

# **MainBoss @Requests 1.2 User's Guide**

Copyright © 2007, Thinkage Ltd.



## Table of Contents

<b>Welcome to MainBoss @Requests .....</b>	<b>1</b>
<b>Basic Principles of MainBoss @Requests .....</b>	<b>2</b>
The Requestor Database.....	2
The MainBoss @Requests Program.....	3
Acknowledgements .....	3
<b>Installing MainBoss @Requests.....</b>	<b>5</b>
Registering MainBoss @Requests .....	6
<b>Configuring MainBoss @Requests .....</b>	<b>7</b>
The Configuration Program .....	7
Quitting the Configuration Program .....	9
Creating the Requestor Database .....	9
The Diagnostic File .....	11
Specifying E-mail Information.....	11
@Requests Configuration Preferences.....	11
Message Logs.....	13
Printing Message Logs .....	13
<b>Using MainBoss @Requests .....</b>	<b>14</b>
Reading Request Messages .....	15
Rejecting Requests .....	15
Updating Logs .....	15
Sending Acknowledgements .....	16
Cleaning Up Message Folders.....	16
<b>The @Requests Command Line .....</b>	<b>18</b>

## Introduction

# Welcome to MainBoss @Requests

Thank you for licensing MainBoss @Requests. We think you'll find it provides all the features you need for managing your organization's maintenance work.

**Note:** This guide to MainBoss @Requests assumes you are already familiar with MainBoss Maintenance Software. For a full explanation of MainBoss itself, see the MainBoss User's Guide and its online documentation.

## Basic Principles of MainBoss @Requests

MainBoss @Requests is an optional add-on to the regular MainBoss CMMS package. The software makes it possible for people in your organization to submit work requests through e-mail or by means of a web page.

In order to use MainBoss @Requests, your organization must create an e-mail address where requests can be sent. This address is called the *Request Mail Receiver*. The MainBoss @Requests software monitors the Mail Receiver address. When a message is received, the message is converted into a normal MainBoss work request and placed into your MainBoss database. The work request can then be handled the way you handle work requests from other sources.

Submitted e-mail messages may be *basic* or *formatted*:

- A basic message is typically just a few sentences describing some problem that the sender wants to report. The subject line of this message will be used as the subject line in the resulting work request, and the body of the message will be used as the description. Other information for the work request will be taken from the Requestor Database (described later on).
- A formatted message submits information in a specific format recognized by the MainBoss @Requests software. Formatted messages may be created by filling out a form on a web page or using special request submission programs.

### The Requestor Database

MainBoss @Requests has a database that can be used to record information about the people who submit e-mail work requests. This database is organized according to e-mail address. For example, suppose you receive a message from `jsmith@abc.com`; your database might record the name of the person who has that e-mail address and the person's phone number. This information can be used to fill in the appropriate fields of any work requests created from that person's e-mail.

You can configure MainBoss @Requests so that it only accepts e-mail requests from people who are already in the requestor database. This prevents unauthorized people from submitting requests. Alternatively, you can set things up so that e-mail requests are accepted from anyone. In this case, you can ask MainBoss @Requests to create a database entry for any submitter who is not already in the database.

<p><b>Note:</b> When you license MainBoss @Requests, you license up to a maximum number of users. For example, you might license for a maximum of 10 authorized requestors.</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## The MainBoss @Requests Program

The MainBoss @Requests program processes e-mail submissions and creates work requests based on those submissions. If you have a local area network, the program must be installed on a computer with:

- Appropriate permissions to access e-mail sent to the Request Mail Receiver.
- Appropriate permissions to access your regular MainBoss database.

The @Requests program typically runs continuously. Every so often, the program checks to see if the Request Mail Receiver has received any new e-mail messages; if so, the program creates appropriate work requests from the new messages and sends mail to the submitters telling them that their requests have been received. In this way, @Requests deals with requests soon after they are received.

