Migrating from MainBoss Basic
to MainBoss Advanced

Table of Contents

Introduction
Welcome to MainBoss Advanced

This guide introduces MainBoss Advanced to people who are already familiar with MainBoss Basic. The goal is to highlight differences between the two, and to help sites upgrade from Basic to Advanced.

The Essential Difference: SQL Server

The most important difference between MainBoss Basic and MainBoss Advanced will be invisible in day-to-day use: the two versions of MainBoss store data in different ways.

MainBoss Basic uses a database that is good for small installations but less efficient for larger ones. On the other hand, MainBoss Advanced uses Microsoft's SQL Server. This is actually a family of compatible database products: whether your site is large or small, there's a version of SQL Server that will work well for you. By using SQL Server, MainBoss Advanced can appeal to the widest possible range of customers.

SQL Server has one other advantage: familiarity to people in the computing industry. The computer departments in large corporations prefer to work with known software products. SQL Server is something they already understand. For example, IT personnel know how to back up the contents of SQL Server databases and how to restore data if necessary. Using SQL Server makes life easier for those who keep computer systems running.

SQL Server is therefore a smart choice to use with the next generation of MainBoss. It's efficient, widely-used, and up to date.

> **Note:** SQL Server is designed for a network environment that uses domains. If your organization doesn't use domains, and you don't need the extra functionality of MainBoss Advanced, we recommend that you stay with MainBoss Basic.

Web Capabilities

MainBoss Advanced has two modules that let you use MainBoss on the Web (Internet):

☐ **Web Requests:** Lets authorized users submit work requests through a web page. Such requests are automatically placed into the MainBoss database. MainBoss can also send automatic e-mail acknowledgements to submitters at various points of request processing (e.g. when a work order is created from the request, and when the request is closed).

☐ **Web Access**: Lets workers use the web to access assigned work orders. Not only can they read the work orders, they can record actual material and labor usage once the job is done.

These two modules can be used with any device that has web capabilities. In particular, they can be used through web-enabled handheld devices (e.g. iPhone and BlackBerry devices).

> For more information on the web capabilities of MainBoss Advanced, see the *Web Operations* guide.

Redesign

MainBoss also has a new look. While we were rewriting the software to accommodate SQL Server, we made many design changes in response to requests from users. You'll see a number of differences in MainBoss's look and feel, but more importantly, important changes have been made in less visible parts of the software. The initial version of MainBoss Advanced has most of the same functionality as MainBoss Basic, but our changes open the door to larger enhancements in future releases.

Our new look is based on the following principles:

☐ *Bigger monitor screens:* MainBoss Basic dates back to a time when many computers had small display screens. These days, a typical screen is big enough to display a great deal more information; therefore, MainBoss Advanced takes advantage of the extra screen size, so you can see more at a single glance.

   We strongly recommend that you run MainBoss Advanced with the window maximized to the full size of your monitor screen.

☐ *More processing power:* Modern computers have more processing power than they used to. Again, MainBoss Advanced takes advantage of that power to do more for you.

> Here's an example of what bigger screens and more processing power make possible. If you look at the table of work order closing codes and if you click on a particular code, the table viewer can immediately provide you with a list of work orders that use that code. A similar principle applies to many other tables—if you look at a particular record, MainBoss can list where that record is used.

☐ *Preparing for future functionality:* Many features of MainBoss Advanced are first steps toward enhancements. For example, the accounting features were designed with the intention that MainBoss will someday be able to interface directly with accounting software packages. Such features are not yet implemented, but the groundwork has been laid.

☐ *Responding to customer feedback:* Customers frequently give us good suggestions for what they want MainBoss to do. MainBoss Advanced incorporates many of those suggestions, such as:

   ☐ The ability to make windows bigger or smaller, to have several windows open at once, to create entries as needed (i.e. while you're creating one record, creating some entry you need in another table)

   ☐ Submission of requests via the web

   ☐ Web access to work orders

   ☐ Better audit trails and support for future accounting functionality

   ☐ More versatile security facilities

   ☐ Simpler ways to find requests, work orders, and purchase orders that are directly relevant to you (especially work orders to which you've been assigned)

   ☐ The use of calendars to specify dates rather than entering dates as text

   ☐ Better handling of situations where workers are paid extra for overtime

   ☐ Handling for situations where workers or outside contractors are paid by the job rather than by the hour (e.g. where a mechanic takes a flat rate for an oil change, no matter how long it takes)

   ☐ The ability to create *boilerplate* work orders (e.g. you write up a standard "oil change" work order that you can use to quickly create an oil change work order for any vehicle)

   ☐ The separation of chargeback costs from real costs. For example, a property management company may charge tenants for labor at a different rate than the workers actually receive. MainBoss Advanced lets you record chargebacks at one rate and real costs at a different rate, if appropriate.

   ☐ The ability to assign temporary storage locations to work orders. For example, if you're doing a large renovation, you may want to store materials close to the work site rather than in a regular storeroom. By designating a temporary storage location for the job, you can still keep track of where all your materials are, so that you don't waste time looking for them.

   ☐ A more useful structure for dealing with locations, sub locations, units and sub-units.

   ☐ Flexible ways to select what is shown when viewing tables

---

We strongly recommend that you read the *Getting Started* guide to learn the fundamentals of MainBoss Advanced. The rest of this document assumes that you've already read that guide.

Importing Data from MainBoss Basic

If you have an existing installation of MainBoss Basic, you will probably want to transfer information to MainBoss Advanced. This is called *importing* data from your existing MainBoss Basic configuration.

The importing process does *not* obtain all information from your database. It imports *definitions* (like information about buildings, tasks, personnel, etc.) but not historical information about day-to-day operations. Roughly speaking, the importing process obtains information for the **Coding Definitions** part of MainBoss Advanced, including unit information and set-ups for planned maintenance. The process does *not* import existing work orders or requests.

> **Important: Before you begin this process, you must install the most up-to-date version of MainBoss Basic. This is version 2.9, update 9. You can obtain this software from the MainBoss web site.** (The software is also available on CD, but we recommend that you go to the web site, since it will always have the most up-to-date version.)

Preparation for Migration

Before you start the export/import process, it's important to have a clean database: one with no oddities that might interfere with migration. For example, unrealistic dates may interfere with the migration process (e.g. a work order whose effective closing date has accidentally been set far into the past or future). The more inappropriate records you can eliminate, the fewer problems you'll run into during migration.

To clean up your database, follow the instructions given in:

http://www.mainboss.com/english/support/faq/mb28_cleanup.shtml

This explains how to use the MainBoss auditing facilities to remove anomalies from your database. **You can audit and clean your database far in advance of the actual migration process. This reduces the work that you have to do when you actually begin migration.**

When the time actually comes for migration, it's time for one last clean-up:

1. Make sure you've installed the most recent update of MainBoss 2.9—version 2.9, update 9.

2. Do one last audit to make sure the database is clean (**Administration** → **Database** → **Audit**). Correct any problems reported.

3. Make sure that no one else uses the database over the course of the next operations.

4. Reindex the database (**Administration** → **Database** → **Re-Index**).

5. Backup the database so that you can recover if anything goes wrong (**Administration** → **Database** → **Backup**).

6. Follow the instructions in

   http://www.mainboss.com/english/support/faq/mb28_dbcopy.shtml

   to make a copy of your database—a test database—and use this test database in all subsequent parts of the migration.

By making a test database, you have the security of knowing that if you make mistakes during the migration process, your original database is still intact and usable. Also, if some data is lost during migration, you can consult your original database and fill in the data by hand.

The Migration Process

You can transfer information from MainBoss Basic to MainBoss Advanced by following these steps.

1. Start MainBoss Basic (on the *test* database that you made in the last section).

2. Login as Administrator (using **Login** from the **Session** menu).

3. Select **Administration** → **Database** → **Re-Index**. This tidies up your database.

4. Select **Administration** → **Database** → **Export XML**. This lets you export data from your existing database to a file. You'll see a window like this:
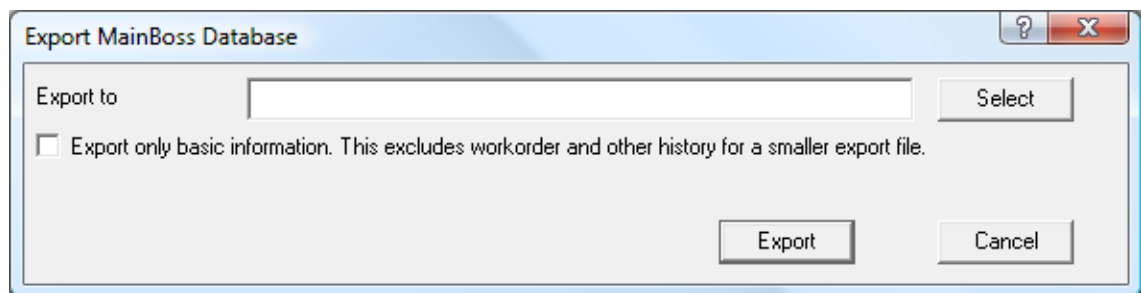
5. In "**Export to**", specify the name of a new file where MainBoss can save information from your existing database.

6. Checkmark **Export only basic information.** This prevents a great many headaches that can arise in the export-import process. It also reduces the size of the export file and greatly decreases the time required in the import process.

> MainBoss Advanced only imports basic information, even if you export more. Whether or not you checkmark **Export only basic information**, MainBoss Advanced imports the same subset of your original data.

7. Click **Export**. MainBoss exports your existing data.

8. Quit MainBoss Basic.

9. If you aren't already logged in to a Windows account with SQL Server Administrator privileges, log in to such an account now.

