:en-US`. See your Windows documentation for recognized “CultureInfo” string identifiers.

Options on the MainBoss command line are not affected by `/CultureInfo`. For example, the option `/Mode:Requests` does *not* change if you switch to a language where “requests” is translated to a different word.

Note: If you change the culture information with `/CultureInfo`, MainBoss makes the change *after* the command line arguments have been parsed. This means that if your command line has syntax errors, you may see error messages in the user’s original preferred language, not in the language specified by `/CultureInfo`.

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. Right-click an open area of the Windows desktop. In the resulting menu, click **New**, then **Shortcut**.
2. Windows asks you to specify the file to execute. Click the **Browse** button and use it to locate the MainBoss executable file (named `mainboss.exe` in the folder where you installed the MainBoss software—typically `\Program Files\Thinkage\MainBoss\3.2`).
3. Click **Next**.

4. Enter a descriptive name for the shortcut, then click **Finish**. This creates a new desktop icon with the name you specified.
5. Right-click on the new desktop icon. In the resulting menu, click **Properties**. Windows will open a Properties window for the icon.
6. 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.
7. 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 makes 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 databases. 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 have a number of options:

- You can do backups manually through SQL Server Management Studio. For more information, see *Backups with SQL Server on page 51*.
- You can use the MBUtility command as described in *Backing Up Your Database with MBUtility on page 46*.
- You can back up your database inside MainBoss by using **Administration | Backups**. For details, see the next section.

Ideally, however, you should use automated backup facilities as discussed above, either through commercial software or through SQL Server’s built-in facilities. Because these facilities operate automatically, you won’t have to remember to do backups manually.

Backups within MainBoss

You can perform backups within MainBoss using **Administration | Backups**. The general procedure is:

1. Create one or more *backup names*. A backup name is the name of a file into which MainBoss should write backup data. For example, you might create seven backup names: one for each day of the week.
2. When you want to perform a backup, select a name from the list of available backup names. Then click **Backup**. This submits a request to the SQL Server asking the server to create a backup. (In order to use this function, you must have SQL Server Administrator privileges.) When the backup occurs, it will overwrite any current contents of the specified backup file.
3. The next time you want to do a backup, select a different name from the list and click **Backup** again. MainBoss sorts the backup names so that the oldest backup file is at the top of the list. Therefore, if you have seven backup names, and you always select the top name whenever you make a backup, your backup will always overwrite the oldest file. This makes it easy to reuse the same set of files by cycling through them.
4. When you initiate a backup operation, MainBoss makes an appropriate note in the database history (**Administration | Database History**). If the backup fails for some reason, MainBoss makes note of the failure in the database history; however, if the backup succeeds, MainBoss doesn't make a history note. In other words, if there's no note about the backup failing, you can assume that it succeeded.

A number of important considerations apply to the backup name:

- Backup operations are run by SQL Server itself on the computer where SQL Server is running. Because SQL Server does the backup, backup names must refer to files that are accessible to the login name under which SQL Server runs. Furthermore, backup names are interpreted relative to the computer where SQL Server is running.

For example, suppose SQL Server is running on Computer X, but you're working on Computer Y. When you specify the backup name,

```
C:\MyBackups\Oct20_2009.bak
```

you may think that this refers to a file on Computer Y's C: drive. However, the backup is done by SQL Server on Computer X; therefore, SQL Server will try to save to Computer X's C: drive, not Computer Y's. If the directory C:\MyBackups doesn't exist on Computer X (or if SQL Server's login name doesn't have permissions to write to that folder), you'll get an error message and the backup won't work.

- If you specify a backup name that does not begin with "\\\" and does not begin with a drive letter (e.g. "C:\\"), SQL Server automatically assumes the name refers to a file under SQL Server's working folder. The location of this working folder depends on information specified when SQL Server was installed.
- SQL Server backup files typically have names ending in the extension ".bak", as in MyBackup.bak. Therefore, if you specify a simple name for a backup name—a name that doesn't have any slashes, backslashes or dots—MainBoss automatically adds ".bak" on

the end of the name. Therefore, if you specify a simple name like “abc”, MainBoss automatically turns this into “abc.bak”.

- For safety’s sake, backup names should refer to a disk drive that is *different* from the one that contains the actual MainBoss database. Otherwise, if the disk drive malfunctions, you could lose both your original database and your backups. It’s even better if the backup name refers to a completely different computer, as in

\\NotTheSameComputer\MyBackups\Backup1.bak.

It’s also a good idea to make copies of backup files and to keep those copies somewhere off your premises. That way, if you have a fire or some other problem that affects your entire site, you’ll have backups someplace else safe.

(By default, SQL Server puts backup files on the same disk as the corresponding database. Therefore, you should make sure *not* to use the default location.)

When you specify a backup name, the “**Defaults**” line in **Administration | Backups** specifies the computer, the directory, and the file name extension. **We strongly recommend that you examine the “Defaults” line whenever you create a backup name and before you click Backup.** This will avoid failed backups or accidentally creating a backup file somewhere you don’t expect.

The information in “**Defaults**” is generated by MainBoss at the time the window is displayed. It’s dependent on your SQL Server configuration. Therefore, suppose you specify a backup name of “abc”. “**Defaults**” shows where the backup file would be written if you clicked **Backup** right now. However, if your SQL Server configuration changes, future backups with the “abc” backup name might be written to a different folder.

The window for viewing backup names includes the following:

Name list: As noted above, the default is to put the oldest backup file at the top of the list.

Information area: The area below the list shows information about the selected backup name/file.

Last backup: The last date/time that backup information was stored in the selected file.

Name: The name you specified for the file.

Defaults: Information telling where the backup information will be written if you click **Backup** right now.

Comments: Any comments you’ve associated with this backup name.

New Backup Name: Opens a window where you can specify a new backup name.

Backup: Begins a backup on your database. See the notes at the start of this section for further information on the backup process.

Note: If you perform backups and restores entirely through SQL Server, MainBoss won't store information about such operations. You won't see corresponding entries in the **Backups** list or in the database history.

When MainBoss initiates a backup in this way, it makes an appropriate entry in the database history (**Administration** | **Database History**). If the backup fails, MainBoss makes a second entry in the database history reporting the failure. If the backup succeeds, MainBoss doesn't record this—the default is to assume that the backup was successful.

Restoring a Backup Within MainBoss

SQL Server offers various methods for restoring a database from a backup file. For further information, see the SQL Server documentation.

However, you can also restore a database from within MainBoss itself, using facilities in the Maintenance Organization window. (This is the window that lets you choose a database to open. One way to open this window is to go to MainBoss's **Session** menu and click **Change Maintenance Organization**.)

MainBoss offers two operations for restoring databases from backup:

- **Restore Organization** overwrites an existing database. First you select the database to be overwritten (by clicking on the entry for that database in your Maintenance Organization list), then you click on **Restore Organization** to specify the backup file you want to use.

Typically, you would use **Restore Organization** if you made some sort of large-scale error that you wanted to undo (e.g. mistakenly generating and committing a large number of planned maintenance work orders). **Restore Organization** is not intended for situations where your database is no longer usable because it's been deleted or corrupted.

Restore Organization is particularly useful when you're just getting started with MainBoss. For example, you can backup your database and then experiment with various ways of setting up your work (especially planned maintenance tasks). If you decide to take a different approach, you can restore from your backup and try again.

- **Create New Organization from a Backup** creates a new database using data from a backup file.

Typically, you would use **Create New Organization from a Backup** if your original database has been deleted or corrupted. You could also use this operation if you wanted to create a test database using data obtained from your production database.