**Note:** You use a configuration program to dictate how often @Requests “wakes up” to do its work. For example, you can tell @Requests to check for incoming messages every fifteen minutes, every three hours, once a day, or any other schedule that suits your operation’s needs. For further information on scheduling @Requests, see [@Requests Configuration Preferences on page 11](#).

Instead of running the @Requests program automatically, you might choose to run it manually. In this case, the program only processes incoming requests when you tell it to do so. This gives you more control over what @Requests does, but it also means that incoming requests won’t be handled until you tell @Requests to do so.

@Requests stores configuration information and other types of data in the directory containing the MainBoss database.

**Important:** You cannot have more than one copy of the @Requests program running at the same time. If you have one copy running and try to start another copy, the second copy will not work.

## Acknowledgements

@Requests uses e-mail messages to inform the requestor about the progress of a work request. These e-mail messages are called *acknowledgements*.

Acknowledgements are sent at the following times:

- When @Requests successfully creates a work request based on the original e-mail submission.
- When @Requests sees that the work request has been transferred into a work order.
- When @Requests sees that the work order has been completed.

- If @Requests sees that the work request or work order has been voided.
- If @Requests sees that a completed or voided work request has been reopened.
- If @Requests sees that a work order has been rescheduled to have a different start time.

## Installing MainBoss @Requests

Before you install MainBoss @Requests, you must do the following:

1. Make sure you have installed the regular MainBoss package and that the MainBoss database is accessible from the computer where the server program will run.
2. Create a POP3 e-mail account that is accessible from the computer where the @Requests program will run. (For information on creating POP3 e-mail accounts, see the documentation for your e-mail server software, e.g. Mail Exchange Server, Lotus Notes Server, or SENDMAIL.)
3. Make sure that the computer where the @Requests program runs has access to an SMTP server. (For information on SMTP servers, see the documentation for your e-mail software.)

**Note:** If you obtain e-mail services direct from an ISP (Internet Service Provider), you may need to ask your ISP to create a new e-mail account that @Requests can use.

Once you have met these requirements, you are ready to install the software.

**If you download the package from the MainBoss web site:**

1. Follow the installation instructions on the web page from which you downloaded the software. The installation process will ask you to specify a directory where the software should be installed. It will also ask you for the directory that contains your regular MainBoss database. Once you have answered these questions, the Setup program will install the software.

**If you received the software on a CD:**

1. Insert the CD in the computer where you intend to run the @Requests program.
2. If the computer is configured to run CDs when they are inserted, the Setup program will start automatically. Otherwise, use Windows Explorer to browse the CD and execute the `StartUp.bat` file.
3. The installation process will ask you to specify a directory where the software should be installed. It will also ask you for the directory that contains your regular MainBoss database. Once you have answered these questions, the Setup program will install the software.

Once you have installed the package, you should register the software. For more on registration, see *Registering MainBoss @Requests* on page 6.



## Registering MainBoss @Requests

To register MainBoss @Requests, you must enter an appropriate license key in the MainBoss database. You will be sent a license key when you first license MainBoss @Requests; you will also be sent new license keys if and when you renew your yearly Annual Support Plan subscription.

To enter a license key, you must use the MainBoss program itself, not @Requests.