10. Start MainBoss Advanced.

11. In the **Session** menu, select **Change Maintenance Organization**. MainBoss opens a window showing your current maintenance organizations.
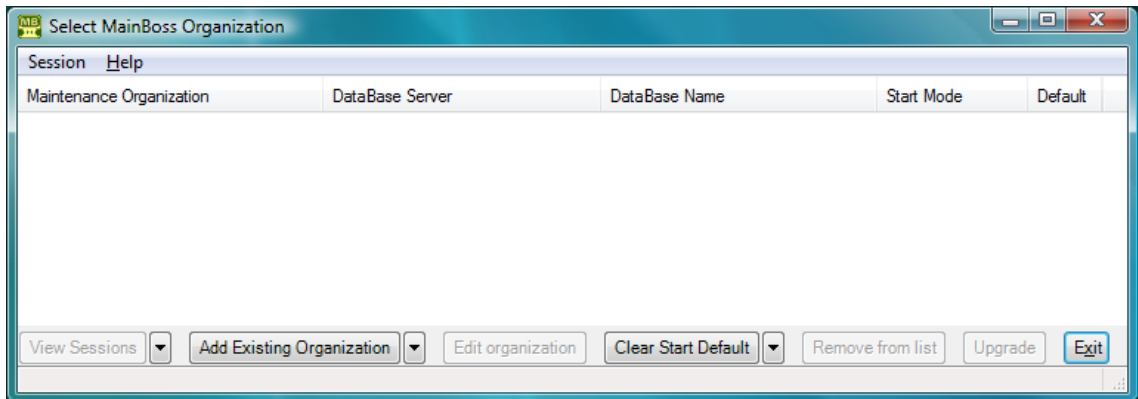
12. Drop the arrow on **Add Existing Organization**, and click **Import MainBoss 2.9 Organization**. MainBoss opens the following window:
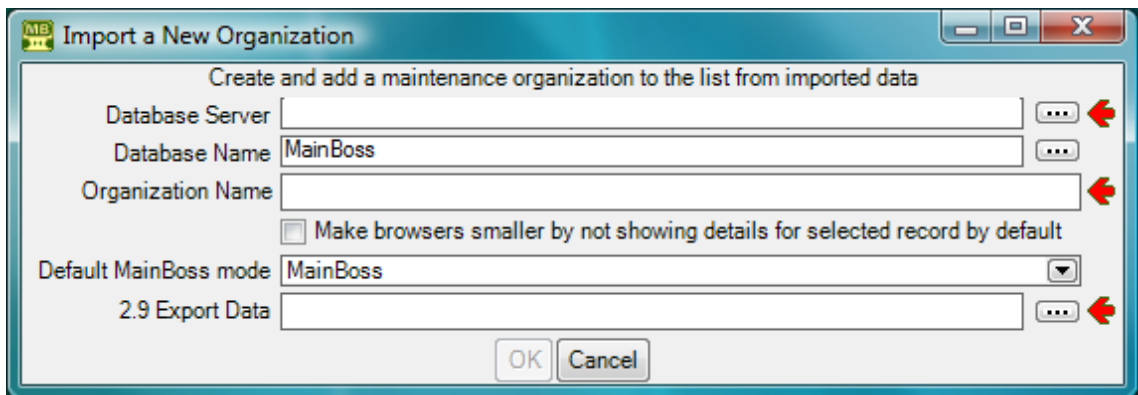
13. In "**Database Server**", specify the name of the SQL Server instance that you will use.

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

14. "**Database Name**" is set to MainBoss. Keep this name, unless you already have a SQL Server database named MainBoss. If you need to specify a new name, the

name should only consist of letters and numbers, with no spaces or other punctuation characters.

> If you click the "..." button after "**Database Name**", MainBoss will list any MainBoss databases that are already maintained by "**Database Server**". This may make it easier to choose a name that isn't already being used.

15. In "**Organization Name**", enter the name of your company. For further details, see the online help associated with **Administration** | **Company Information**.

16. In "**2.9 Export Data**", specify the name of the file where you stored exported data from MainBoss 2.9.

17. Click **OK**. MainBoss creates a new database containing the imported data.

> As the importing process takes place, MainBoss displays messages tracking the progress of what's happening. Depending on the size of your database and the speed of your computer, the importing process may take a *long* time (possibly hours).

18. When the database has been created, you'll return to the list of your existing maintenance organization databases. Click the name of the one you just created, then click **Open Organization**.

19. The new database will need license keys—license keys from MainBoss 2.9 do *not* work with MainBoss Advanced. Follow the steps given in the *Installation and Administration Guide* to record the license keys you've been given by your dealer or by Thinkage.

_____

Errors During the Import Process

If errors occur during an import process, MainBoss displays one or more messages about what went wrong. To see what the errors were, go to **Administration** | **Database History** which will contain a list of error messages. You'll be offered the choice whether to keep the imported database (despite the errors) or to discard it. In many cases, you should take note of the errors and then discard the database.

Usually, errors occur because:

☐ You didn't use the most up-to-date version of MainBoss 2.9. Make sure you have update 9 or later. (To see what version you're using, check **Help** → **About**.)

☐ Your database contains errors of some kind. You'll have to correct the errors, then go through the migration process again. For tips on correcting errors and keeping MainBoss 2.9 databases clean, see

http://www.mainboss.com/english/support/faq/mb28_cleanup.shtml

> **Note:** When MainBoss encounters an error in the export/import process, it reports the location of the error in the XML file that contains the exported data. This XML file can be edited with any standard text editor (e.g. Notepad) and knowledgeable users may find this the easiest way to correct simple errors. However, MainBoss Support will not deal with any errors that might arise from using an XML file that has been edited by hand.

Optimizing and Improving Your Configuration

MainBoss Advanced has features that are not available in MainBoss Basic, most importantly sub locations and sub units. MainBoss Basic couldn't record which locations were contained by other locations, nor could it subdivide units into sub units. After the importing process, you should look at the organization of your **Locations** and **Units** tables to see if they should be revised.

As another example, MainBoss Basic let you establish a list of work order priorities, but there was no way to specify which priority classes were most important. MainBoss Advanced lets you assign a ranking to each priority class.

> The import process assumes that your Basic priority table listed priorities from most important to least, as sorted in alphabetical order (i.e. the way they're listed in MainBoss Basic). You should review the work order priority list after the import process to make sure the priorities have been ranked in the way you wish.

In general, you should review *all* your tables after they're imported to see if the information should be reorganized in some ways. In the short term, you may decide not to make any changes, but in the long term, you should ask yourself how to reorganize your data to get the most out of MainBoss Advanced capabilities.

Authorizing Users

MainBoss Basic looked after its own security. Many organizations didn't require MainBoss users to enter a password—anyone could use the software. Other organizations created special user names and passwords within MainBoss that didn't match Windows login names and passwords.

The security in MainBoss Advanced is based entirely on Windows login names. When you first create or import a database, the only person authorized to use that database is the person who created it. At that point, only the creator can authorize other people to use MainBoss.

In order to use MainBoss Advanced, you must be explicitly authorized in the MainBoss Users list. Typically, after importing a database, the person who does the importing then uses **Coding Definitions** | **Users** to record all the people who will be authorized to use that database.

For more information on authorizing people to use a database, see the *Installation and Administration Guide* or the online help for **Coding Definitions** | **Users**.

Security Roles

MainBoss Basic handled security on a feature-by-feature basis. For each user, you could specify which program features the user could or could not use.

With MainBoss Advanced, security is handled by assigning each user one or more *security roles*. A security role grants a user a set of related permissions. For example, the `Requests` role grants a user the ability to use all program features in the **Requests** section of the control panel. The `RequestsView` role is similar, but just grants read-only permission on request data; in other words, the user can look at requests but not modify them.

Important: Since certain fields are not displayed unless you have appropriate permissions, the windows you see when using MainBoss may be different than the pictures shown in this guide.

MainBoss users will often need multiple security roles in order to do their jobs. The *Quick Configuration Guide* suggests which roles are appropriate for various purposes. (The person who creates a MainBoss database is automatically given a set of roles that allows the person to use all of MainBoss's features.)

There are several types of security roles:

☐ *Summary* roles: Let you read a subset of the information that a record contains. For example, the `UnitSummary` role lets you read basic information about a unit but doesn't let you to read all the information in a unit record.

Summary roles are intended for people whose primary concern is some other aspect of MainBoss but who may have need for a small amount of information on other subjects. For example, someone working on a help-desk is primarily concerned with requests, but might also be given `WorkOrderSummary` (in order to tell clients, "Yes, your job is on our schedule for next Thursday" and to link requests to existing work orders) and `UnitSummary` (in order to make sure that the unit specified in a request really is the unit that needs service).

We recommend that organizations be generous in assigning *Summary* roles, but cautious in handing out roles that allow wider access to information.

☐ *View* roles: Let you read information but not change it. For example, the `UnitView` role lets you read all the information in a unit record, but doesn't let you change any of that information.

A view role includes all the information available through the corresponding summary role. For example, `WorkOrderView` includes all the information of `WorkOrderSummary` (and more). Therefore, if you assign someone a view role, you don't need to assign the corresponding summary role.

☐ *Fulfillment* roles: Let you perform selected operations on a type of record. Loosely speaking, these operations are ones you'd do in day-to-day work *except* for actually creating the record. For example, `WorkOrderFulfillment` doesn't let you create work orders, but lets you perform operations related to closing existing work orders. `RequestFulfillment` lets you add comments to requests and to send comments to requestors (if you have licensed the **MainBoss Service** module). `ItemFulfillment` doesn't let you create item records, but lets you record item information including physical counts, item issues, and item transfers.

With `WorkOrderFulfillment`, you can only affect work orders that you can already see because of other permissions. In particular, if you don't have `WorkOrder` or `WorkOrderView`, you will only be allowed to deal with work orders to which you've been assigned. A similar principle applies to `RequestFulfillment` and `PurchaseOrderReceive`.

☐ Create/edit roles: Let you create and edit various types of records. For example, the `WorkOrder` role lets you create/edit work orders while the `Item` role lets you create/edit item records.

A create/edit role includes all the information available through roles in the same "family". For example, if you assign someone `WorkOrder`, you don't have to assign that person `WorkOrderSummary`, `WorkOrderView` or `WorkOrderFulfillment`, since `WorkOrder` includes all the permissions of the other three roles (and more). Therefore, if you assign someone a create/edit role, you don't need to assign any other roles in the same "family".