To execute either **Restore Organization** or **Create New Organization from a Backup**, click the drop-down button associated with **Add Existing Organization** and choose the operation from the resulting menu.

In both cases, MainBoss performs the operation by submitting requests to SQL Server. You must therefore have SQL Server Administrator privileges to perform either operation.

Restore operations are run by SQL Server itself on the computer where SQL Server is running. Because SQL Server does the work, the backup file that will be used must be accessible to the login name under which SQL Server runs. Furthermore, the name of the backup file should be relative to the computer where SQL Server is running. Unless the file name explicitly begins with `\\computer` or a drive name (e.g. `C:\`), the file name will be assumed to be relative to SQL Server's home directory. (This is true even if the file name begins with a `\`.)

For example, suppose SQL Server is running on Computer X and you submit a restore request while working on Computer Y. MainBoss will ask you to specify the name of the backup file containing the database you want to restore. Suppose you give the name

```
C:\MyBackups\Oct20_2009.bak
```

You may think this refers to a file on Computer Y's `C:` drive. However, the restore operation is performed by SQL Server on Computer X; therefore, SQL Server will try to find the file on Computer X's `C:` drive, not Computer Y. If the directory `C:\MyBackups` doesn't exist on Computer X (or if SQL Server's login name doesn't have permissions to read that folder), you'll get an error message and the restore operation won't work.

For this reason, we recommend that you only submit restore requests while logged in to the computer where SQL Server is running. You should also be aware of what permissions SQL Server has and what files will actually be accessible to SQL Server's login name.

Both [Restore Organization](#) and [Create New Organization from a Backup](#) ask you to specify the following:

Database Server: The network location of the SQL Server through which you will access the database. If you click the accompanying "...” button, MainBoss displays a list of database servers that are known to your Windows system. There may be additional servers that you can use but that Windows doesn't currently know about.

Database Name: The name of the database you want to overwrite or create. With [Restore Organization](#), this must be the name of a valid MainBoss database, and the "...” button will show all known MainBoss databases on the specified “**Database Server**”. With [Create New Organization from a Backup](#), you can specify any name (existing or not), and the "...” button shows all databases, not just MainBoss ones.

Organization Name: A name that you personally will use for this database. This should not be the same as the organization name for any other database.

[Restore Organization](#) also asks you to specify:

Backup File Name: Fill in the field with one of the names in your list of backup names (as given in [Administration | Backups](#)). If you choose a name from this list, MainBoss tries to restore from the associated backup file.

MainBoss's list of backup files only contains files made inside MainBoss using **Administration | Backups**. The list doesn't backups made directly with SQL Server Management Studio.

Create New Organization from a Backup asks you to specify:

Restore Data: Fill in the field with the name of a SQL Server backup file containing a backup of a MainBoss database.

This field has an associated “...” button. If you are logged in to the same computer where SQL Server is running, clicking “...” opens a standard “file open” dialog to let you find the backup file you want.

If you are logged in to any other computer, the “...” button does nothing and you must type in the backup file's name by hand. The name should start with \\computer or a drive name (e.g. c:\), or else it should be relative to the SQL Server's default folder on the computer where SQL Server is running. (See above for more details on specifying the file name.)

Once you have specified the required data, click **OK**. This submits an appropriate request to SQL Server, which performs the restoration.

The MBUtility Command

The MBUtility command offers a command-line interface to selected MainBoss operations. This allows administrators and IT personnel to create shell scripts (e.g. .bat or .cmd command files) which perform common administrative functions.

MBUtility is installed as an .exe file in the same folder as other MainBoss software. To execute MBUtility, you can cd to this folder or you can create a shortcut to the program. You can also add the MainBoss software folder to your PATH environment variable.

MBUtility is only available if you have installed the MainBoss package manually. The software will not be available on your computer if you use ClickOnce deployment.

In general, the format of the command is

```
mbutility verb option option option ...
```

The verb must always come immediately after `mbutility`, but the options may appear in any order. Options may have any of the following forms:

/keyword:value

Specifies a value. For example, /Output:filename specifies an output file.

+Option

Turns on an option. For example, +Probe may be used with the AddOrganization verb; it tells MBUtility to check that the corresponding database is actually accessible from the current computer.

-Option

Turns off an option. For example, -Probe may be used with the AddOrganization verb; it tells MBUtility not to check the database's accessibility, but just to add it to the list of known organizations.

Abbreviations on the MBUtility Command Line: Most keywords on the MBUtility command line may be abbreviated. In the sections that follow, letters that *must* be included are shown in upper case; any or all other letters may be omitted. For example, the following are all valid abbreviations for the option /DataBaseName

```
/databasename  
/databasen  
/dbname  
/dbn
```

Although the documentation shows some letters in upper case and some in lower, you can type letters in any case you want. For example, /DBN is exactly the same as /dbn or /Dbn on the MainBoss command line.

Getting Help for MBUtility

The command

```
mutility Help
```

displays a summary of MBUtility's syntax and options. As noted in the previous section, Help may be abbreviated by omitting any or all lower case letters, as in

```
mutility h
```

Listing MainBoss Organizations

The ListOrganizations verb of MBUtility lists MainBoss maintenance organizations known to be available on the current computer. The command has the format

```
mutility ListOrganizations option ...
```

where the possible options are:

+Probe

Determines whether you can actually connect with each organization listed. The default behavior is -Probe, which means that MBUtility doesn't test connections in this way.

+RealNames

Displays the real registry name for each database, as well as the databases' usual display names. The default behavior is `-RealNames`, which means MBUtility only shows the usual display names.

Built-In Data Schemas

A *data schema* consists of XML code describing the format of data in a database. Schemas are used when importing data into a database or exporting data from the database. MBUtility has a number of built-in schemas that can be used for this purpose. To get a list of these built-in schemas, execute the command

```
mbutility ListImportSchemas
```

The result is a list of keywords that may be used on an MBUtility command line, as described in later sections of this guide.

Generating Schema Definitions: You can obtain the actual XML code for a built-in schema using MBUtility's `GenerateImportSchema` verb. The command line has the format:

```
mbutility GenerateImportSchema option option ...
```

where the options are:

`/SchemaIdentification:schema`

Specifies the schema whose definition you wish to obtain. This should be one of MBUtility's built-in schemas. Names are case-sensitive. If the schema name contains blank characters, the name should be enclosed in quotes, as in

```
/si:"Storeroom Assignment"
```

`/Output:filename`

A file where MBUtility should write the XML definition of the schema. This file will be written in Unicode characters, which means it should only be edited with software that can deal with Unicode (e.g. Wordpad).

+EXCEL

Specifies that the XML code in the schema should be in a format acceptable to Microsoft Excel.

For example, the following creates a schema for unit records, in a format acceptable to Excel:

```
mbutility gis /si:Unit /out:unitdef.xml +excel
```

Exporting Data from Your Database

The `Export` verb of MBUtility exports selected data from your database to an output file. For example, you might regularly wish to export data to another application for some reason; by

making a command file that uses MBUtility, you could automate the export process, thereby simplifying the task (and also avoiding errors).

The command line for exporting is

```
mutility Export option option ...
```

where the options are:

`/SchemaIdentification:schema`

Specifies a data schema for exporting the data. This should be one of MBUtility's built-in schemas (as discussed in *Built-In Data Schemas on page 41*). If the schema name contains blank characters, the name should be enclosed in quotes, as in

```
/si:"Storerroom Assignment"
```

`/+EmbeddedSchema`

Tells MBUtility to embed the specified data schema in the output file. This is useful if you're exporting the data to software that accepts embedded schemas. However, Microsoft Excel doesn't seem to accept such schemas, so don't use `+es` for going to Excel.

