MainBoss Advanced 3.0:
Configuration Quick-Start

Introduction
Welcome to MainBoss Advanced 3.0

This guide provides a quick introduction to setting up MainBoss Advanced 3.0 before putting the software into active service.

Note: Before you read this guide, please read *Getting Started* so you understand the fundamentals of how MainBoss is used.

- For information on installing the software, see the *Installation and Administration Guide*
- For complete details of MainBoss set-up, see the full *Configuration* guide.
- For information about day-to-day use, see *Operations Quick-Start* or the full *Operations* guide.

The Golden Rules of MainBoss Set-Up

Rule 1: Plan in advance for the things you need, and don't put in anything else.

Here's an example: every work order can be assigned a priority. However, MainBoss doesn't have a predefined set of priority classifications—*you* decide what priorities you want to use. For example, you might use a simple system like

```
High priority
Average
Low priority
```

or time-oriented priorities like

```
Immediately
By the end of the shift
Within 24 hours
Within a week
Whenever
```

Of course, your organization may already have an established set of priorities, in which case, you just use those.

It's up to you to choose what priority system will work best with your operations. However, if you aren't ready to decide, **don't use priorities at all**. Work orders don't *need* priorities...and it's better to use a system you have confidence in than something you make up without much thought. Changing your mind and reorganizing after the fact can be a lot of work.

If you don't use priorities to begin with, you can start using them later. Once you're comfortable with MainBoss and the "newness" has worn off, you can think about what

priority system would be useful and relevant. You still don't have to assign priorities to every work order—just the ones that have higher or lower urgency than normal.

The same principle applies in everyday operations: don't use things you don't need. For example, MainBoss lets you record *access information* on a work order. This might indicate that you need to obtain someone's permission before taking equipment offline or that repairs are only allowed at certain times of day. However, if there are no special restrictions on a particular job, don't fill in the "**Access Code**" field. Leave the field blank in normal situations, so that when you *do* fill it in, workers are more likely to notice that this job has special circumstances.

Rule 2: Choose useful categories and codes

MainBoss offers the option of defining categories and information codes for various types of data. For example, you can assign categories and/or codes to your work orders, your requests, your suppliers, and so on.

Your organization may already have categories and codes for this kind of information. If not, you might create work-order categories like

```
Plumbing
Electrical
Mechanical
Inspection
Clean-up
```

Using such categories can be helpful in getting a clear picture of your operations. Do we spend more on plumbing than electrical jobs? How much time do we actually put in on clean-up? How many inspections has Joe done in the last month?

As for codes, MainBoss lets you define closing codes to be assigned to work orders after the job is finished. Many organizations use these codes to specify the original source of the problem, such as

```
Operator error
Accidental breakage
Poor lubrication
Normal wear
Vandalism
```

By labeling each work order with an appropriate code, you can later get detailed reports on how common each type of problem is and the cost associated with each problem "class."

One of the most important functions of MainBoss is to provide answers to questions—not just your own questions, but questions asked by customers, upper management, and other people outside the maintenance department. When setting up MainBoss, it's important to think about the questions you'll need to answer and to record information from which the answers can be extracted—you can't change what you can't measure.

MainBoss can answer questions easily if you attach categories and codes to your records.

- You might choose to figure out categories and codes *before* you enter any other data. This provides a structured way of thinking about your operations: you come up with an organization plan before you put in detailed information about jobs, equipment, work materials, etc.
- You might decide on categories and codes *after* you've used MainBoss for a while. You can then draw on your experience with the software to decide which categories and codes will be useful.

Either way can work. If you're already familiar with computerized maintenance management, you may have an existing set of categories and codes you like to use. If not, you may prefer to wait until you're more comfortable with MainBoss.

Whether you start with categories and codes or add them later, you have to think about what will actually be useful to you. If, for example, your company's budgeting process breaks down expenses in a certain way, then you'd better use the same approach in MainBoss. Otherwise, you'll end up with headaches trying to deal with differences.

Even if you don't have to worry about external factors, you still should think through what categories and codes you want to track. Otherwise, just omit categories and codes from work orders until you have time to decide what would be useful. Choosing a set of categories and codes that *don't* work for you (or don't give you the whole picture) will be more trouble than it's worth.

Rule 3: Make a phase-in plan

If you've never used computerized maintenance management before, we recommend that you phase in MainBoss a little at a time. Trying to computerize your entire operation in one fell swoop will be a daunting job. Furthermore, you'll be making decisions before you have a chance to get some experience with the software.