☐ *Accounting* roles: Provide access to monetary information. If you do not have an appropriate accounting role, you may be prevented from seeing prices and costs; for example, if you have `WorkOrder` role but not `AccountingWorkOrder`, you will be able to record, say, the quantities of materials used on a job, but you will not see the actual cost of those materials.

In addition to the selective roles listed above, there is a role called `All`. This grants a user permission to use every aspect of the program. We recommend that you avoid using `All`; too often, we have seen organizations give `All` permission to users without thinking about it. Instead of taking the shortcut of `All`, decide which permissions a particular user really needs and only give the user those permissions.

Security roles are listed in the **Security Roles** table (**Administration** | **Security Roles**). To give someone one or more security roles, you open that person's record from the **Users** table (**Administration** | **Users**) and go to the record's **Security Roles** section. You then use **New User Security Role** to add role records to the user's list of roles.

> Each security role record contains a "**Comments**" field explaining what permissions the role provides.

Some security roles combine with each other to provide users with more information. For example, if you have both ItemSummary and WorkOrderView, you can see more information about the items on a work order than either ItemSummary or WorkOrderView would provide individually.

> When you assign security roles to a user, it can be difficult to figure out the effect of those roles: what the user can and can't do with the roles you've assigned. To make it easier to see the effects of a person's security roles, you can use the **Evaluate Security As** button in **Administration** | **Users**. For more information, see the online help.

Additional Importing Notes

Here are some more notes on importing:

☐ MainBoss Basic let you have non-inventory items with the same names as inventory items. Since MainBoss Advanced doesn't distinguish between inventory and non-inventory items, all items must have unique names. Therefore, the importing process adds (NI) to the name of non-inventory items if they have the same name as inventory items. A non-inventory item named X would therefore become X (NI) (but only if there was also an inventory item named X).

☐ Company information is not retained by the importing process. Therefore, you will have to set the company location and logo in MainBoss Advanced after the import.

☐ Some databases have negative quantities of some inventory items, or quantities of zero on items which have a non-zero value, e.g. you have zero light bulbs on hand, but the value of your light-bulb stock is recorded as non-zero. (This can happen if you make mistakes in recording inventory quantities.) In such cases, the import/export process automatically sets both the quantity and the value of the item(s) to zero.

If MainBoss Advanced adjusts quantity and/or value in this way, it creates an adjustment record with a special adjustment code. The adjustment codes used for this purpose are then deleted from your MainBoss Advanced adjustment code tables, so that they don't take up unwanted space. However, since MainBoss Advanced has an option that lets you see deleted records, you may see the deleted codes if you use this option.

Overview of Changes

This section provides details on specific differences between MainBoss Basic and MainBoss Advanced.
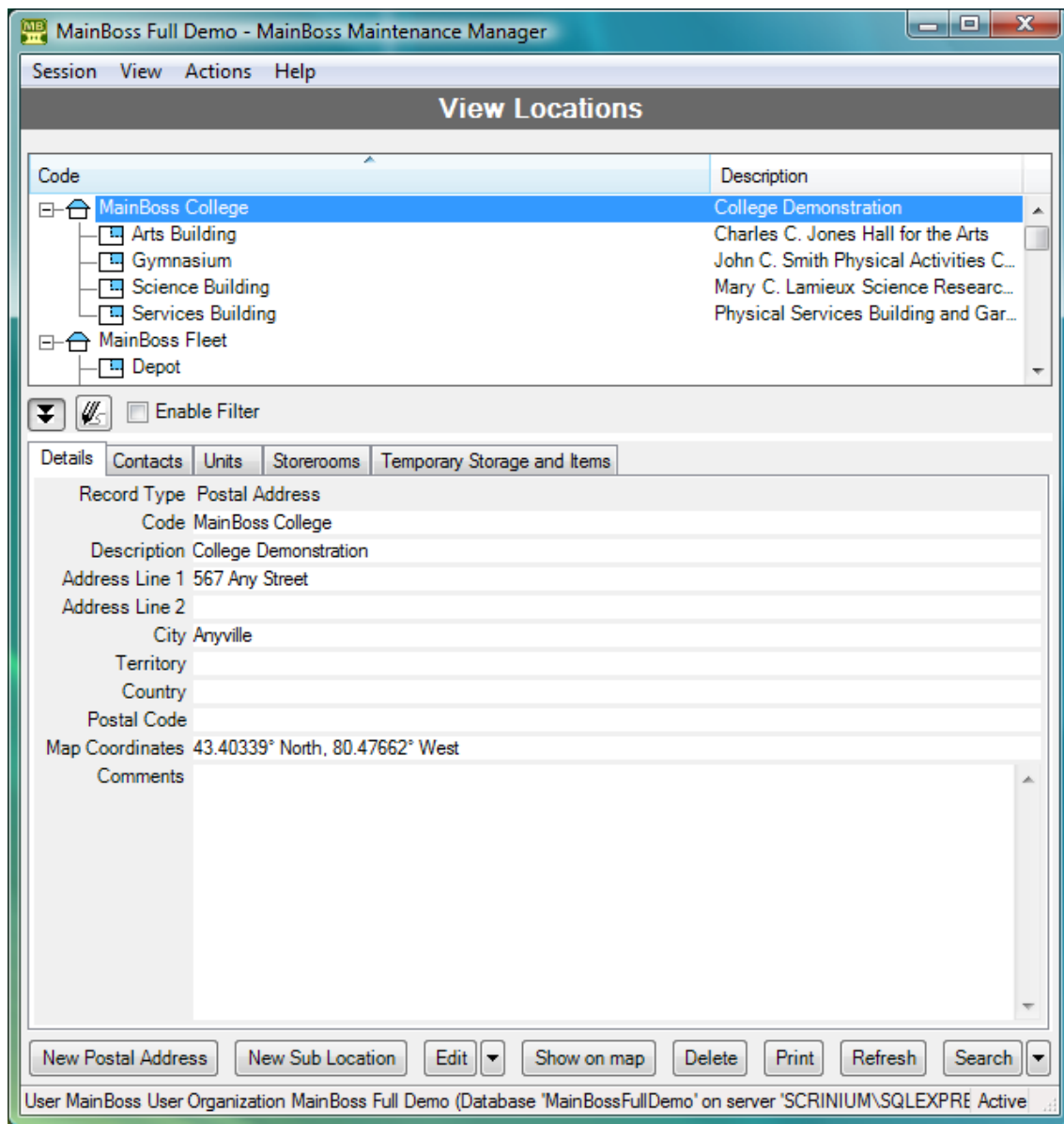
Locations

MainBoss Advanced uses *locations* as a substitute for several related ideas in MainBoss Basic. Locations are controlled through **Locations** in the control panel.

The **Locations** table stores *all* the locations used in your work. For example, your buildings (from MainBoss Basic) are now locations in MainBoss Advanced. You use the **Locations** table to record all other locations too, e.g. for personnel, vendors, and other contacts.

Postal Addresses: The simplest type of location is a postal address. This type of record contains all the usual address information.

Sub Locations: A sub location is a location inside another location. For example, suppose you have a postal address record representing an office building. You might create sub location records for each floor inside that building. You might then create more sub locations for office suites on each floor; these records are *inside* the individual floor records. You could also create records for each room inside each office suite, and so on.

The **Locations** table shows which locations contain which sub locations. The display format is similar to the way the Windows Explorer shows folders, subfolders, and files. This is called a *location map*:

Location records are used extensively throughout MainBoss Advanced. Here are some examples:

☐ A contact record can have an associated location record giving the person's address information. The same goes for personnel records.

☐ A unit record has a location record telling where the unit is.

☐ An item record can have multiple locations associated with it, indicating places where the item is stored.

☐ The **Storerooms** table tells which units and/or locations are storerooms. Since units can contain units and locations can contain sub locations, you can have "sub-

storerooms" inside a storeroom. In fact, you might consider each shelf within a storeroom to be a sub-storeroom; this makes it easy to specify the precise location of parts within a storeroom.

Imported Locations: When you import data from MainBoss Basic to MainBoss Advanced, you'll see the following:

☐ Entries in your **Buildings** table become postal addresses.

☐ There are no sub locations.

☐ Entries from your **Spaces** table become units. They are marked with Space in the "**Usage**" field.

☐ Entries in your **Equipment** table also become units. They are marked with Equipment in the "**Usage**" field.

☐ Entries in your **Storerooms** table become storerooms.

> **Reorganization:** We strongly recommend that you examine the **Locations** table after the import process and consider reorganizing to use the new features. For example, instead of one long list of rooms (spaces), it's useful to break up the list by floors (as shown in the preceding picture) to make it faster and easier for you to find the entry for a specific place.

In general, location and sub location records should only be used for large-scale entities (buildings and groups of buildings). Anything smaller should be a unit.

Contacts

The **Contacts** table contains contact information for vendors, personnel, work requestors, and anyone else whose contact information you want to record.

> The **Locations** table contains street addresses and geographical information about people and organizations. The **Contacts** table contains phone numbers, e-mail addresses, and other contact information.

Contacts may be associated with multiple locations. For example, a contact might have multiple postal addresses (at work, at home, etc.). Similarly, a location may have multiple contacts associated with it (e.g. contact information for several people who all work in the same building).