`/-EmbeddedSchema`

Tells MBUtility not to embed a data schema in the output file. This is the default.

`/Output:filename`

Specifies a file where the output should be written. This file will be written in Unicode characters, which means it should only be edited with software that can deal with Unicode (e.g. Wordpad).

`/OrganizationName:name`

The name of the maintenance organization (associated with your MainBoss database). This name is case-sensitive; for example, `MainBoss` is not the same as `mainboss`.

`/DataBaseServer:servername`

The name of the server that holds the MainBoss database.

`/DataBaseName:name`

The name of the MainBoss database.

If you specify `/OrganizationName`, you don't have to specify either `/DataBaseServer` or `/DataBaseName`. If you don't specify `/OrganizationName`, you must specify both `/DataBaseServer` and `/DataBaseName`.

For example,

```
mutility ex /si:Unit /out:myunits /on:MyMainBoss +es
```

exports data about units to a file named `myunits`. The output file will include an embedded data schema for unit records.

An Example of Exporting to Excel 2007

The previous sections of this manual gave basic details of exporting MainBoss data. This section provides a concrete example, showing how to export unit data to Microsoft Excel 2007.

The first step is to create a file containing the data schema for unit data. You do this with the `GenerateImportSchema` verb:

```
mutility gis /si:unit /out:unitschema.xml +excel
```

This writes the schema into a file called `unitschema.xml`.

The next step is to export the data:

```
mutility ex /si:unit /out:data.xml /on:mymb
```

This exports data from a database with the organization name `mymb`, and writes the data into a file named `data.xml`.

The next step is to modify the data file's XML code so that it refers to the schema file. Start the Windows `wordpad` command and read in the data file. The second line will look like this:

```
<dsUnit xmlns="http://thinkage.ca/MainBoss/dsUnit.xsd">
```

Using Wordpad, edit the line so that it looks like this:

```
<dsUnit xmlns="http://thinkage.ca/MainBoss/dsUnit.xsd"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://thinkage.ca/MainBoss/dsUnit.xsd unitschema.xml">
```

Note that this line ends with the name of the file containing the schema (followed by the characters `">`). Once you've made these changes, save the file.

The data file is now ready to be used by Excel. If you have Microsoft Office 2007, here's what you do:

1. Start Excel.
2. Click the "Office" button (in the upper left hand corner of the Excel window).
3. In the resulting window, click "Excel Options" (at the bottom of the window).
4. In the resulting window, click "Popular" in the left part of the window, then checkmark "Show Developer Tab in the Ribbon" (if it isn't already checkmarked).
5. Click OK to close the options window.
6. In the main Excel window, click "Developer" in the ribbon.
7. In the XML section of the ribbon, click "Import".
8. In the resulting window, specify the name of the data file (in our example `data.xml`).

Excel gives you the option of reading the data file into the current spreadsheet or into a new one.

Importing Data into Your Database

The `Import` verb of MBUtility imports data into your database from an input file. For example, if you are migrating to MainBoss from some other maintenance management software package, you might be able to import some subset of the data from that other package.

Important: Importing is a complicated procedure, and should only be done by people with strong IT expertise. We strongly recommend that you backup your database before you attempt an import operation. If an error occurs during an import operation, restore the database from backup, correct the error in the import data, then try again.

The command line for importing is

```
mbutility import option option ...
```

where the options are:

`/SchemaIdentification:schema`

Specifies a data schema for exporting the data. This should be one of MBUtility's built-in schemas (as discussed in *Built-In Data Schemas on page 41*). Names are case-sensitive. If the schema name contains blank characters, the name should be enclosed in quotes, as in

```
/si:"Storeroom Assignment"
```

`/Input:filename`

Specifies a file containing the data to input. This file should use Unicode characters.

`/ErrorOutput:file`

Specifies a file where MainBoss should write any error messages that arise during the import process.

`/OrganizationName:name`

The name of the maintenance organization (associated with your MainBoss database). This name is case-sensitive; for example, MainBoss is not the same as mainboss.

`/DataBaseServer:servername`

The name of the server that holds the MainBoss database.

`/DataBaseName:name`

The name of the MainBoss database.

If you specify `/OrganizationName`, you don't have to specify either `/DataBaseServer` or `/DataBaseName`. If you don't specify `/OrganizationName`, you must specify both of `/DataBaseServer` and `/DataBaseName`.

For example,

```
mbutility im /si:Unit /in:myfile /eo:errs /on:MainBoss
```

imports unit data from a file named `myfile`.

Important: This version of MainBoss will not let you import data that conflicts with existing records in the database. For example, if you are importing unit data, each new unit's identifier code must be different from any units that are already in the database. This means that you can't export MainBoss data, edit it with some other piece of software, then import it again to overwrite MainBoss's existing data.

If you wish to create a database for testing, we recommend using backup and restore (using [Create New Organization from Backup](#)) rather than exporting and importing data. For more information, see [Backups within MainBoss on page 34](#) and [Restoring a Backup Within MainBoss on page 37](#).

Tip: Sometimes it's difficult to figure out what format to use when you are preparing data for import. When in doubt, start up MainBoss and create a record of the type you intend to import; for example, if you intend to import Unit records, create a Unit record inside MainBoss. Then export the record as previously described in the exporting section. The format of the exported record will be the same that's expected for imported records.

Adding Organizations

The `AddOrganization` verb of MBUtility lets you add an organization to the current user's organization list. The database for this organization must already exist and the user must have appropriate permissions on the database (which means that the user's name must be specified in the database's **Users** table).

For example, if you are logged in under the name `jsmith`, you can use `AddOrganization` to put a new organization in `jsmith`'s list of known organizations, provided that `jsmith` appears in the database's **Users** table.

The format of the command is

```
mbutility AddOrganization option option ...
```

where the options are:

`+Replace`

This option is relevant if the user's organization list already contains an organization with the same name. `+Replace` says the old organization should be replaced with the new organization. If you do not specify `+Replace`, MBUtility will not replace old organization's with new ones; if there is a name conflict, you will be receive an error message.

+Probe

If you specify this option, MBUtility will attempt to verify that the user can actually access the database. If you do not specify +Probe, the database will be added to the user's organization list, but MBUtility doesn't check to see if the user can actually connect to the database.

/Mode:startmode

Specifies the default start-up mode when the user connects to the database. The startmode value must be one of the following strings:

```
MainBoss Requests
MainBoss
View Sessions
Administration
```

For explanations of these startup modes, see *Start Modes* on page 31 of this guide.

/OrganizationName:name

The name that should be used in the user's organization list. This name is case-sensitive.

/DataBaseServer:servername

The name of the server where the database resides.

/DataBaseName:name

The actual name of the database that you want to add.

You must specify all three of /OrganizationName, /DataBaseServer and /DataBaseName.

Backing Up Your Database with MBUtility

The Backup verb of MBUtility lets you back up a MainBoss maintenance organization database. The format of the command is

```
mutility Backup option option ...
```

where the options are:

/BackupFile:filename

The name to be used for the backup file. This file should not already exist. Traditionally, SQL Server backup files end in ".bak", so

```
/bu:"c:\SomeFolder\SomeFile.bak"
```

tells MBUtility to save your database's data in a file with the given name.

/OrganizationName:name

The name of the maintenance organization (associated with your MainBoss database). This name is case-sensitive; for example, MainBoss is not the same as mainboss.

`/DataBaseServer:servername`

The name of the server that holds the MainBoss database.

`/DataBaseName:name`

The name of the MainBoss database.