### □ To enter a new license key:

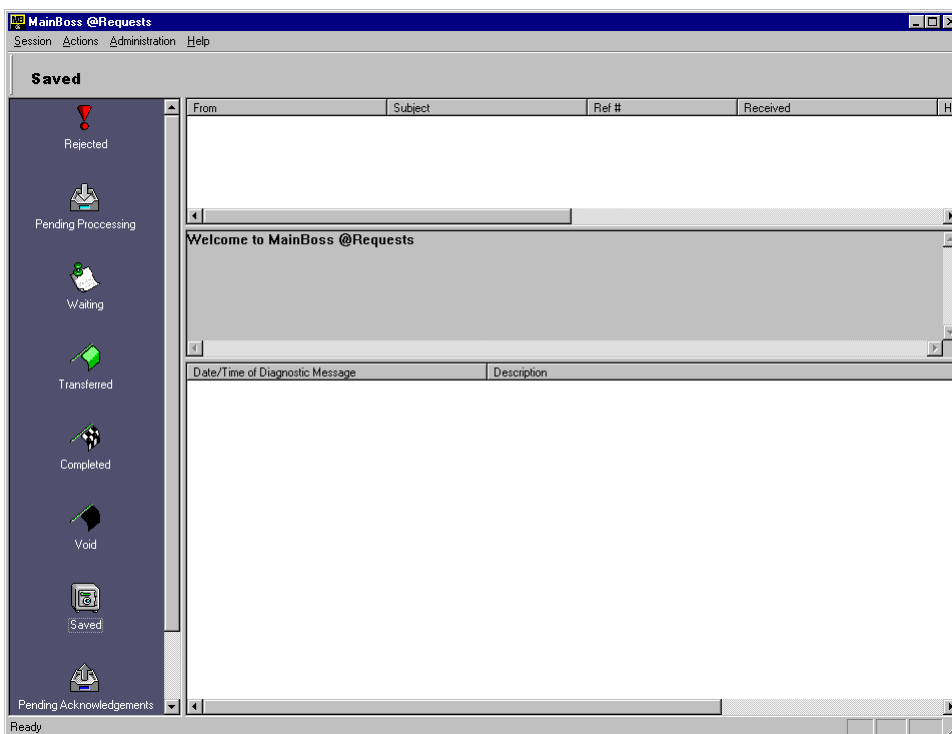
1. Start the MainBoss program by clicking the **Start** button on your Windows desktop, then click **Programs**, then **MainBoss**, then **MainBoss**.
2. Before you can enter the new key, you must login as ADMINISTRATOR. Therefore, click **Login** on the **Session** menu.
3. In “**User Name**”, type ADMINISTRATOR.
4. In “**Password**”, type the appropriate password. (This will be the password given to you when you installed the software, unless you have changed the password to something else.)
5. Click **Ok** to close the password window.
6. Click **License Keys** on the **Administration** menu. MainBoss displays the License Key browser, listing any existing license keys.
7. Click **New**. MainBoss opens a window where you can enter the new key.
8. In “**License Key**” type the license key you have been given by MainBoss headquarters.
9. Click **Save** to save the license key, then click **Cancel** to close the window.
10. Click **Done** to close the License Key browser.

## Configuring MainBoss @Requests

Once you have installed the package, you must *configure* the software. For example, you must tell the @Requests program the e-mail address of the Request Mail Receiver and specify one or more users from whom you will accept requests by e-mail.

### The Configuration Program

To configure the software, click on the normal windows **Start** button, then select **Programs** → **MainBoss** → **@Requests** from the menu. This starts the configuration program:



This window has four sections to it:

**Message Type Area** : The Message Type Area runs vertically down the left side of the window. When you click a particular type of message in this area, the Message List shows messages of that type.

**Message List**: The Message List is the top section to the right of the Message Type Area. It shows messages of the selected type.

**Preview Area**: The Preview Area appears below the Message List. It displays a brief description of the message that is currently selected in the Message List.

**Diagnostics:** Diagnostics are given in the bottom section to the right of the Message Type Area. Diagnostics may include error messages or warnings about special circumstances related to the server.

@Requests stores e-mail messages in a number of separate folders:

**Rejected:** Messages that were rejected. This may be because you have configured MainBoss @Requests to refuse requests from unauthorized users, or because the incoming request could not be processed for some reason. The HISTORY line in the Preview Area will explain why the request was rejected.

Note that users are *not* automatically notified if their requests are rejected.

**Pending Processing:** Messages that have been accepted but not yet been processed to create a MainBoss work request.

**Waiting:** Messages for which a work request has been generated and placed in the regular MainBoss database. The resulting work request has not yet been used to create a work order.