If you are creating/editing a contact record, you can create new locations for that contact. You can also indicate that the contact is associated with an existing location.
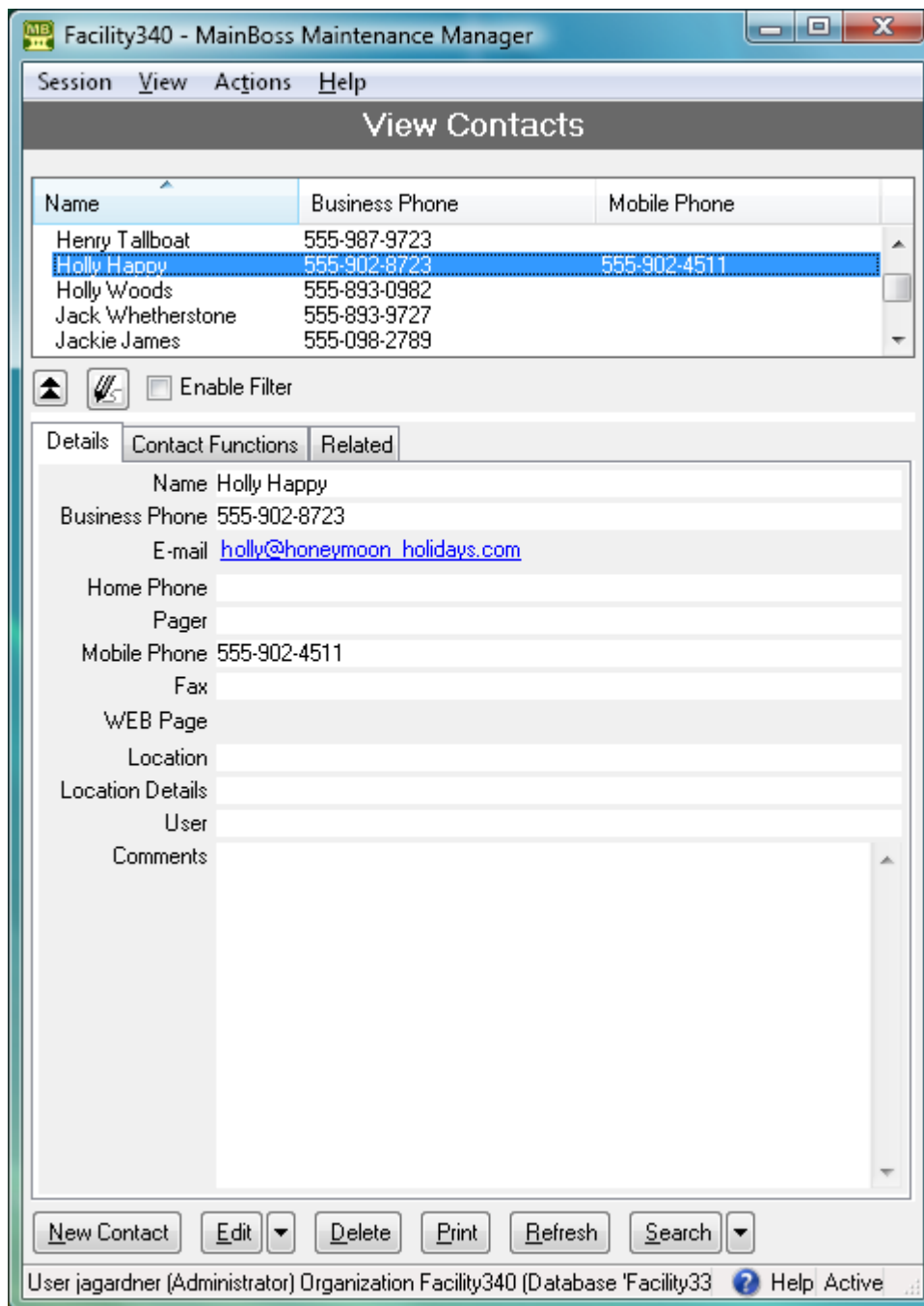
As shown above, the bottom part of the **Contacts** window has three sections:

☐ **Details** lists the contact information in the record.

☐ **Contact Functions** lists tables where the selected person appears (e.g. requestors, employees, vendors, etc.).

☐ **Units** lists units (if any) for which the selected person is the designated contact.

Imported Contacts: After importing information from MainBoss Basic, the **Contacts** table will contain contact information from your old **Contacts**, **Personnel** and **Vendors** tables.

The **Vendors** table now allows multiple contacts to be associated with the same vendor: a sales contact, a service contact, and an accounting contact. You should fill in this information as appropriate.

Accounting Facilities

While MainBoss Basic kept track of costs, it allowed monetary values to be changed retroactively. MainBoss Advanced is designed to keep stricter accounting records, including a complete history of all actions that may affect your company's ledgers.

For example, whenever you save a record which contains a monetary value that may affect your company's ledgers, MainBoss creates a permanent accounting record associated with that value. If you wish to change the value later on, you can't directly edit it in the original record; instead, you must make a *correction* record (which also creates a permanent accounting record). In this way, MainBoss Advanced keeps a rigorous history of all your transactions.

Furthermore, MainBoss Advanced pays attention to a number of possible accounting transactions that MainBoss Basic ignored. For example, in some organizations, transferring materials from one storeroom to another may mean transferring the materials' value from one ledger account to another. MainBoss Basic ignored this; however, MainBoss Advanced can generate appropriate accounting records for the transfer (when appropriate).

> **Note:** MainBoss is not a full-fledged accounting package. However, it does keep track of the transactions needed by such a package. This information can be exported for use with an accounting package. However, there is currently no direct method for using these records in conjunction with any accounting software package. You will have to export the information from MainBoss and process it into a format acceptable to your accounting package.

Because MainBoss Accounting can be more thorough, it may take more work to set up than the accounting facilities in MainBoss Basic. However, the set-up procedure begins the same way: **Create Cost Center records corresponding to all the general ledger accounts that are relevant to your maintenance activities.** You set up cost centers with **Coding Definitions** | **Cost Centers**. For further information, see the *Configuration Guide*.

> In order to use cost centers, you must have licensed the MainBoss **Accounting** module. If you do not have an appropriate license, you will not see cost centers in any MainBoss window.

You can have multiple cost centers associated with the same general ledger account. You might do this if you want to track more detailed financial information than your ledger accounts allow. For example, suppose you have a single general ledger account for paying outside contractors. However, you want to separate the amounts you pay to plumbers from the amounts you pay to other contractors. You could create two separate cost centers (`Plumbers` and `Other`) which are both associated with the outside contractor ledger account. In this way, MainBoss assigns the costs to the correct general ledger account, but also gives you a way to separate the costs by cost center.

Once you've set up your cost centers, there are two other important types of records to set up:

☐ *Expense Models:* An expense model is a type of MainBoss record that describes how you're going to "cost out" a particular work order. For example, on jobs that you do for Client X, you might be required to categorize expenses in one way; on jobs for Client Y, you may have to use a different system of classification. As another example, organizations that receive government grants may have to write up expenses in different ways on grant projects than on other types of work.

When you create a work order, you choose the appropriate expense model for the job. You should have a separate expense model for all the different expense-reporting formats that you may have to use.

☐ *Expense Categories:* Each expense model has a set of associated expense categories. These are the categories that must be used to classify expenses in this particular situation. You might have categories like

```
Spare parts
Work materials
Labor from outside contractors
```

When you prepare a work order, you look at each expense and ask, "What *type* of expense is this?" You choose the expense category from the list of available categories associated with the expense model.

If you have licensed the **Accounting** module, each expense category within an expense model has an associated cost center. This indicates the general ledger account that should be used for that expense category within that expense model. Linking an expense category to a cost center is called an *expense mapping*.

Expense categories may be restricted to certain uses. For example, you might specify that an expense category can be used for labor expenses but not for inventory items or for miscellaneous expenses. This prevents expense categories being used for inappropriate entries.

As an example, suppose that a property management company creates two separate general ledger accounts for each tenant: one for labor expenses and one for material expenses. This might lead you to set up the following cost centers:

```
    Cost Centers        Ledger Account
TenantA Labor               00001
TenantA Materials           00002
TenantB Labor               00003
TenantB Materials           00004
TenantC Labor               00005
TenantC Materials           00006
```

For each tenant, you create an expense model: `TenantA Model`, `TenantB Model`, `TenantC Model`. The `TenantA Model` would contain the following expense categories:

```
    Category            Associated Cost Center
     Labor                   TenantA Labor
     Materials               TenantB Materials
```

`TenantB Model` and `TenantC Model` would be similar.

Whenever you create a work order, you have to assign it an expense model. Suppose you're doing a job for Tenant A. You look at your list of expense models and see that there is one especially associated with Tenant A; you therefore assign that expense model to the work order.

Next you have to assign an expense category to each expense on the work order. When you look at the list of categories associated with `TenantA Model`, you see there are only two: `Labor` and `Materials`. Therefore, you just have to assign one of those two categories to each expense on the work order. If you have the **Accounting** module, this also associates a cost center with each expense, and therefore a general ledger account. Note, however, that when you fill out the work order, you only have to make two types of decisions:

☐ Which model do we use (what general approach will we use to "cost out" this job)?

☐ For each expense on the work order, which available expense category does it belong to?

Default Accounts: The MainBoss accounting facilities let you separate out expenses in a sophisticated way (e.g. if you keep separate accounts for each tenant). If your needs are more simple, MainBoss has default expense models and expense categories that you can use, with or without modification.

If you don't need accounting at all, just use the default expense model for all work orders. Choose the default expense categories for every item on the work order.

Whenever you create a work order, you *must* specify an expense model. If you haven't set up your own expense models, use the default model.

Similarly, whenever you add an item to a work order, you *must* specify an expense category. If you haven't set up your own categories, use the default ones.

Accounting Records: Every accounting record describes a movement of money from one cost center to another. For example, when you purchase an item with a purchase order, the accounting record describes a movement of money from the Accounts Payable cost center associated with the vendor to the cost center associated with the item's storeroom assignment. As another example, when you actualize an item on a work order, the accounting record describes a movement of money from the cost center associated with the item's storeroom assignment to the cost center associated with the expense category of the item on the work order.

To display the accounting records that MainBoss creates, you use **Administration** | **Accounting**.

Select for Printing

MainBoss Advanced has a **Select for Printing** feature on requests, work orders, and purchase order. Unlike MainBoss Basic, however, MainBoss Advanced does *not* automatically clear the **Select for Printing** checkbox after something is printed. Instead, there is a **Clear Select for Printing** button on the window that does the print-out. When you click this button on the window that prints work orders, it clears the **Select for Printing** checkbox on *all* work orders. Requests and purchase orders work similarly.

Assignments

As a new feature, MainBoss Advanced introduces the concept of assigning people to requests, work orders, and/or purchase orders. Someone assigned to a record is called an *assignee*.