The MBUtility `Backup` verb is subject to all the conditions that affect the behavior of **Administration | Backups** within MainBoss itself. For more, see *Backups within MainBoss on page 34* of this guide.

Backups done with MBUtility are *not* added to the list in **Administration | Backups**.

Adding an Administrator to a Database

In some rare cases, a MainBoss database may not contain any users who have administration privileges. This may happen, for example, if only one user has administration privileges and that user accidentally deletes his/her name from the **Users** table. It may also happen if an administrator leaves your organization without authorizing anyone to take over administration duties.

In this case, the `BreakIn` verb of MBUtility gives you a way to plant your own login name in the **Users** table of an existing database, and to give yourself MainBoss Administration privileges. To use this function, you need SQL Server Administrator privileges.

For information on user records, see the reference manual or the online help.

The format of the `BreakIn` command is:

```
mbutility BreakIn option option ...
```

where the options are:

`/OrganizationName:name`

The name of the maintenance organization (associated with your MainBoss database).

This name is case-sensitive; for example, `MainBoss` is not the same as `mainboss`.

`/DataBaseServer:servername`

The name of the server that holds the MainBoss database.

`/DataBaseName:name`

The name of the MainBoss database.

If you specify `/OrganizationName`, you don't have to specify either `/DataBaseServer` or `/DataBaseName`. If you don't specify `/OrganizationName`, you must specify both of `/DataBaseServer` and `/DataBaseName`.

`/ContactCode:name`

Specifies a **Contacts** table entry to be associated with the created **User** record. For example,

```
/cc:"Joe Smith"
```

indicates that the **User** record should be associated with the **Contacts** record with the name Joe Smith. This argument is mandatory for MainBoss Advanced 3.2 and up. With earlier versions of MainBoss, the argument should be omitted.

When the user record is created, it will be in your name (i.e. the login name of whoever executes the MBUtility `BreakIn`). This user record may or may not have a scope name; in particular, MBUtility may set the scope name to the name of the computer where you executed MBUtility. This means that you will only be able to login from that computer. This is no problem if you are only breaking into the database so that you can set up someone else as a new database administrator. However, if you intend to keep this user record in the long-term, you may wish to edit the record to set the scope name to something more useful.

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) or SQL Server 2008 SP1. 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 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.

Multiple versions of SQL Server: It is technically possible to run different versions of SQL Server simultaneously on the same computer (e.g. SQL Server 2003 running alongside SQL Server 2008). However, we do not recommend this arrangement. The reason is that only one version will be able to use TCP/IP connections; that version will “steal” TCP/IP from the other version, which will lead to various difficulties. Therefore, we recommend that you only have one version of the software running on the computer that manages the MainBoss database. (You can, however, have multiple instances of the same server software running on that computer, e.g. multiple instances of SQL Server 2005. For more on instances, see [Instances on page 50.](#))

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 52*.

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 or Server 2008: If you will be installing SQL Server on Vista or Server 2008, 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.

Default Backup Folder: SQL Server 2008 lets you specify the default folder to be used to hold backup files. (SQL Server 2005 does not provide this ability.) We recommend that you set the default folder to be some device *other* than the disk drive that holds the actual MainBoss database. If your backups go to the same disk that holds your original database, a hardware error on that disk may kill both your database and the backups.

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.

We do not recommend using the default save folder for the backup file, unless you set the default folder as discussed in the preceding section of this guide. SQL Server's usual default backup folder makes backup files difficult to find and work with, especially on Vista. Instead, we recommend that you save backup files in some folder that you'll remember and find easily. Also, backup files should be copied to some medium that can be removed from the premises; it's important to keep a recent backup off-site in case of fire.

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.

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.

Note: From time to time, you should check the SQL Server logs to make sure that backups really are taking place successfully. Thinkage has encountered situations where the backup software reports no problems, but where the SQL Server logs show that the database backup failed.

If you do backups directly through SQL Server Management Studio, information about the backup will *not* appear in MainBoss’s internal database history (**Administration | Database History**). Furthermore, the backup will not appear in MainBoss’s list of known backups (**Administration | Backups**).

Using SQL Server to Restore a Database from Backup

To restore a database from a backup file, 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 **Databases**.
3. Right-click on the entry for the database you want to restore, then click **Tasks**, then click **Restore**, then click **Database**.
4. MainBoss opens a window displaying your existing backup files. In the list of available backups, click the one you want to restore.
5. Click **OK**.

The restoration process will overwrite any existing contents of the database. Therefore, you should only restore a database if the current contents are no longer usable.

SQL Express Considerations

SQL Express is a free version of SQL Server. It is freely available from Microsoft. Note, however, that you must install Microsoft’s (free) Silverlight software before you install SQL Express. To obtain Silverlight, go to

<http://www.microsoft.com/silverlight/>

To obtain SQL Express, go to

<http://www.microsoft.com/express/sql/download/>

If you obtain SQL Express from this page, make sure you download the version “with Tools” or “with Advanced Software”—you’ll need Management Studio to do such things as backups and restores. When you install the downloaded software, follow the instructions given below in *Installing SQL Express on page 53*.

SQL Express will provide acceptable performance for small sites. Larger sites should buy one of the larger versions of SQL Server.

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. 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 or Server 2008).

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. 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 62*.

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 65*.

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 67*.

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>

Otherwise, just use the backup facilities within MainBoss.

Important: By default, SQL Server configures itself so that backup files are written to the same folder as the databases themselves. **This offers no protection against hardware errors**—if you get a disk crash on the disk that holds the databases, you'll lose the backup files too. Therefore, you should configure SQL Server so that backup files are written to a completely different device, preferably on a completely different computer.

Ideally, you should make copies of your backup files and keep those copies off your premises. That way you don't lose data if there's a fire or some other disaster that affects your entire site.

Upgrading from SQL Server 2005 to SQL Server 2008

Over time, many MainBoss users may choose to upgrade from SQL Server 2005 to SQL Server 2008. This isn't necessary for using MainBoss, but we have observed that SQL Server 2008 sometimes gives better performance.

The upgrade process from SQL Server 2005 to SQL Server 2008 is entirely controlled by Microsoft. Therefore, Thinkage Ltd. takes no responsibility for any problems that users might encounter during the upgrade. However, to give our customers some guidance, we offer the following notes:

- If you upgrade from a full version of SQL Server 2005 to a comparable version of SQL Server 2008, simply install the software according to Microsoft’s specifications. The installation process will handle everything; this includes migrating your databases from the old software to the new.
- If you are currently running the free SQL Server 2005 Express, you can migrate to a comparable free product called SQL Server 2008 Express with Tools. Go to

<http://www.microsoft.com/express/sql/download/>

Make sure you download the version “with Tools” or the one “with Advanced Services”—you want to be certain you have a version of Management Studio. Once you have downloaded the software, install it according to Microsoft’s specifications. The installation process handles the software upgrade and also migrates your databases.

- If you are currently running the free SQL Server 2005 Express but you intend to upgrade to a full (not free) version of SQL Server 2008, the situation is more complicated. The standard installation procedure for SQL Server 2008 will *not* upgrade from SQL Server 2005 Express. Therefore, we recommend that you first upgrade from SQL Server 2005 Express to SQL Server 2008 Express (as discussed above). Once you have installed SQL Server 2008 Express, you can install your full version of SQL Server 2008; the installation procedure will automatically upgrade the “Express” version into the full version, and do all the necessary database migration.

SQL Server Maintenance

From time to time, you should clean-up your SQL Server database. Clean-up operations can include reorganizing the database index, rebuilding the index from scratch, and various other actions.