**Transferred:** Messages where the corresponding work request has been transferred into a normal MainBoss work order. The work order is currently open.

**Completed:** Messages where the corresponding work request has been transferred into a normal MainBoss work order, and the work order has been closed.

**Void:** Messages where the corresponding work request or work order was voided by a MainBoss user.

**Saved:** Messages that have been saved manually through the configuration program. Typically, you save messages after the corresponding work order has been completed. By saving a message, you retain a record trail but avoid having a lot of old messages in the **Completed** list.

**Pending Acknowledgements:** Messages for which an acknowledgement needs to be sent to the requestor.

For example, suppose that @Requests discovers that a particular work request has been voided. The @Requests program moves the corresponding message into the **Void** folder (to indicate that it has now been voided) and also places a copy of the message into the **Pending Acknowledgements** folder (since the submitter should be informed that the request was voided). Once the acknowledgement has been sent, the copy of the message is deleted from the **Pending Acknowledgements** folder; however, the original message will remain in the **Void** folder to show that the request was voided.

**Deleted:** Messages that were deleted manually through the configuration program. Typically, you delete messages if the corresponding work order has been completed and you don't need to retain a record trail. As with many e-mail programs, deleted messages remain in the deleted list so that they are still available. If you want to get rid of these messages permanently, delete the messages from the **Deleted** list.

For information on saving and deleting messages, see *Cleaning Up Message Folders* on page 16.

## Quitting the Configuration Program

To quit the configuration program, select **Session** → **Exit** from the configuration program menu.

## Creating the Requestor Database

The first step in configuring MainBoss @Requests is to specify information about requestors.

**Note:** When you license MainBoss @Requests, you license up to a maximum number of users. For example, you might license for a maximum of 10 authorized requestors (plus **Everybody**). The **Add** button will not let you add more requestors if you have already reached your licensed limit.

To specify information about requestors, select **Administration** → **Accounts** from the configuration program menu. This opens a window that contains the following:

**Requestor list:** The main part of the window is a list of people who are authorized to submit work requests via e-mail. The list gives each person's name, phone number, and other information including the e-mail address from which they will submit requests. When a request is submitted from an authorized e-mail address, @Requests will look up that address in the Requestor list and use all the associated information (name, phone number, etc.) to fill in fields when creating a work request.

**Disabled:** If this box is checkmarked, @Requests will reject any messages from the person who is currently selected in the Requestor list. If this box is not checkmarked, @Requests will accept requests from that person.

**Note:** It's unusual to disable a specific email address. More often, you disable the **Everybody** address (as discussed below). However, you might disable a specific address if you normally accept email from **Everybody**, but wish to reject email from a person who's become a nuisance.

**Everybody:** If you click this button, you can specify what to do with requests received from an email address that isn't currently in the Requestor list. Clicking the button opens a window containing the following:

**Disabled:** If you checkmark this box, @Requests will reject email from anyone who isn't currently on the Requestor list.

If the **Disabled** box is not checkmarked, anyone can submit requests. If you receive email from someone who is not currently in the Requestor list, @Requests creates an entry for that person. It also creates an entry for that person in the Contacts table of your MainBoss database (using the name from the `Name :` field of the email message).

If your Requestor list is already full (i.e. you've reached the maximum number of

requestors allowed), @Requests will not create a Requestor list entry. Instead, it rejects all email from anyone who isn't already on the list.

**Receive acknowledgements:** If this box is checkmarked, the **Receive acknowledgements** option (described below) is automatically turned on whenever a new requestor entry is created because of **Everybody**. If this box is not checkmarked, the **Receive acknowledgements** option is turned off.

**Ok:** Saves the options you have entered and closes the **Everybody** window.

**Cancel:** Closes the window without saving any changes you have made to **Everybody** options.

**Add:** Clicking this button opens a window that displays the Contacts table from your MainBoss database. Click a name from this table, then click **Ok**. The selected name will be added to your list of requestors.