> The purpose of assigning someone to a record is to draw that record to the person's attention. Assignment also makes it easier to find assigned records.

The most important use of assignment is to make it easy for workers to find the work orders for their jobs. When workers start up MainBoss, **Assigned to current user** | **Open Work Orders** lists all open work orders to which they've been assigned. In fact, if workers aren't given any security roles, the only thing they can do in MainBoss is view their assigned requests, work orders and/or purchase orders.

Typically then, when you create a work order, you'll go to the **Assignments** section and create assignments for every worker who'll work on the job. You can also assign the work order to shift supervisors and anyone else who might have a direct interest in the job. Requests and purchase orders follow the same principle: assign them to anyone who has a direct interest.

Assignee Tables: In order for someone to be assigned to a record, that person must first be in the **Contacts** table. That person must also be authorized to be an assignee by being in one of the three **Assignee** tables:

- ☐ **Requests** | **Assignees** or **Coding Definitions** | **Requests** | **Assignees**

- ☐ **Work Orders** | **Assignees** or **Coding Definitions** | **Work Orders** | **Assignees**

- ☐ **Purchase Orders** | **Assignees** or **Coding Definitions** | **Purchase Orders** | **Assignees**

Notifications: When you create a record in an assignee table, the record has a checkbox labeled **Receive Notifications**. If this is checkmarked, and if you have licensed the **MainBoss Service** module, MainBoss sends e-mail to the assignee each time a history record is created for any request, work order or purchase order to which the person is assigned. For example, work order assignees will receive e-mail when a work order is opened or closed, or when someone adds a comment to the work order using **Add Work Order Comment**. In this way, the assignee is automatically notified of significant changes in the work order or its status. The same principle applies to requests and purchase orders.

Status Codes

Requests, work orders and purchase orders may be assigned *status codes*. These codes are user-defined; you can set up whatever codes might be useful to your organization. For example, the following may be useful codes for work orders in some organizations:

```
Waiting for management approval
Waiting for parts
Waiting for customer sign-off
Waiting for government inspection
Temporarily suspended at customer request
```

While organizations can use status codes for any purpose, we recommend that they only be used when a request, work order, or purchase order is in some state that demands attention. A status should mean, "Somebody needs to do something about this." For example, `Waiting for management approval` means that some manager should look at the work order and deal with it somehow. If a work order is proceeding in normal fashion, its status should be blank.

You change the status of a request, work order, or purchase order using **New Requestor Comment**, **Add Work Order Comment**, or **Add Purchase Order Comment**. All three of these open a window where you can specify a status code or remove an existing one.

Note that status codes can be a useful form of communication between users. For example, suppose that a worker has the `PurchaseOrderReceive` security role (meaning that the worker can record the receipt of purchase order line items) but does

not have the `PurchaseOrderClose` security role (which is needed to close a purchase order). The worker could give the purchase order a status like `Ready to Close`, indicating that all appropriate information has been recorded. A supervisor with `PurchaseOrderClose` permission could regularly check all open purchase orders to see if any are `Ready to Close`; if so, the supervisor can review those purchase orders and close them as appropriate (removing the status code in the process).

Remember that a status code is intended to bring a purchase order to someone's attention. You can therefore use status codes as signals for other users.

The Active Filter

The *active filter* determines what counts as `Active` in the request, work order and purchase order tables. In other words, it determines what a table will display when the word `Active` appears in the table's bottom right-hand corner.

Requests, work orders, and purchase orders are always considered active until they are closed or voided. For example, if a work order is open, its age doesn't matter—it will be active until you close it.

To set your active filter, go to the **View** menu and click **Change Active Filter**. This opens a window that contains the following:

Show only records less than this many days old: You can set this field's value to a number. If you do, requests, work orders and purchase orders will only be considered `Active` if their most recent history record is less than the given number of days old. For example, if you set the value to 400, the only active requests, work orders and purchase orders will be ones whose histories have changed in the past 400 days. Typically, this means you won't see any requests, work orders and purchase orders that were closed more than 400 days ago.

Show only records since: You can set this field's value to a date. If you do, requests, work orders and purchase orders will only be considered `Active` if their most recent history record was created after the specified date. For example, if you set the value to `January 1, 2009`, the only active requests, work orders and purchase orders will be ones which have had some activity since that date.

Save: Saves the given information. The window stays open in case you want to make more changes.

Save & Close: Saves the information and closes the editor window.

Cancel: Closes the window without saving any changes you've made since the last time you saved. When you **Save** the record, the **Cancel** button changes to **Close**, indicating that you can safely close the window without losing any changes.

Close: Closes the window. This button only appears after you've saved changes with **Save** or before you've entered any data at all. Otherwise, the button is labeled **Cancel**.

The options you set through the **View** menu only affect your own MainBoss sessions. For example, if you only want to see very recent records, you might set "**Show only records less than this many days old**" to 30. MainBoss would only show you requests, work orders and purchase orders less than 30 days old. You can change this number at any time.

Your MainBoss administrator can use **Administration** | **Company Information** to set active filter defaults for everyone who uses the MainBoss database.

Selecting Multiple Table Entries

A table viewer lets you select multiple records from the table. To do this, you use the standard Windows conventions:

☐ If you hold down the **<Shift>** key, click on one record, then click on another, you will select both records and all the records in between.

☐ If you hold down the **<Ctrl>** key and click on a record, you select that record. If you hold down **<Ctrl>** and click a different record, you select that record and the first record stays selected. In this way, you can use **<Ctrl+Click>** to select any number of separate records.

After you select multiple records, you can perform the following operations:

Delete: Deletes the selected records. A window will open to ask if you really want to delete those records.

Edit: Opens the first selected record in an editor window. This window offers you all the features available in a normal editor window for this type of record. In addition, it has a **Next** button; if you click **Next**, MainBoss closes the current window and opens one where you can edit the next record that you selected from the table viewer. By repeatedly clicking **Next**, you can move through the records that you selected, editing each one as you wish.

If you use the editor window to change the contents of a record, the **Next** button disappears until you click **Save** (to save your changes) or **Cancel** (to cancel the changes). You will not be able to go to the next record until you save or cancel your changes.

Next is a drop-down button with the following entries:

Previous: Closes the current window and opens an editor window to edit the previous record from the list of selected records. By using **Next** and

Previous, you can go forward and backward through the list of selected records.

Close: Closes the current window and goes back to the table viewer. This says that you are finished using **Next** and **Previous** to edit the selected records.

Cancel: Works like the usual **Cancel** button, canceling any changes you have made.

View: Works exactly like **Edit** except that you can only look at the contents of the selected records, not make changes. You can use **Next** and **Previous** buttons to go forward and backward through the list of selected records.

Restore: Is only enabled if the records you have selected are all currently deleted. If so, MainBoss opens an editor window that works exactly like **Edit**. If you **Save** the current record, it will be restored. You can use **Next** and **Previous** to move forward and backward through the list of selected records.

Right-Clicking: When looking at a multi-selection list, you can right-click on the list to get a menu of useful actions that can be performed. For example, the menu gives you the option of checkmarking all the entries in the list or blanking them out. With long lists, we recommend that you click the "..." button to get a selection window, then right-click on the list in the window to select the entries you want.

Requests

Work requests in MainBoss Advanced are similar to those in MainBoss Basic. However, there are a number of differences:

☐ The requestor must be in your **Requestors** table (**Coding Definitions** | **Requests** | **Requestors**). This is true for both e-mail requestors (i.e. people authorized to submit requests by e-mail) and for other requestors.

    ☐ The importing process fills your **Requestors** table with all the names that were in your old **Contacts** table. After the import process, you should delete any names from whom you don't want to accept requests.

    ☐ Since there is now no distinction between requestors authorized to submit requests by e-mail and other requestors, all MainBoss Basic @Requests customers will be granted "unlimited requestor" licenses to deal with the (potential) increase in the size of the **Requestors** table.

☐ Requests may now progress through several states:

New: The request has been created but nothing has been done with it.

In Progress: Someone in the maintenance department has classified the request as in progress by clicking the **In Progress** button. When you click this button, MainBoss lets you enter any comments about the job.

Closed: The request has been closed because the job is finished. When you close the request, you have one last opportunity to send e-mail comments to the requestor by clicking **Close Request (With Comment)**.

☐ By default, e-mail acknowledgements will be sent to a requestor every time the request changes state (provided that you have the **MainBoss Service** module and you've set an option that allows this requestor to be sent such acknowledgements). Each time this happens, MainBoss asks if you want to add a specific message to the acknowledgement. If not, MainBoss just sends a standardized message indicating what happened.

Note that acknowledgements are only sent when the request itself changes state. In particular, acknowledgements are NOT sent automatically when you create a work order from a request; you must explicitly make a state change by clicking In Progress or send a comment to the requestor by clicking New Requestor Comment. Similarly, acknowledgements aren't sent automatically when you close the associated work order; the acknowledgement is only sent when you close the request itself.

For example, suppose a requestor submits a request by e-mail via MainBoss Service, calling for repairs to a piece of equipment. Here's a sample scenario showing when acknowledgements are sent:

☐ When MainBoss receives the message and creates a new request, MainBoss sends an acknowledgement saying that the request has been created.

☐ A dispatcher reads the request and clicks **In Progress**. This sends a message to the requestor saying that the request is in progress. The dispatcher might add something to the message, such as, "We'll fix that tonight after your shift is over."

☐ When workers try to do the job, they find serious problems in the equipment. The dispatcher therefore clicks **New Requestor Comment** to send another message to the requestor: "We've made some temporary repairs, but we'll have to wait till the weekend before we can deal with everything."

☐ Once the work is actually finished, the dispatcher clicks **Close Request (With Comment)** and sends a final message: "All fixed."

Note: The acknowledgment facility works for requests submitted through **MainBoss Service**, **Web Requests**, and for requests created any other way. All requests are treated the same. If you do not want to send e-mail acknowledgements to a particular user, go to that person's record in the **Requestors** list and blank out the checkbox labeled **Receive Acknowledgements**.