If you have a full SQL Server (as opposed to SQL Server Express), you can arrange clean-up operations through SQL Server Management Studio. To do so, follow these steps:

1. Login to an account with Windows Administrator privileges.
2. Start SQL Server Management Studio. (On Vista, you should use **Run as Administrator**.)
3. In the left-hand panel, expand the entry for **Management**.
4. Right-click the entry for **Maintenance Plans**, and click **Maintenance Plan Wizard**.

The wizard will then walk you through the process of creating a maintenance plan that performs various operations automatically. During the process, you can choose what operations will be performed. As a minimum, we recommend the following:

- Check Database Integrity
- Reorganize Index
- Rebuild Index

Clean Up History
Maintenance Cleanup Task

(You can also include a backup operation, although you should already be backing up your database with a different maintenance plan—you don't need to clean your database as often as you need to back it up.)

Once you've used the wizard to create the maintenance plan, you can execute the operations by right-clicking the entry for the maintenance plan, then clicking **Execute**.

If you ever have a system crash or power failure on the server computer, you should make a point of cleaning up your database as soon as possible afterward.

If you have SQL Express, the associated Management Studio does not allow you to make maintenance plans. However, you can perform similar clean-up maintenance operations using the `expressmaint` software discussed in *SQL Express Considerations* on page 52. For more information on cleaning up the database, see the `expressmaint` documentation.

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 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 or Server 2008:
 - In the Control Panel, click **Windows Firewall**.
 - In the resulting window, click **Change settings**.
2. On Windows XP or Server 2003:
 - 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.

There are two types of pings: `ping -4` and `ping -6` (which test for IP4 and IP6, respectively). SQL Server generally uses IP4, so you usually want to use `ping -4`.

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. (You can save the output of `net use` in a file by using a command line of the form `net use >file`).
 - 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 67* 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.

Comparing the DNS Name and Active Directory Name

It is possible for the DNS name of a system to get out of synch with the Active Directory name. (We have observed that this happens relatively often on Vista, though not on XP.) If the names get out of synch, all kinds of anomalous problems may occur. Therefore, it's important to make sure these names refer to the same thing. To do this, follow these steps (on both the server computer and the remote computer where you're having trouble):

1. Open a Command Prompt window.
2. Type the following:

```
ipconfig
```

The system will display a list of information, including the IP address of the computer.

3. Type the following:

```
nslookup
```

The system will display some information, then type `>` to prompt for more input.

4. Type the name of the current computer (then press **<Enter>**). The computer will type the IP address of the computer.
5. Type `exit` to quit the `nslookup` program.

Both `ipconfig` and `nslookup` should display the same IP address for the computer. If not, the names are out of synch and you'll have to reset one or the other. For information on resetting these names, see your Windows documentation.

SQL Server Configuration

As noted in *SQL Server Installation on page 50*, 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** or **Microsoft SQL Server 2008** (whichever you are using). 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** or **SQL Server 2008 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. See below for more on named pipes.

5. In the left-hand side of the window, click **SQL Server 2005 Services** or **SQL Server 2008 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.

Named Pipes: As noted above, you have the option of allowing SQL Server to use named pipes in addition to TCP/IP. We recommend that you do not enable named pipes. The reason is that each named pipe counts against the maximum number of sessions that Windows allows, while TCP/IP connections do not. Therefore, using named pipes decreases the number of people who can use the Windows system for other purposes.

If you *do* enable named pipes (e.g. because some other software using SQL Server requires them), it may affect the performance of MainBoss. The reason is that if you enable both named pipes and TCP/IP, SQL Server tries to use named pipes first. In our experience, the Windows firewall often blocks named pipes. This leads to a situation where SQL Server is trying to use named pipes, while the firewall won't allow it.

It may take some time for SQL Server to recognize that named pipes aren't going to work. After SQL Server realizes the problem, it will use TCP/IP...but in the meantime, MainBoss may time-out because SQL Server is taking so long to answer. What you'll see is that MainBoss takes a very long time to start up; then MainBoss displays a diagnostic message saying that it has timed out and puts you into the "Select Maintenance Organization" window. If you try to open the same maintenance organization, the operation should succeed because SQL Server will have switched over to TCP/IP.

As this discussion shows, we see no good reason for using named pipes and a number of reasons not to. Avoid them if possible.

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 53*.

Starting the SQL Server Browser: The SQL Server browser makes it possible for other computers to detect SQL Server on the server system. If the browser is not running, instances of SQL Server on this computer will not appear in the drop-down list for "**Database Server**" when you are specifying a maintenance organization. (See, for example, *Creating a Maintenance Organization on page 9*.)

By default, the browser is installed when you install the rest of SQL Server. For full versions of SQL Server, the browser is also set up to start automatically. However, for SQL Express, the browser is *not* set up to start automatically. To set it up that way, follow these steps:

1. In the Windows **Start** menu, locate **Microsoft SQL Server 2005** or **Microsoft SQL Server 2008** (whichever you are using). Under this entry, locate **Configuration Tools** and

then **SQL Server Configuration Manager**. This opens a window where you can configure your SQL Server.

2. In the left-hand panel, click **SQL Server 2005 Services** or **SQL Server 2008 Services**.
3. In the right-hand panel, examine the entry for **SQL Server Browser**. If the state is *Running*, the browser is already configured to be active. You can quit and skip the remaining steps below.
4. If the state is *Stopped*, you must set up the browser for automatic start. Right-click the entry for **SQL Server Browser**, then click **Properties**.
5. In the resulting window, go to the **Service** section.
6. Click the entry for **Start Mode**. Click the value associated with this entry; this opens a drop-down list. In the list, click *Automatic*.
7. Click **OK**.
8. When you return to the previous window, right-click the entry for **SQL Server Browser**. In the resulting menu, click **Start**.
9. Exit the Configuration Manager.

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** or **SQL Server 2008 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 or Server 2008:
 - In the Control Panel, click **Windows Firewall**.
 - In the resulting window, click **Change settings**.
2. On Windows XP or Server 2003:
 - 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:
 - Double-click on `sqlbrowser.exe`. A window will open to show you the full pathname of the file. Check that the pathname refers to the SQL Server you will be using. (This is necessary in case you have multiple versions of SQL Server on your computer, or if the entry for `sqlbrowser.exe` refers to the software for some previous version of SQL Server.) If so, click **OK** to exit.
 - Repeat the above procedure for `sqlservr.exe`.
5. If `sqlbrowser.exe` and `sqlservr.exe` are both in the list, both checkmarked, and both refer to the correct programs, you're finished—click **OK** to exit.
6. If the correct files are in the list but not checkmarked, checkmark them then click **OK** to exit.
7. If one or both file names are missing, click **Add Program**.
8. In the resulting window, click **Browse**.
9. 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**.
10. You'll return to the previous window. Click **OK**. The file is added to the list.

11. Repeat steps 7-10 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 and Server 2008, 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.msp>

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 Service** 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 20*). 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.

Odd Processing by Outlook/Exchange: Several MainBoss customers have encountered an odd behavior in Outlook and/or Exchange where HTML messages are mistaken for plain text. As a result, Outlook/Exchange tells MainBoss that a message is plain text, when it's really HTML; the result is that MainBoss displays mail messages as raw (hard-to-read) HTML, rather than processing them correctly. For help in fixing this behavior, see our online FAQ article at:

http://www.mainboss.com/english/support/faq/mb30_outlookhtml.shtml

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 26*). 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 and Server 2008).
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 20*, 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.

Appendix C: MainBossRemote

This appendix offers instructions on setting up a web site for use with the MainBoss Advanced **Web Requests** and **Web Access** modules. These modules provide the ability to deal with work orders and requests through the Internet. The web-based facilities of MainBoss Advanced are supported by software called **MainBossRemote**.