**Note:** Normally, before you can add a name to the requestor list, it *must* be in your MainBoss Contacts table. To access the Contacts table, select **Tables** → **Contacts** from the MainBoss menu.

**Edit:** Clicking this button opens a window where you can change information about an existing requestor. The window contains the following:

**Receive acknowledgements:** If this box is checkmarked, e-mail acknowledgements will be sent to the requestor at various points in the request process (for example, when a work request is created from the original e-mail problem report). If the box is not checkmarked, the requestor will not receive acknowledgements.

**Delete:** Clicking this button deletes the currently selected entry in the Requestor list.

The window also has buttons that let you proceed to other steps in the configuration process or just to close the window.

**One Easy Way to Create Database Entries:** If you have not **Disabled** the **Everybody** account, @Requests automatically creates a new account each time a message is received from someone who is not already in the database. You can make use of this feature as an easy way to create database entries, especially when you intend to authorize a lot of people to submit requests.

Instead of trying to create database entries for everybody in advance, simply wait until users actually submit requests. The first time each person submits a request, @Requests automatically creates an entry in the Requestor database.

**Note:** Your license for MainBoss @Requests imposes a limit on the number of entries allowed in the Requestor Database. If you have already reached that limit, new requestors cannot be added. If a message comes in from someone who is not already in the list, the message will be rejected, even if you are accepting messages from **Everybody**—if @Requests has reached its limit, it will not create a new account and it will reject requests from new users.

## *The Diagnostic File*

@Requests maintains a file named `diagnostic.txt` in the directory that contains the requestor database. @Requests uses this file to record any error messages that occur when processing requests; this information may be useful if you find @Requests is having problems.

## **Specifying E-mail Information**

The next step in configuring @Requests is to specify information used in sending and receiving e-mail. To do this, select **Administration** → **E-Mail** from the configuration program menu. This opens a window that contains the following:

**Incoming:** Information in this area allows @Requests to receive incoming e-mail.

**POP3 Server:** The location (file name) of the POP3 mail server program available to this computer.

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

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

**Outgoing:** Information in this area allows @Requests to send e-mail acknowledgements to people who have submitted work requests.

**SMTP Server:** The location (file name) of the SMTP mail server program available to this computer.

**Note:** For information on the POP3 and SMTP servers, see the documentation for your e-mail software.

The window also has buttons that let you proceed to other steps in the configuration process or just to close the window.

## **@Requests Configuration Preferences**

The next step in configuring @Requests is to specify preferences that control the way the software works. To do this, select **Administration** → **Preferences** from the configuration program menu. This opens a window that contains the following:

**Send/Retrieve email and update logs every:** This controls how often @Requests checks for new requests. There are two associated fields: one where you can type in a number and a drop-down list where you can select units (minutes, hours, days or weeks). For example, you might tell @Requests to check for new requests every 15 minutes or every 2 hours.

Every time @Requests checks for new requests, it also checks on the status of requests that have been submitted to MainBoss. For example, @Requests might discover that a previously submitted request has now been transferred to a work order. Therefore, @Requests will move the corresponding message from the **Waiting** list to the **Transferred** list.

**Important:** If you set this field to zero or leave it blank, @Requests will not handle requests automatically. @Requests will only deal with incoming requests when you explicitly tell it to do so. For a description of how to tell @Requests to handle requests, see [Using MainBoss @Requests on page 14](#).

**Clear diagnostics messages when program starts:** Specifies whether or not to erase all diagnostic messages previously written to `diagnostic.txt` in the @Requests database directory every time @Requests starts. Checkmarking this box will prevent diagnostic messages from building up over time; however, if you find you are having problems with @Requests, you should turn off the checkmark on this box so that important diagnostic messages are not erased.