☐ A request has a **Work Orders** section showing work orders linked to the request. Multiple requests may be linked to a single work order and vice versa. For example, suppose three different requestors report the same problem. The three separate requests can all be linked to a single work order that fixes the problem.

As an example of the reverse, suppose a requestor reports a major problem like a flood in a bathroom. Such a problem may result in several work orders: a short-term fix (cut off the water and clean up), a medium-term fix (bring in a plumber), and multiple long-term fixes repairing damage from the flood. In this case, you could link the original request to some or all of the relevant work orders.

☐ The **E-mail Request** section of a work request is useful if you have the **MainBoss Service** module. The section will show the original e-mail message that was processed in order to create the current request.

☐ The **State History** section of a request records when the request changed its state (e.g. when it was created, when it was marked "in progress", and when it was closed).

Closing: By default, closing a work order does *not* close related requests. This is because there is no simple correspondence between work orders and requests: since a request can link to multiple work orders and vice versa, there's no correct answer to whether a request should be closed when a related work order is closed. Therefore, **you usually must close requests by hand.**

If you turn on the **Requests are automatically set In Progress when linked to a work order, and are closed when work order closes** option, closing a work order automatically closes all linked requests, provided that the requests are only linked to this single work order. (If a request is linked to multiple work orders, you must close it manually—the request isn't closed automatically, even if you close all its linked work orders.) To turn on this option, go to **Requests** in the control panel, go to the **Defaults for Requests** section, and click **Edit Defaults**. Turning on this option also means that when you create a work order from a new request, the request is automatically put into the "in progress" state.

The **Edit Defaults** window also has a field labeled **Comment to requestor for requests that are automatically closed when their linked work order closes**. This field lets you enter a message that will automatically be emailed to requestors whenever one of their requests closes because a linked work order closed. For example, you might use this message to direct requestors to a web page that contains a user satisfaction

survey. (Note that the message is only emailed out if you have licensed the **MainBoss Service** module.)

Work Orders

When you import data from MainBoss Basic into MainBoss Advanced, the process does *not* import old work orders. However, it imports information from the tables related to work orders—closing codes, work categories, projects, etc.

The sections that follow describe how work orders in MainBoss Advanced differ from those in MainBoss Basic.

Labor

☐ Work orders can now record labor charges by the hour or by the job. For example, an outside contractor may charge a fixed price for a particular type of job, no matter how long the job takes. This means there are four types of labor charges: hourly inside, hourly outside, per job inside, and per job outside.

☐ Instead of putting hourly rates directly on employee records, MainBoss Advanced introduces a new type of record called an *hourly inside* record. This record specifies an employee and an hourly rate. A similar principle applies to the other types of labor charges.

> **Overtime and Holiday Rates:** The use of *hourly inside* records makes it easy to deal with situations where an employee gets paid extra for overtime or for work on holidays. Make a record for the employee's standard hourly rate, and make additional records for the same employee, stating special rates that the employee might be paid (overtime, holidays, etc.). When writing up a work order, choose whichever pay-rate is appropriate to the work done.

One easy way to create hourly inside records is to open each employee's record (using **Coding Definitions** | **Work Orders** | **Labor** | **Employees**) and then to create appropriate hourly inside records for that employee. You do this by clicking **New Hourly Inside** in the **Hourly Inside** area of the employee records. You can also create hourly inside records with in **Coding Definitions** | **Work Orders** | **Labor** | **Hourly Inside**. You can use similar processes to create per job inside, hourly outside, and per job outside records.

Resources

Labor and materials are reserved and recorded in the **Resources** section. This section has a number of buttons:

Demands (e.g. **New Demand Item**): These are comparable to **Add Estimate** in MainBoss Basic work orders. Demands are intended to be made *before* the job begins. You can demand items, inside workers, or outside workers. Demanding a worker means assigning that person to the job. Demanding an item means reserving it for use.

When you reserve an item, MainBoss will ask you to specify a *storeroom assignment*: not just the item you want, but the storeroom that you'll take the item from.

> You must have a demand for all costs on the work order. If you don't create such demands before the job, you must create them after. You can't record actual costs until you have an appropriate demand.

Miscellaneous costs: The **Resources** section of a work order lets you record miscellaneous costs incurred during a job. This covers expenses that aren't directly related to labor or inventory materials. For example, you might have to rent special equipment for some job. As another example, a worker might have to pay for parking at a job site. Such costs are recorded in the **Miscellaneous** subsection of the **Resources** section.

Actualize: Select a demand line and click **Actualize**. MainBoss opens a window where you can record what you actually used from the demand.

You can actualize part of a demand. For example, suppose an hourly inside demand assigns a worker to a job for three consecutive days—a total of 24 hours of work. At the end of each day, you can actualize part of the original demand; this means recording how many hours the worker actually worked on that day. In this way, you get more detailed breakdowns of how labor and materials were used.

Temporary Storage Locations

Work orders may have associated temporary storage locations. For example, suppose you're doing a big renovation; you may decide to store lumber, paint, etc. in a room close to the renovation site (instead of one of your usual storerooms). In this case, you can designate the room as a temporary storage location for the renovation job. Temporary storage is an intermediate stage between being in a normal storeroom and actually being used for a job.

Remember that the point of inventory tracking is to know where materials are. For example, suppose a worker removes materials from storage one day, but calls in sick the next day. Where have the materials gone? If someone has recorded that the materials are in some temporary storage location, fill-in workers can find the materials and carry on with the job, without having to search for the materials they need.

As another example, suppose there's an emergency and you need a particular spare part. The part may not be in your regular storerooms, but it may still be available if it's in temporary storage waiting use in a different job. If you know what's in temporary storage, you might be able to claim the part for higher priority work.

Temporary storage locations make it possible to keep more accurate accounting records. For example, you can deal with the situation where you've taken materials out of a storeroom but haven't actually used them yet—those materials are in some temporary storage location waiting to be used. In this way, you never have to record that materials are used until they actually are.

> MainBoss Basic didn't record accounting information when you removed materials from storage, then later put them back (because they weren't needed after all). MainBoss Advanced preserves all such accounting information.

Here are a few more examples of temporary storage locations:

☐ When repairing a unit, you can use the unit itself as a temporary storage location. For example, in a shopping mall, each store may be considered a unit. When renovating a store for a new tenant, you can use the store itself as temporary storage for renovation materials. Similarly, when performing lengthy repairs on a piece of equipment, you can designate that equipment as temporary storage for parts, meaning that the parts are being kept someplace obvious right next to the equipment.

☐ A worker's vehicle or a repair truck could be used for temporary storage. Your records would show what materials the vehicle contains. This would make it easier for you to track the locations of key spare parts.

Note that if a vehicle always contains certain materials, you should consider that vehicle a storeroom rather than temporary storage. For example, if an electrician's van is always supposed to contain certain electrical supplies, the van should be regarded as a storeroom and its stock should be tracked just like any other storeroom.

You specify temporary storage locations in the **Temporary Storage** section of a work order. **New Temporary Storage** specifies a new location; this is automatically given the same ID code as the work order. **New Temporary Storage Assignment** specifies an item that can be stored in an existing temporary storage location. You should have a temporary storage assignment for each item that will be stored in a particular temporary location.

Once you've set up temporary storage assignments, you can transfer items to the temporary storage location. To record a transfer, click the corresponding temporary storage assignment record (in the work order's **Temporary Storage** section), then click **Edit**. This opens the temporary storage assignment record. In the record's **Activity**

section, you can use **Item Transfer To** and **Item Transfer From** (on the drop-down list of **New Item Issue**) to record materials transferred between your normal storerooms and the temporary storage location. You can also **Receive** materials directly into a temporary storage location.

> Tasks can have temporary storage assignments too. They're defined in the **Task Temporary Storage** section of the task record.

Chargebacks

Chargebacks are associated with *billable requestors* (specified in **Coding Definitions** | **Work Orders** | **Billable Requestors**. A billable requestor record has a contact person (from the **Contacts** table) and a cost center that's associated with accounts receivable. A work order may have multiple chargeback items and multiple billable requestors.

> When you import data from MainBoss Basic into MainBoss Advanced, the process does *not* put any entries into the **Billable Requestors** table.

Chargeback costs are unconnected to the actual costs on a work order. For example, when you charge back the cost of a worker's time, you can charge more or less than the actual cost of the worker's pay.

Other Work Order Differences

The **Requests** section of a work order shows any requests associated with the work order. As noted earlier, a work order may have more than one associated request (and a request may lead to multiple work orders). When you close a work order, the associated requests are *not* closed automatically—you have to close them explicitly. The easiest way to do this is to start in the work order's **Requests** section and to close each request listed there.

When you close a work order, MainBoss displays a window where you can record comments. In order to close the work order, you must **Save** this closing information.

The **State History** section of a work order records when the work order changed its state (e.g. when it was created and when it was closed).

The **Advanced** section of **Work Orders** | **Print** lets you specify additional text for printed work orders. For example, if your organization serves customers, you can add a signature line where customers sign off on a job after it's finished.

Planned Maintenance (Preventive Maintenance)

MainBoss Advanced handles planned maintenance (also called *preventive maintenance*) in a similar way to MainBoss Basic. The steps go like this:

1. Create task descriptions using **Unit Maintenance Plans** | **Tasks**.

2. Create *timing* schedules using **Coding Definitions** | **Unit Maintenance Plans** | **Maintenance Timing**. Timing schedules simply state how often work should be done; they might say "every three months" or "every 3000 miles". A single timing record can contain multiple schedules (e.g. "every three months" *and* "every 3000 miles").

3. Assign a task and a timing record to a unit using **Unit Maintenance Plans**. For example, you might assign an "oil change" task and the timing "every three months" to a company car. This states that the car should get an oil change every three months.

4. Set the *scheduling basis* for each maintenance plan. The scheduling basis is when you want to start the clock ticking for future work. For example, if you're scheduling regular oil changes for a car, the scheduling basis should be the date or meter reading of the most recent oil change. To set a scheduling basis for a maintenance plan, select the plan in **Coding Definitions** | **Unit Maintenance Plans**, then click **New Select Scheduling Basis**.