Before setting up MainBossRemote, first make sure that MainBoss Advanced is working properly through direct network access (as described in the main body of this guide).

MainBossRemote System Requirements

MainBossRemote is an ASP.NET web application which must run on a Windows system. This means that the web site used by MainBossRemote must be hosted on a Windows system and must use the IIS web server.

We recommend that you use IIS 7 running on either Windows Server 2008 (with Service Pack 2 or later) or R2.

IIS 6 on Windows Server 2003: MainBossRemote is also supported on IIS 6, running on Windows Server 2003 (with Service Pack 2 or later). However, setting up and configuring MainBossRemote on IIS 6 is a complicated procedure that requires IT personnel to make several important implementation decisions. These decisions may have significant side-effects, especially if other applications use the same IIS web server. For this reason, we recommend that IT personnel consult with MainBoss Support before attempting set-up and configuration. Once you have determined which choices are best for your needs, MainBoss Support personnel will guide you through the configuration process. *Note that a fee may be charged for this consultation, especially if the IT personnel need a significant amount of help working their way through the process.*

Thinkage does not support the installation of MainBossRemote on the workstation versions of Windows XP, Windows Vista or Windows 7.

Whatever version of IIS and Windows you use, you should use Windows Update to keep the software up to date.

License Keys

Before you begin setting up MainBossRemote, record your **Web Requests** and/or **Web Access** license keys in the MainBoss database. To do this, follow the instructions in *Entering License Keys on page 10*.

You cannot use **Web Requests** or **Web Access** without appropriate license keys. To obtain such keys, contact your MainBoss dealer or e-mail info@mainboss.com.

Initial Set-Up

Create a folder which will hold all the files required to support the web site. By default, IIS runs from a folder named `C:\inetpub`; therefore, you might run MainBossRemote from a subfolder called

```
C:\inetpub\wwwroot\MainBossRemote
```

Very Important: Set-up will be much easier if you set up the IIS server on the same computer where you run the SQL Server for MainBoss's database. Having the two on different computers adds many complications to the configuration process.

There are two ways to obtain the MainBossRemote software:

From the mainboss.com web site:

If you obtain the software from our web site, follow the instructions on the web page to download the files and copy them to the appropriate folder.

From a MainBoss DVD distribution disk:

All the files needed for MainBossRemote will be on the DVD in an appropriate directory. (When you run the DVD, it will tell you where this directory is.) Copy all the files in this directory to the appropriate folder on the server computer.

In setting up IIS, follow Microsoft's documentation. Considerable differences exist between the various Windows systems that might be used. For a discussion of IIS configuration, see

<http://msdn.microsoft.com/en-us/library/ms178477.aspx>

The above web page contains links that explain how to start the IIS manager on supported versions of Windows.

Note: The rest of this appendix applies to set-up and configuration using IIS 7 on Windows Server 2008 (SP2 or R2).

Security Considerations

Before proceeding to configure your MainBossRemote web site, you must make some decisions about security. This section discusses various factors that you should consider.

Depending on your ways of working, especially with cell phones and similar devices, you may find you have to sacrifice some level of security in order to get anything done. You may decide this means you won't use cell phones with MainBossRemote; alternatively, you may decide to live with reduced security or other complications. Thinkage takes no responsibility for any problems you may encounter if you opt for reduced security.

Web Requests: In order to use the **Web Requests** module, users must type in their e-mail address and send it to the MainBossRemote web page. This process must have anonymous authentication: anyone who can access the web page is allowed to type in an e-mail address and submit it. (Processing only continues if the e-mail address is authorized to submit requests.)

If you do not have the **Web Requests** module, you do not have to allow anonymous authentication to this particular facility.

Web Access: In order to use the **Web Access** module, your MainBossRemote web site must be able to authenticate people as Windows users on the system where the web site runs. This means that people must enter their Windows login names and passwords.

The safest way to do this is to use ASP.NET's integrated Windows Authentication. When users attempt to use the **Web Access** module, they will be asked to send the login names and passwords, which will be securely encrypted when transmitted to the MainBossRemote site.

Unfortunately, integrated authentication does not work with some cell phone and PDA services. Whether or not the device itself can handle encryption, the service provider may use proxy servers that do not support integrated authentication. (This may mean that the device works fine when connected to your own Wi-Fi network but not when connecting through the device's usual service provider.) The symptom of this is that you aren't asked for your login name and password; you simply get a "permissions denied" message.

If the cell phones or PDAs that you intend to use can't handle integrated authentication, you must consider your options.

- One choice would be not to support access through these devices—users would only access MainBossRemote through networked computers or through handheld devices that *do* allow integrated authentication.
- The other choice would be to use "classical" authentication. If you go this route, you have two further options:
 - Using SSL (Secure Socket Layer) or TLS (Transport Layer Security). Users will access the MainBossRemote web site with `https` rather than `http`, as in `https://www.ourorg.com/MainBossRemote`

If the device supports this facility, login names and passwords will be encrypted when transmitted.

In order to take this approach, IT personnel will be required to set up SSL/TLS

access, which includes the use of a security certificate. See below for more details.

- No encryption. In this case, login names and passwords will be sent in plain text through wireless connections. This leaves you seriously exposed to wireless eavesdroppers, and is therefore not recommended.

Security Certificates: A security certificate may be obtained from a trusted Certification Authority (CA) or may be self-generated. A CA security certificate may cost a lot of money; however, large organizations often have a CA certificate already, in which case the same certificate may be used for MainBossRemote.

You can create your own self-generated security certificate using the IIS 7 manager. (See [Initial Set-Up on page 77](#) of this guide for a reference on how to start the IIS manager.) Once you've started the manager, click the entry for the server in the left-hand panel, then click **Server Certificates** in the **IIS** section of the middle panel. In the resulting window, click **Create Self-Signed Certificate** (in the right-hand panel) to create a self-signed certificate.

If you use a self-signed certificate, devices using `https` to connect to your MainBossRemote web site must be told to trust this certificate. Note that most browsers display strong warning messages when a user first tries to connect with a web site that has a self-signed certificate; therefore, users must be reassured that connecting with your site really is secure.

If you are using SSL/TLS, make sure that your firewall allows such communications through. Typically, SSL/TLS use port 443, so the firewall should allow connections to this port (if you wish to open your web site to outside connections).

Application Pools

Roughly speaking, an application pool consists of one or more web applications that share processing time together as a single process. You must decide whether to put MainBossRemote into an existing application pool or to create a new pool that only contains MainBossRemote. The following considerations apply:

- Each application pool takes up a considerable amount of memory. Therefore, you can reduce memory requirements if you put MainBossRemote into an existing pool.
- Since all applications in a pool must share the processing time allocated to that pool, MainBossRemote will have to share time with everything else in the pool. In particular, if you put MainBossRemote into a pool that contains a high-traffic process, the performance of MainBossRemote will suffer—the high-traffic process will monopolize allotted processing time.

Perhaps the ideal is to put MainBossRemote into an existing pool whose other members only have a modest amount of traffic.

Application Pool Identity: Each application pool must have an *identity*. This corresponds to a login name that will be used to run all the applications in the application pool. The login

name must have sufficient permissions to run all the applications in the pool. In particular, the identity login name for the pool that contains MainBossRemote must have all the permissions required by MainBoss.