**Title:** Specifies text to be used in the `From:` line of acknowledgement e-mail messages.

**Subject line prefix:** Specifies text to be placed at the beginning of the `Subject:` line in acknowledgement e-mail messages. By default, this is **@Re:**. For example, if someone sends an e-mail message with the subject line

```
Leaky faucet
```

the acknowledgement message will have the subject

```
@Re: Leaky faucet
```

**First line:** Specifies text to be used as the first line in the body of acknowledgement messages. The default is **Current status of your MainBoss @Request**.

**Return e-mail address:** Specifies an e-mail address to be used if people want to reply to acknowledgement messages; this is also the address that is notified in an acknowledgement message cannot be delivered. Typically, you should specify the e-mail address of the person who administers MainBoss @Requests.

Note that this should be a *different* address than the Request Mail Recipient address. For example, if the Request Mail Recipient address is `workreqs`, you might create another e-mail account `workreq_comments` for people who want to send in questions or complaints after receiving an acknowledgement. You don't want such responses going to `workreqs` itself or @Requests will try to turn those responses into new work requests.

If you leave **Return e-mail address:** blank, @Requests does not send out acknowledgement messages.

The window also has buttons that let you proceed to other steps in the configuration process or just to close the window.

## Message Logs

When you double-click on a message in the configuration program's Message List, the program displays a *message log*. This log includes the original message received, plus information on how the message has been processed so far. By examining message logs, you can see what @Requests is doing.

The message log begins with lines giving information about the requestor. For a basic message, this information is taken from the Requestor database, based on the requestor's e-mail address; for a formatted message, this information is taken from the message itself.

The next part of the message log tells what has happened to the request since the original message was received. For example, you might see dates and times for when the request was originally placed into the MainBoss database, when it was transferred to a work order, and so on.

The last part of the message log gives the original message as it was received by @Requests.

### *Printing Message Logs*

When @Requests displays a message log, the display window has a **Print** button that lets you print the message log. To set up the printer used for printing, select **Session** → **Printer Setup** from the configuration program menu. This opens a normal Windows printer setup dialog that lets you specify printing details.

**E-mailing the Requestor:** In some cases, you may want to send e-mail to the person who submitted a particular request; for example, you may wish to ask for further details about a problem or to explain something about the progress of the request. The window that displays the message log for a request has a button labeled **E-mail Requestor**. If you click this button, @Requests will open a window where you can create and send such a message.



## Using MainBoss @Requests

The @Requests program is designed to deal with work requests automatically. However, there may be times when you want to force @Requests to perform a particular action outside its usual schedule.

For example, suppose you have configured @Requests to run once a day. However, someone submits an emergency problem report that should be handled immediately. In this case, you can use the configuration program to tell the program, “Handle all received work requests right away, even if you aren’t scheduled to do so.”

You may also have configured @Requests so that it does not run automatically. (See [@Requests Configuration Preferences on page 11](#) for a description of how to do this.) If @Requests does not run automatically, incoming requests will not be processed until you explicitly tell @Requests to do so.

You control @Requests through the configuration program’s **Actions** menu. Possible actions are:

- Retrieve E-mail:** Checking for incoming messages
- Update Logs:** Processing the incoming messages and checking to see what has happened with previous work requests
- Send Acknowledgements:** Sending acknowledgements to tell requestors about the progress of their requests
- Do All:** Performs all three of the above steps

Typically, you would just select **Do All**—this does everything necessary to handle new and old requests. However, there may be situations in which you want to perform one type of action without performing others. For example, if you want to review incoming messages before actually processing them, you would perform **Retrieve E-mail** so you could see what messages you receive. You could then reject any received requests that you wished to ignore. After you have got rid of such messages, you can proceed with **Update Logs** and **Send Acknowledgements** (in that order).

The sections that follow describe the separate steps of processing requests. Remember that you can use **Do All** to do all the steps in the proper order.

**Important:** If you have configured @Requests to run automatically, all of the actions described in these sections are handled for you. You only have to perform these actions “by hand” if @Requests does not run automatically. For an explanation of running @Requests automatically, see [@Requests Configuration Preferences on page 11](#).