5. To create planned maintenance work orders, go to **Unit Maintenance Plans** | **Generate Planned Maintenance** in the control panel. Click **New Planned Maintenance Batch** (which means create a new bunch of PM work orders).

6. In the resulting window, click **Generate**. Then go to the **Generation Details** section to see a list of *potential* work orders. These work orders have not been created yet; you have a chance to review the list and decide if it makes sense. Reviewing the list is especially useful when you're setting up your initial timing records—you can create a timing record, then click **Generate** to see if you get the work orders you expect. Make any corrections to the timing records, then click **Refresh** to try again.

7. Once you're happy with the list in **Generation Details**, click **Commit**. This generates work orders from the entries in the details list.

> **Note:** In MainBoss Basic, generating work orders was a one-step process—you clicked **Generate** and the work orders were generated immediately. MainBoss Advanced has a two-step process: **Generate**, then **Commit**. This gives you a chance to step back and start again if you see some problem after the **Generate** phase.

Each time you generate PM work orders, MainBoss creates a record in the associated table. This makes it easy to review past generation actions: when you generated the work orders and what work orders were generated.

Importing Considerations: The following principles apply to importing planned (preventive) maintenance information:

☐ Unit maintenance plans correspond to scheduled units in MainBoss Basic. Thus your old scheduled unit records become unit maintenance plans in MainBoss Advanced.

☐ Every task record in the old database becomes a comparable task record in the new one.

☐ The import process creates maintenance timing records based on all the schedules that were used in your MainBoss Basic database. Each such record is given an identification code taken from the information in the schedule. You may want to revise some of these codes to make them more understandable.

Task Specializations

A *task specialization* is a variation on an existing task. For example, suppose that you have a standard "oil change" task, but that a particular type of vehicle requires something different from the standard: perhaps a different grade of oil, or additional inspections that should be carried out during the oil change process. Rather than writing up a whole new task for this vehicle, you can use a task specialization.

To create a task specialization, go to **Unit Maintenance Plans** | **Tasks**, select the base task (i.e. the original one), and click **New Task Specialization**. MainBoss opens a window where you can specify differences between the original task and the specialization. When MainBoss generates work orders from a specialization, information in the specialization is either added to information from the original task or used instead of the original task's information.

Standard (Boilerplate) Work Orders

Tasks can be used to create work orders on demand. For example, a property management company might create one or more tasks describing what should be done after a tenant moves out or before a new tenant moves in. Similarly, a company with a number of vehicles to maintain might have standard procedures for what should be done when changing a muffler or brake pads.

Once you've written up the task, you can create a work order at any time using the task as a template: just go to **Unit Maintenance Plans** | **Tasks** and click **New Work Order From Task**. This opens a window where you can specify the unit on which you want to perform the task. For example, if you want to perform a muffler change on a car, find the "muffler change" task and click **New Work Order From Task**. In the resulting

window, specify the car whose muffler you want to change. When you **Save & Close** the window, MainBoss will generate a muffler-change work order for that car.

You can also create work orders directly from a unit maintenance plan record. In this case, you go to **Unit Maintenance Plans**, click on the appropriate plan, and then click **Create unplanned Maintenance Work Order**. In the resulting window, click **Commit** to create the work order.

Because it's so convenient to generate work orders in this way, you might choose to create a number of unit maintenance plans specifically designed for this type of use. For example, consider the job of changing a muffler on a car. You could create a "change muffler" task and use this to create unit maintenance plan records: one for each car in your company fleet. Whenever you want to perform a muffler change on a particular car, find the "muffler change" unit maintenance plan for that car and use **Create unplanned Maintenance Work Order** to create an appropriate work order.

You might ask, "What kind of maintenance timing record would such a unit maintenance plan have?" A muffler change isn't usually done on a predictable schedule. For that reason, you could create a special maintenance timing record that stands for "no schedule". To create such a maintenance timing record, simply create one with nothing in the **Periods** section. Such a schedule will never come due. Instead, you just use the unit maintenance plan "manually" through **Create unplanned Maintenance Work Order**.

Units

MainBoss Advanced does not distinguish between equipment and spaces. Instead, it uses the term *unit* for any place maintenance can be performed (on equipment or in a space). The **Units** table viewer displays a map showing which locations contain which units.

Units can contain other units. For example, you could consider an assembly line a unit, with each machine on the line a sub-unit.

In the documentation for MainBoss Basic, we recommended defining an assembly line as a *system*, with each piece of equipment as part of the system. Now that units can have sub-units, it's more useful to think of the assembly line as the top-level unit with the line's components as sub-units. However, defining systems is still useful for grouping together units that are spread across multiple locations. For example, you can consider all your heating and air-conditioning units as part of one overall HVAC system.

You can make an existing unit into a sub-unit of some other unit simply by changing the unit's "**Location**" field. For example, suppose that MachineX, MachineY, and MachineZ are all part of an assembly line. Create a new unit named AssemblyLine, then set the "**Location**" of MachineX, MachineY, and MachineZ to AssemblyLine.

This indicates that all three machines are part of the assembly line. You can use a similar approach to reorganize all your units if appropriate.

Usages: Units may have associated *usage classification* codes. These are codes you define for any purpose you find helpful. For example, a property management company might set up usage codes to label which pieces of equipment are used by which tenant.

When a MainBoss Basic database is converted to MainBoss Advanced, usage classifications are used to indicate whether MainBoss Basic labeled the unit as equipment or a space. MainBoss Basic obsolete codes are also added to the Usage Classification table; obsolete equipment is then marked with the Usage Classification value corresponding to the old obsolete code.

Specifications

Specifications are more sophisticated versions of MainBoss Basic's templates. Like a template, a specification lets you record information about a unit. You see your specification forms with **Coding Definitions** | **Units** | **Specification Forms**.

> The importing process creates MainBoss Advanced specification forms from your MainBoss Basic templates.

A specification consists of a collection of *fields*. Each field has a name, a size (maximum number of characters allowed for field values), and a label (to be displayed when the specification is actually filled in with values). You may also specify an order in which the fields appear, so that Field 1 appears first, Field 2 appears next, and so on.

> The "**Field Order**" of a field is a number. When a specification is displayed, fields are shown in increasing order. Numbers don't have to be consecutive.

A specification editor window has several sections at the bottom of the screen. The **Fields** section shows the fields that are currently defined. **New** lets you add more fields and **Edit** lets you change the ones that are already there (e.g. changing the field's label).

You will not be allowed to change the definition of a specification field if it is already in use for one or more unit records. For example, you can't change the size of a specification field if it's already in use, since existing information might be lost.

The **Default Report Layout** section of the editor window shows how field values will appear when printed in a report. Lines in the default report have the format

```
        Edit Label: <Field name>
```

where `Edit Label` is the label you've specified for the field and `<Field name>` represents the value that the field holds for a particular unit.

The **Custom Report Layout** section of the editor window gives you the option of setting your own format instead of accepting the default layout. You specify the custom layout in the same format used in the **Default Report Layout** section. For example, suppose the default layout is given as

```
Edit Label1: <Field name1/>
Edit Label2: <Field name2/>
```

but you want your custom layout to display both values on the same line. In the "**Custom Report Layout**" section, you'd type

```
Edit Label1: <Field name1/>   Edit Label2: <Field name2/>
```

Again, the construct `<Field name1/>` stands for the value that the field holds for a particular unit. As another example of a custom layout, you could leave out the edit labels and just have:

```
<Field name1/>   <Field name2/>
```

This might be a useful way to save space in reports, especially if the field values are self-explanatory and don't need to be labeled.

The **Specifications** section of a specification form record lists units which use the specification form.

Meters

MainBoss Advanced introduces the concept of *meter classes*. A meter class is a general type of meter; for example, you might have a meter class for odometers (mileage meters), another for meters that measure hours of operation, and so on. Meter classes are defined in **Coding Definitions** | **Units** | **Meter Classes**.

The importing process makes a separate meter class for each different meter name in your MainBoss Basic database.

> **Before you move from MainBoss Basic to MainBoss Advanced, we recommend that you go through all your MainBoss Basic meters and use the same name for meters that measure the same thing.**

For example, if you have a number of vehicles, you might use the meter name `Odometer` for the odometer in each vehicle. (In MainBoss Basic, different meters can have the same name, provided they are in different units.) By giving similar meters the same name, you simplify the migration process, and set the foundation for meter classes.

> After migration, you should examine the meter classes created by import process. If there are multiple entries for the same type of meter, put all of the appropriate meters into a single class and delete the redundant classes.

Once you define a meter class, you can record meters that belong to that class. For example, once you define the odometer class, you can create meter records for all the odometers on all your vehicles.

When you define a unit, you can specify that it contains meters of various classes. For example, a car would have an odometer. The meters associated with a unit are specified in the unit's **Meters** section.

You can also define meters in **Coding Definitions** | **Units** | **Meters**. For each meter, you specify the meter class—what it measures—and the unit where the meter is found.

Relationships

MainBoss Advanced introduces the concept of *relationships* between two units, and between a unit and a person.

For example, a hot-water heater might supply hot water to several different bathrooms. If so, it's useful to record this information in your MainBoss unit records; then, if a bathroom isn't getting hot water, you can quickly identify the heater involved, even if it's a long way from the bathroom where the problem was reported.

Similarly, it may be that a piece of manufacturing equipment is usually operated by a particular person. If something goes wrong with that equipment, workers might find that it's helpful to talk to that person for more information about the problem.

Relationships are recorded in the **Related** section of a unit record. This is worth checking whenever you work on a unit, in case it holds helpful information.

MainBoss Basic had a "**Contact**" field in every unit record, indicating a contact person for that unit. MainBoss Advanced lists this contact person in the unit's **Related** section; it is not a special field, it is just a type of relationship.

Inventory Control

MainBoss Advanced has the following changes affecting inventory control.

☐ Storeroom records (**Coding Definitions** | **Items** | **Storerooms**) have a new field: "**Physical Count Sheet Sort Order**". This field takes a number that controls where the storeroom will appear in a physical count sheet. A value of 1 in this field means the storeroom will appear at the beginning of the sheet, a value of 2 means the storeroom will appear second, and so on. In this way, you can sort your storerooms into the most convenient order for taking inventory.