Due to a problem in SQL Server, the pool containing MainBossRemote must run under the **NetworkService** identity. (For more information about this problem, see <http://support.microsoft.com/default.aspx/kb/2002980>.) The **NetworkService** identity corresponds to the login name **NT AUTHORITY\NETWORK SERVICE**, which is a name automatically built into Windows. If you do not have any existing application pools that run under this identity, you must create a new pool that does. Otherwise, you can add MainBossRemote to any existing pool that has the **NetworkService** identity.

Important: Whatever application pool you use for the web site, we recommend that it should run in the *Integrated* managed pipeline mode. This ensures appropriate authentication security. However, as discussed in *Security Considerations on page 77*, some cell phone and PDA services do not support integrated authentication, so you may have to choose a different authentication scheme.

To work with application pools, you use the IIS manager. Start the manager on the computer where you will run the web server. (See *Initial Set-Up on page 77* of this guide for a reference on how to start the IIS manager.)

If you decide to create a new application pool for MainBossRemote, you must create the pool:

1. In the left-hand panel of the IIS manager, expand the entry labeled with the name of the computer.
2. Right-click on **Application Pools**. In the resulting menu, click **Add Application Pool**.
3. In the resulting window, enter a name for the pool (e.g. MainBoss). The .NET Framework version should be the highest available and the managed pipeline mode should be *Integrated* (or *Classical* if you make that security choice). When you click **OK**, the pool will be created, using default settings.
4. The new pool is automatically given a default identity. Typically, this is either **NetworkService** or **ApplicationPoolIdentity**. If the default identity is not **NetworkService**, you must change the pool's identity to **NetworkService** by following these steps:
 - Right-click on the name of the pool in the list of application pools.
 - In the resulting menu, click **Advanced Settings**.
 - In the resulting window, click the entry for **Identity** (under **Process Model**).
 - Click the "..." button on the right-hand side of the field.
 - In the resulting window, click **Built-in Account**, drop the arrow, and select **NetworkService** from the list.
 - Keep clicking **OK** until you get back to the IIS manager.

Web Sites vs. Applications

The next decision to make is whether you want to implement MainBossRemote using a separate web site or an application within an existing web site.

- A separate web site has its own unique IP address and/or port number on the server computer.
- An application uses the same IP address and port number as an existing web site on the server computer. (Loosely speaking, an application is a type of *virtual directory*.)

If you already have a web site on the server computer, we recommend that you implement MainBossRemote as an application under that web site. The only reason to give MainBossRemote its own separate web site is if the server computer doesn't have any existing web sites, or if there is an incompatibility between the needs of MainBossRemote and the configuration of the existing sites.

Adding and Configuring an Application

This section describes what to do if you decide to implement MainBossRemote as an application under an existing web site.

As noted in a previous section, you should have copied the MainBossRemote files to a subfolder under the folder used by the IIS web service. If you have not done so, do so now.

Start the IIS manager. In the left-hand panel of the IIS manager, expand the entry for the server computer, then click on **Sites**. Expand the entry for sites until you see the name of the web site under which you will put MainBossRemote.

In order to add MainBossRemote to this web site as an application, **you must stop the web site** (i.e. shut down the web site temporarily). To stop the web site, click on its entry in the left-hand panel of the IIS manager, then click **Stop** in the far right-hand panel.

Once you've stopped the web site, right-click on the web site's entry in the left-hand panel. In the resulting menu, click **Add Application**. This opens a window where you can specify information on MainBossRemote.

- In "**Alias**", specify a name for the application. This is simply a name that will be displayed in the IIS manager to refer to this application; it can be any name that isn't already in use.
- Click the **Select** button associated with the "**Application pool**" field and specify the name of the application pool you have chosen for MainBossRemote—either an existing pool or one you created expressly for MainBossRemote.
- In "**Physical Path**", specify the folder that contains the MainBossRemote files (e.g. C:\inetpub\wwwroot\MainBossRemote).

- Click **OK** to close the window.

Once you have configured the web site, you can restart the web site. In the left-hand panel, click the entry for the web site that contains the MainBossRemote application. Then, in the far right-hand panel, click **Start**.

Adding and Configuring a Web Site

This section describes what to do if you decide to implement MainBossRemote as a separate web site.

As noted in a previous section, you should have copied the MainBossRemote files to a subfolder under the folder used by the IIS web service. If you have not done so, do so now.

Start the IIS manager. In the left-hand panel of the IIS manager, expand the entry for the server computer, then click on **Sites**. In the far right-hand panel, click **Add Web Site**. This opens a window where you can specify information on MainBossRemote.

- Click the **Select** button associated with the “**Application pool**” field and specify the name of the application pool you have chosen for MainBossRemote—either an existing pool or one you created expressly for MainBossRemote.
- In “**Physical Path**”, specify the folder that contains the MainBossRemote files (e.g. C:\inetpub\wwwroot\MainBossRemote).
- In “**Host Name**”, enter the name through which users will access your MainBossRemote site.
- In “**Site Name**”, enter MainBossRemote (or some other suitable name). This name will be used in the URL for accessing the MainBossRemote services. Specifically, users will access MainBossRemote using the URL

<http://HOSTNAME/SITENAME>

Therefore if “**Host Name**” is ourcomputer.ourdomain.com and “**Site Name**” is MainBossRemote, users will access the software with the URL

<http://ourcomputer.ourdomain.com/MainBossRemote>

- In “**Binding**”, specify the appropriate information (e.g. the port number the web site should use). The default port number is 80, and we strongly recommend that you use this default—anything else will make the URL more complicated, since the port number will have to be specified explicitly. (Also, some cell phone services do not support connections to non-standard port numbers.)

Note that if you’re using SSL or TLS security (as discussed in *Security Considerations on page 77*), you specify https as the binding type and then specify an SSL certificate using the drop-down list.

- Click **OK** to close the window.

Once you have configured the web site, you can start it. In the left-hand panel, click the entry for the web site. Then, in the far right-hand panel, click [Start](#).

Authentication Settings

Whether MainBossRemote is a separate web site or an application within another web site, you must set its authentication settings (i.e. how the software determines who is or isn't an authorized user).

In the left-hand panel of the IIS manager, expand **Sites** and click on the name of the MainBossRemote web site or application.

In the middle panel of the IIS manager, the IIS section has an entry for *IIS Authentication*. Double-click this entry to open a window where you can specify authentication settings. We recommend the following:

- Enable Anonymous Authentication
- Enable Windows Authentication
- Disable ASP.NET Impersonation (this is the default)

Application Settings

Whether MainBossRemote is a separate web site or an application within another web site, you must set its application settings. These specify which database the software should access.

In the left-hand panel of the IIS manager, click on the name of the MainBossRemote site. In the middle panel of the IIS manager, the ASP.NET section has an entry for *Application Settings*. Double-click this entry to open a window where you can specify the settings.

You will see there are settings for `dbserver` and `dbname`. Initially, these have stand-in names that should be replaced with the real names used by your site.

Click on the entry for `dbserver`. In the far right-hand panel, click [Edit](#). In the resulting window, set “**Value**” to the name of the SQL Server that maintains the MainBoss database. This name will have one of the following two forms:

COMPUTER\INSTANCE

The name of the computer where the SQL Server resides and the name of the particular SQL Server instance that maintains the MainBoss database.

COMPUTER

Just the name of the computer where the SQL Server resides. This indicates that the MainBoss database is maintained by an unnamed instance of SQL Server.

Click [OK](#) once you have entered the appropriate “**Value**”.

Now click on the entry for `dbname`. In the far right-hand panel of IIS manager, click [Edit](#). In the resulting window, set “**Value**” to the name of the MainBoss database as maintained by the server specified with the `dbserver` setting. Click [OK](#) once you have entered the appropriate “**Value**”.

After setting this information, you may have to stop and start the web site in order for the settings to take effect.