## Reading Request Messages

You can tell @Requests to check for incoming mail messages immediately. To do this, select **Actions** → **Retrieve E-mail** from the configuration program menu.

@Requests will immediately check the Request Mail Receiver account for new e-mail messages. If messages are found from authorized requestors, they will be placed in the **Pending Processing** folder ready for processing. Messages from unauthorized requestors are put into the **Rejected** folder.

Note that this action does *not* actually process the messages and create corresponding work requests. That is done by the next action, updating logs.

### *Rejecting Requests*

After @Requests has retrieved all the incoming messages, you can examine those messages by looking at the **Pending Processing** folder. You can read any message by double-clicking on its entry in the Message List.

If you find a message that you do not wish to service, you can reject that message immediately—this prevents any further processing. To reject a message, follow these steps:

1. Use the mouse to drag the message from the Message List to the **Rejected** icon in the Message Type Area. This moves the message from **Pending Processing** to **Rejected**.
2. Click **Rejected** in the Message Type Area. This shows you the list of rejected messages, including the one you just rejected.
3. Hold down the right-hand mouse button and drag the message you just rejected to the **Pending Acknowledgements** icon in the Message Type Area. @Requests will display a menu giving you the choices **Move here** and **Copy here**. Click on **Copy here**.

The final step above places a note in **Pending Acknowledgements** indicating that the user should be told the request has been rejected. If you skip this step, the user will not be informed of the rejection.

## Updating Logs

The next step in processing incoming messages is called *updating logs*. To do this, select **Actions** → **Update Logs** from the configuration program menu.

For this action, @Requests does the following:

- Creates work requests based on any messages in the **Pending Processing** folder. After a work request has been created, the corresponding message is moved out of **Pending Processing** and into **Waiting**.

- Checks on the status of work requests that have been previously submitted. For example, @Requests determines if any of those work requests have been transferred into work orders. Messages are moved as appropriate; for example, if @Requests finds that a work request has been voided, the work request will be moved into the **Void** folder. Similarly, if a work request was previously voided but has been reactivated, it is moved from **Void** to **Transferred** and marked as “reopened”.
- For every message whose status has changed, @Requests puts a *copy* of the request into the **Pending Acknowledgements** folder. For example, if a message has moved from **Pending Processing** to **Waiting**, @Requests puts a copy of the message into **Pending Acknowledgements**. This indicates that the requestor should be informed that the request has been processed.

Although this action moves messages from one folder to another, it does not send out acknowledgements to tell requestors what has happened. That is done by the next action, sending acknowledgements.

### **Sending Acknowledgements**

The last step in processing incoming messages is to send out acknowledgements. To do this, select **Actions** → **Send Acknowledgements** from the configuration program menu.

For this action, @Requests goes through the **Pending Acknowledgements** folder and sends appropriate e-mail messages to tell requestors about the progress of their requests. After an acknowledgement has been sent, the corresponding message is removed from the **Pending Acknowledgements** folder.

Although messages are deleted from the **Pending Acknowledgements** folder, there will still be copies of those messages in other folders. For example, when @Requests discovers that a work request has been voided, @Requests moves the message to the **Void** folder and places a copy of the message in **Pending Acknowledgements**. Therefore, when the **Pending Acknowledgements** copy is deleted, there will still be a copy in the **Void** folder.

### **Cleaning Up Message Folders**

The configuration program lets you clean up the various message folders used by @Requests. For example, once a message has been rejected and an acknowledgement sent, you probably want to remove that message from the **Rejected** folder—if you keep old messages in that folder, you’ll have a harder time keeping track of recent rejections.

Similarly, you probably don’t want to accumulate a lot of old messages in the **Void** or **Completed** folders; that only creates clutter that makes it harder to keep track of recent developments. The @Requests program will also run more slowly if there are a lot of messages in the **Void** or **Completed** folders.