☐ In MainBoss Basic, inventory levels were recorded on the item record. The "**Maximum**" field indicated the maximum quantity allowed in all storerooms, while "**Minimum**" indicated the overall minimum.

In MainBoss Advanced, inventory levels are recorded on a storeroom-by-storeroom basis. A *storeroom assignment* record (**Coding Definitions** | **Items** | **Storeroom Assignments**) specifies an item and a storeroom and states the maximum/minimum quantities for that item in that particular storeroom.

When reserving materials for a work order, you specify a storeroom assignment record rather than just an item record. In other words, you specify the item you want and the storeroom where you intend to get it.

In many cases, MainBoss Advanced uses storeroom assignment records where MainBoss Basic would have used an item record.

☐ MainBoss Basic permitted you to have negative inventory counts. This could happen if a work order used more of some item than you actually had in stock. (Of course, that shouldn't have been possible.) With MainBoss Advanced, inventory counts cannot go negative—you can create a work order that demands more of an item than you currently have, but you can't actualize such a demand until you have enough stock to cover it.

Importing Storeroom Assignments: In MainBoss Basic, maximum and minimum quantities were specified for the item as a whole rather than for individual storerooms. When the importing process creates storeroom assignments, it assigns the *total* maximum and minimum values for the item to each storeroom.

For example, suppose that in MainBoss Basic you wanted to keep a minimum of 10 light bulbs on hand and a maximum of 20. In MainBoss Advanced, the importing process would assign a minimum of 10 and a maximum of 20 *to each storeroom* where you had light bulbs stored. You should therefore review all the minimum and maximum values for each storeroom assignment, once the import process is finished.

Cost Centers and Inventory: In MainBoss Advanced, every storeroom assignment has an associated cost center. This cost center is used whenever a quantity of the associated item is added or subtracted from the storeroom:

☐ When an item is received from a vendor and placed into a storeroom, two accounting records are created: one for accounts payable to the vendor, and one increasing the value of the storeroom assignment by the same amount. (Note that the accounting records are written when an item is received, not when it is ordered.)

☐ When an item is issued to a work order, two accounting records are created: one that decreases the value of the storeroom assignment by the cost of the item, and one that charges the cost of the item to the work order (in accordance with the work order's expense model).

The same principle applies when you issue an item for some other purpose. In this case, the issue code specifies the cost center that gets charged. (See **Coding Definitions** | **Items** | **Issue Codes**.)

□ When an item is transferred from one storeroom to another, two accounting records are created: one that decreases the value of the storeroom assignment in the first storeroom, and one that increases the value of the storeroom assignment in the second storeroom.

In MainBoss Basic, there were various ways that a storeroom could have a non-zero quantity of an item, but a zero value for that item. This can't happen in MainBoss Advanced.

Because MainBoss Advanced does things differently, you should pay attention to the cost centers that you associate with storage assignments.

Purchasing/Receiving

For the most part, MainBoss Advanced handles purchase orders in a way similar to MainBoss Basic. The following differences apply:

□ A newly created purchase order is said to be in the *draft* state. It stays in this state until it is issued (or canceled).

□ In MainBoss Advanced, items are considered on order as soon as they are put on a purchase order. (In MainBoss Basic, items were only put on order when the purchase order was issued.)

□ In order to receive anything on a purchase order, you must first issue the purchase order. You must then create a receipt record to associate with receipts (either using **Purchase Orders** | **Receipts** or by clicking **New Receipt** in the purchase order's **Receipts** section). After that, you can use the purchase order's **Receiving** section to record the receipt of line items—click an item in the list of things ordered, then click the **Receive** button.

Administration

Administrative functions are provided through **Administration** on the control panel. In particular, you record your license keys using **Administration** | **Licenses**. For information on recording license keys, see the *Installation and Administration Guide*.

MainBoss Service

MainBoss Basic has the @Requests add-on. In MainBoss Advanced, the same functionality is handled by the **MainBoss Service** module. This is administered through MainBoss itself rather than a separate program (**Administration** | **MainBoss Service**).

By default, people must be in the **Requestors** table (**Coding Definitions** | **Requests** | **Requestors**) before they're allowed to submit requests by e-mail. However, you can configure MainBoss Service so that it accepts e-mail from anyone. In this case, e-mail from a previously unknown address automatically creates an entry in the **Requestors** table. If your MainBoss Service license only allows a limited number of requestors, mail from new addresses will be rejected once the **Requestors** table reaches the maximum that your license allows.

MainBoss Service also sends *notifications* to people who are assigned to requests, work orders, and purchase orders. As noted earlier, notifications are sent whenever a history record is created for the associated request, work order, or purchase order.

For more on using MainBoss Service, see the *Installation and Administration Guide*.

Backups and Restores

As discussed in the *Installation and Administration Guide*, backups are most efficiently done through the automated operations of SQL Server (provided that your version of SQL Server supports such operations). However, MainBoss itself has a facility to ask SQL Server to make an immediate backup. This feature uses the **Backup** button of **Administration** | **Database History**.

We strongly recommend that you make regular backups of your MainBoss database and store copies in a safe place. Crashes happen, and you can't recover what you haven't saved.

We also recommend that you *mirror* your disks, so that you don't put all your eggs in one basket. For more information, see

http://www.mainboss.com/english/resources/tips/mirror.shtml

Restore operations can be done through the `Select Maintenance Organization` window. Use the drop-down arrow associated with **Add Existing Organization**, then choose **Restore Organization** or **Create New Organization from a Backup**. For more information on backup and restore operations, see the *Installation and Administration Guide*.

Appendix A: Basic/Advanced Correspondences

The following table summarizes correspondences between concepts in MainBoss Basic and in MainBoss Advanced.

Basic	Advanced
@Requests	**Administration** | **MainBoss Service**
Access Codes	**Coding Definitions** | **Access Codes**
Adjustment Codes	**Coding Definitions** | **Items** | **Adjustment Codes**
Adjustments	In a storeroom assignment record (**Coding Definitions** | **Items** | **Storeroom Assignments**), drop the arrow on **New Physical Count** and click **New Item Adjustment**
Archiving	Not applicable in a SQL Server environment
Asset Accounts	**Coding Definitions** | **Units** | **Asset Codes**
Audit	Use facilities in SQL Server
Backup	**Administration** | **Database History**, **Backup**
Buildings	**Coding Definitions** | **Locations** (postal addresses)
Company Information	**Administration** | **Company Information**
Contacts	**Coding Definitions** | **Contacts**
Cost Centers	**Coding Definitions** | **Cost Centers**
Equipment	**Units** (marked by `Equipment` in "**Usage**")
Equipment Types	**Coding Definitions** | **Units** | **Categories**
Issue Codes	**Coding Definitions** | **Items** | **Issue Codes**
Issues	In a storeroom assignment record (**Coding Definitions** | **Items** | **Storeroom Assignments**), click **New Item Issue**
Item Categories	**Coding Definitions** | **Items** | **Categories**
Item Value	In the **Pricing** section of an item record (**Items**), click **New Item Pricing**
License Keys	**Administration** | **Licenses**
Meters	**Coding Definitions** | **Units** | **Meters** (after setting up **Coding Definitions** | **Units** | **Meter Classes**)
Obsolete Codes	**Coding Definitions** | **Units** | **Usage Classifications**
Ownerships	**Coding Definitions** | **Units** | **Ownerships**
Payment Terms	**Coding Definitions** | **Purchase Orders** | **Payment Terms**
Personnel	**Coding Definitions** | **Work Orders** | **Labor** | **Employees**
Physical Counts	In a storeroom assignment record (**Coding Definitions** | **Items** | **Storeroom Assignments**), click **New Physical Count**
Physical Inventory Sheet	**Items** | **Reports** | **Location and Status**

Preventive Maintenance	**Unit Maintenance Plans** | **Generate Planned Maintenance**
Price Quote	In the **Pricing** section of an item record (**Items**), click **New Item Pricing**
Priorities	**Coding Definitions** | **Requests** | **Priorities** and **Coding Definitions** | **Work Orders** | **Priorities**
Projects	**Coding Definitions** | **Work Orders** | **Projects**
Purchase Orders	**Purchase Orders**
Receipts	In a storeroom assignment record (**Coding Definitions** | **Items** | **Storeroom Assignments**), drop the arrow on **New Item Issue** and click **Receive**
Repair Closing Codes	**Coding Definitions** | **Work Orders** | **Closing Codes**
Requests	**Requests**
Restore	**Restore Organization** in the `Select Maintenance Organization` window
Schedules	**Coding Definitions** | **Unit Maintenance Plans** | **Maintenance Timing**
Service Contracts	**Coding Definitions** | **Units** | **Service Contracts**
Shipping Modes	**Coding Definitions** | **Purchase Orders** | **Shipping Modes**
Space Types	**Coding Definitions** | **Units** | **Categories**
Spaces	**Units** (marked by `Space` in "**Usage**")
Storerooms	**Coding Definitions** | **Items** | **Storerooms**
Systems	**Coding Definitions** | **Units** | **Systems**
Tasks	**Unit Maintenance Plans** | **Tasks**
Templates	**Coding Definitions** | **Units** | **Specification Forms**
Trades	**Coding Definitions** | **Work Orders** | **Labor** | **Trades**
Transfer	In a storeroom assignment record (**Coding Definitions** | **Items** | **Storeroom Assignments**), drop the arrow on **New Item Issue** and click **Item Transfer To** or **Item Transfer From**
Units of Measure	**Coding Definitions** | **Units of Measure**
User Permissions	**Administration** | **Users** and **Administration** | **Security Roles**
Vendor Types	**Coding Definitions** | **Vendors** | **Categories**
Vendors	**Coding Definitions** | **Vendors**
Viewing Programs	Unnecessary
Warranties	Write up as service contract with manufacturer
Work Categories	**Coding Definitions** | **Work Orders** | **Work Categories**
Work Orders	**Work Orders**

INDEX