Software Initialization

The first time someone connects with the MainBossRemote web site, the software must compile itself and perform various set-up processes. This can take a long time and use a lot of computer memory. Therefore, you should make the first connection at a time when the increased processor and memory load will not affect important work on the computer.

Once the software has set itself up, it doesn't have to go through the process again unless something significant changes in your configuration, e.g. you install a new version of the web server, SQL Server, or MainBoss.

Accessing SQL Server on a Different Computer

As noted previously, configuring MainBossRemote is easier if the web server is on the same computer as the SQL Server that manages MainBoss. If the two are on different computers, more configuration is required.

1. Create a domain account that will be used to login to the SQL Server. For the sake of illustration, we will call this account `IIS_MAINBOSS`. Also Give the account a suitable password. (Be sure to remember this password—you may need it in future.)
2. Give `IIS_MAINBOSS` Domain User permissions.
3. On the SQL Server computer, start SQL Server Management Studio.
4. Using Management Studio, configure the SQL server's logins to allow `IIS_MAINBOSS` to login. (Note that you do *not* need to add `IIS_MAINBOSS` to MainBoss's own list of authorized users, given in [Administration | Users](#).)
5. Using Management Studio, add `IIS_MAINBOSS` to the list of accepted users of the MainBoss database.
6. Using Management Studio, give `IIS_MAINBOSS` the `MainBoss` security role.
7. Start the IIS manager. In the left-hand panel, expand **Sites** and click on the MainBossRemote web site. In the middle panel, IIS section, double-click *Authentication*.

In the resulting window, click *ASP.NET Impersonation*, then click [Edit](#) (in the far right-hand panel).

In the resulting window, click **Specific user**. Enter the IIS_MAINBOSS login name in the accompanying field, using the format DOMAIN\IIS_MAINBOSS, where DOMAIN is the name of the domain where IIS_MAINBOSS is defined. You will be asked to enter the login name's password.

Note: The password for this login name will be stored in plain text (not encrypted or hidden) in the web.config file. For information on how to fix this, see <http://support.microsoft.com/default.aspx?scid=kb;EN-US;329290>.

8. On 64-bit versions of Windows, give DOMAIN\IIS_MAINBOSS read/write file system privileges to the directory

%SystemRoot%\Microsoft.NET\Framework64\versionNumber\Temporary ASP.NET Files

On 32-bit versions of Windows, do this for

%SystemRoot%\Microsoft.NET\Framework\versionNumber\Temporary ASP.NET Files

MainBossRemote Troubleshooting

This section lists a number of error messages you might see when trying to use MainBossRemote, plus ways to correct the problems.

There was a problem validating access to the database XXX on server YYY. The database-principal 'dbo' does not exist or user is not a member. Failed SQL command is: declare @result int; exec @result = sp_getapplock 'MainBossSession', 'shared', 'session', 0, 'dbo'; select @result

This indicates you did not give permissions to the login name for the identity of the application pool where MainBossRemote runs. Follow the instructions in *Application Pools on page 79*.

An error occurred while processing your request. Database server 'ServerMainBossDatabaseIsOn' was not found or was not accessible.

This indicates that you did not change the default dbserver application setting. See *Application Settings on page 83* of this guide.

Could not load file or assembly 'System.Web.Mvc' or one of its dependencies. Access is denied.

This indicates that you did not give DOMAIN\IIS_MAINBOSS read/write file system privileges to the directory containing temporary ASP.NET files. See the previous section of this guide for more information.

Note: Whenever you encounter diagnostic messages, you can usually find useful information by typing them into Google and searching for information.

The Windows Firewall: In our experience, many of the difficulties that arise when configuring MainBossRemote are due to firewall problems. If the Windows firewall is not adjusted to allow various information to pass through, MainBossRemote will not work. Firewall problems are especially difficult to diagnose because they generate no error messages or log entries; you just don't see anything happening.

Cookie Usage

Cookies used by MainBossRemote are stored as subkeys under the cookie key MainBossRemote. The following subkeys are defined:

`requestorEmail`

The email address that a user last entered, provided that the address was recognized as a valid requestor in MainBoss.

INDEX

- .NET framework, 1
- @Requests, 75
 - configuring, 22
 - edit, 22
 - event log, 26, 74
 - installing, 20
- abbreviations, 40
- active directory, 62
- add organizations, 45
- administration, 31
 - @Requests, 21
 - backups, 34, 52
 - database history, 35, 52
 - licenses, 11
 - security roles, 16
 - users, 14, 18
- administration backups, 38
- administrator privileges, 28
- application pools, 79
- applications, 81
- ASP.NET, 76
- backups, 34, 46, 51
 - database, 36
 - restoring, 37
- breakin, 47
- browser
 - SQL Server, 64
- clear service association, 28
- ClickOnce, 6
- ClickOnce deployment, 4
- command files, 39
- command line interface, 39
- command line options, 32
- CultureInfo, 33
- currency, 9
- database, 2
 - backup, 36
 - database history, 52
 - database name, 38
 - database server, 38
 - desktop icon, 33
 - DNS name, 62
 - domain, 2
 - domains, 16
- evaluate security as, 18
- Excel, 43
- exporting data, 41
- expressmaint, 52, 56
- file-sharing, 58
- firewall, 58, 65, 66
- hardware requirements
 - MainBoss, 1
 - SQL Server, 49
- identity, 79
- IIS, 76
- IIS 6, 76
- IIS 7, 76
- IMAP4, 73
- importing data, 44
- installation
 - ClickOnce, 6
 - MainBoss software, 6
 - manual, 5
- instance, 68
- instances, 50
- international support, 33
- Internet security, 77

- language support, 33
- license keys, 10
 - MainBossRemote, 76
- Linux, 2
- local users, 16

- MainBoss manages SQL Security, 68
- MainBossRemote, 76
 - application settings, 83
 - authentication settings, 83
 - URL, 82
- maintenance
 - SQL Server, 55
- maintenance organizations
 - creating, 9
 - multiple, 30
- maintenance plans, 51
- mbutility, 39
 - adding organizations, 45
 - AddOrganization, 45
 - Backup, 46
 - breakin, 47
 - built-in schemas, 41
 - Excel, 43
 - export, 41
 - GenerateImportSchema, 41
 - Help, 40
 - import, 44
 - ListImportSchemas, 41
 - ListOrganizations, 40
 - MainBoss organizations, 40
- Microsoft Report Viewer, 3
- Microsoft Word, 71
- modes
 - administration, 31
 - MainBoss Requests, 31
 - start, 31
 - view sessions, 31

- named pipes, 64
- NetworkService, 80

- organization name, 38

- ping, 58
- POP3, 73
- privileges, 28

- remote registry, 27
- report viewer, 1
- Report Viewer, 3
- requests
 - mode, 31
- reset to user security, 18
- restoring, 52
- restoring from backup, 37
- roles, 13

- Samba, 2
- schemas, 41
- scopes, 12
- script, 39
- security, 77
- security roles, 13, 16
 - testing, 18
- security settings, 12
- shell script, 39
- Silverlight, 52
- SQL, 2
- SQL Express, 3, 52
- SQL Security, 68
- SQL Server, 50, 57
 - browser, 64
 - instances, 50
 - maintenance, 55
- SQL Server upgrade, 54
- start modes, 31
- system requirements
 - MainBossRemote, 76
- system update required, 4, 8

- troubleshooting, 57
- trusted domains, 12

upgrading MainBoss, 30
upgrading SQL Server, 54
users, 14

view sessions, 31

virtual directory, 81

Web Access module, 76
Web Requests module, 76
web sites, 81