To remove a message in any Message List, click on the message and then press the right-hand mouse button. This opens a menu of actions that can be performed on the message:

**Save:** Moves the message to the **Saved** folder. This removes the message from the current Message List. For example, if you are looking at the **Completed** message list and **Save** a particular message, the message is removed from the **Completed** list and moved to the **Saved** list.

**Delete:** Moves the message to the **Deleted** folder. Notes that this removes the message from the current Message List.

**Important:** You should not **Save** or **Delete** messages that are **Waiting** or **Transferred**. Doing so may interfere with the normal processing of the work request. For example, if a message is **Waiting**, you should leave the message alone; if you want to get rid of the corresponding work request, use the regular MainBoss program to void the work request. After that, @Requests will move the message to the **Void** list and you can delete the message from that list.

The configuration program lets you move messages from one Message List to another by dragging. For example, if you click on a message in a Message List and drag it to **Saved** in the Message Type Area, the effect is the same as doing a **Save** operation on the message. Similarly, dragging the message to **Deleted** in the Message Type Area has the same effect as doing a **Delete** operation. As noted above, you should not move messages that are **Waiting** or **Transferred**.

If you remove a message from the **Pending Acknowledgements** message list, you prevent @Requests from sending the corresponding acknowledgement to the requestor. For example, if there is a message in **Pending Acknowledgements** telling the requestor that a request has been completed, deleting that message means that the requestor will not receive an acknowledgement about this. In general therefore, you should not save or delete messages from **Pending Acknowledgements** unless you want to prevent someone from receiving an acknowledgement.

**Note:** You should make sure to examine **Rejected** messages on a regular basis to make sure that you are not missing important complaint reports. You may **Save** or **Delete** messages in the **Rejected** folder in the same way that you do for any other folder.

## The @Requests Command Line

The @Requests program may be invoked from a command line (for example, in a .bat or .cmd file). To do this in a Command Prompt window, you typically CD to the directory that contains the @Requests program, then invoke the program (named atrserv.exe). Here is a typical example:

```
cd "C:\Program Files\MainBoss\@Requests\1.1"  
atserv /mbd:C:\MainBoss
```

There are two recognized command line options:

/MainBossDatabase:folder

specifies the folder that contains your MainBoss database. If you omit this option, MainBoss assumes that the current directory contains your database.

/LANGID:xxxx

Sets the language to the one represented by code xxxx. For example, /LANGID:0007 sets the language to basic German. (The code numbers used are the standard Microsoft identifiers; for more information, see your Windows documentation.)

As a special case,

```
atserv /?
```

prints information about using MainBoss in command-line format, including a list of all subcommands and options.

**Abbreviations:** Option keywords on the MainBoss command line may be abbreviated. In MainBoss documentation, the 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 /MainBossDatabase:

```
/mainbosddb  
/mbdata  
/mbdb  
/mbd
```

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

## INDEX

- @Requests, 2
- abbreviations, 18
- account name, 11
- acknowledgements, 3, 16
- command line, 18
- contacts table, 10
- database
  - requestor, 2
- diagnostic file, 11, 12
- e-mail, 2
- everybody, 9
- first line in acknowledgements, 12
- installation
  - MainBoss @Requests, 5
- license keys, 6
- MainBoss @Requests, 2
  - configuration, 7
  - installation, 5
  - preferences, 11
- message logs, 13
- password, 11
- pending acknowledgements, 8, 16
- pending processing, 8, 15, 16
- POP3, 5
- POP3 server, 11
- preferences
  - MainBoss @Requests, 11
- preview area, 7
- receive acknowledgements, 10
- rejected, 8
- request mail receiver, 2, 11
- requestor database, 2, 9, 13
- requestors, 9
- return e-mail address of
  - acknowledgements, 12
- sending acknowledgements, 16
- SMTP, 5
- SMTP server, 11
- subject line, 12
- title, 12
- update logs, 15
- work requests, 2