As an example of phase-in, you might begin with just tracking work requests (problem reports). Once you're comfortable with that, you might move on to work orders and the scheduling of planned maintenance. After that, you add inventory tracking...and so on.

The point is to *have* a plan: know what you want to get working first and what steps to take to reach your goals. What information do you have to collect and record? What decisions do you have to make? Who has to be trained to use the software?

Don't just start entering data and hope for the best.

- Think ahead
- Take manageable steps

- Only start with what you know you need; add extras later

Rule 4: Be Consistent

Before you begin entering *any* information, make sure you have a coding policy that will be followed by everyone who uses MainBoss. Stick to the policy with absolute consistency. This is particularly important in large organizations—you have to make sure every MainBoss user is "on the same page."

For example, consider the names by which you identify your equipment. You might decide on names like this:

```
Air conditioner 1
Air conditioner 2
Air conditioner 3
```

If someone then uses the name

```
AC 4
```

it will throw off the consistency of your records. In a list of equipment sorted in alphabetical order, the "odd man out" won't be sorted with the other air conditioners. It will be harder to find the unit you want when you're visually searching the list; the unit may also slip between the cracks when you're printing off reports about your equipment.

The same applies to all other types of information in your database. **Names must be consistent: similar things should have similar names.** In particular, you should choose the format of names so that similar things appear together when sorted in alphabetical order.

Decide on standard name formats *before* you start entering data. This decision affects how much benefit MainBoss can provide. A haphazard naming scheme reduces the software's ability to organize and analyze data in effective ways.

Rule 5: Be Safe

SQL Server has facilities for backing up and restoring data. **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

Basic Principles of MainBoss Advanced

Unit: **A unit is anything that might require maintenance.**

This includes equipment, vehicles, and places. In a factory, for example, your units will mostly be pieces of manufacturing equipment. In a shopping mall, each store might be considered a separate unit, with additional units for washrooms, the parking lot, and so on.

Sub-Unit: **A sub-unit is part of a unit.**

For example, you might sub-divide complex equipment into smaller sub-units so that you can track the maintenance record of each piece. Similarly, in a shopping mall, you might subdivide large areas into smaller ones (e.g. “Department store, north section”, “Department store, east section”, and so on) so that workers have a clearer idea of where to go when they’re sent to fix a problem.

There is no clear dividing line between what is a sub-unit and what is simply a spare part. For example, if a piece of equipment has a self-contained motor, do you treat the motor as a sub-unit or simply a spare part? The answer depends on whether or not you want to track the motor’s maintenance independently from the containing equipment. If you’re likely to move such motors from one unit to another and if you care about the motor’s past maintenance history, then you should treat the motor as a sub-unit. Otherwise, the motor doesn’t have to be treated as a sub-unit.

Request: A request (or work request) reports a problem.

Requests are often based on complaints from people outside the maintenance department. Requests provide preliminary information about the problem such as the location of the problem, the name of the person reporting it, the date/time the report was received, and a brief description of the problem itself.

Since problems may be reported by non-maintenance personnel, requests are designed to be simple enough that anybody can fill one out.

Work order: A work order is a detailed description of work to be done.

Work orders are designed to be filled out by maintenance personnel. In many organizations, work orders are written up by the maintenance dispatcher, then issued to the people who'll do the actual work.

Work orders are more detailed than work requests. For example, a work order may specify the amount of time the job is expected to take, the materials to be used, and so on. Such details are not present in a work request.

General Principle: Requests describe *problems* while work orders describe *solutions*.

Items: Items are materials used in the course of your maintenance work.

Items include spare parts, tools, lubricants, and anything else whose use you wish to track. MainBoss can report on your materials inventory, including how much of an item you have on hand, where particular items are stored, and when you need to re-order more stock.

Task: A description of work to be done during planned maintenance or in some other standard maintenance job.

For example, you might have a task record describing a standard oil change and inspection on a vehicle. Task records often contain step-by-step instructions or check-lists of actions to be done during the work.

Unit Maintenance Plan: A complete description of a planned maintenance job. This includes the unit that should be serviced, the task that should be done, and the timing for doing that work.

For example, a unit maintenance plan might describe regular oil changes on a car. The unit is the car whose oil needs to be changed; the task might be a check-list of what should be done in each oil change; the timing might be "every three months or 3000 miles".

MainBoss uses your unit maintenance plans to create actual work orders. For example, when the time comes for an oil change on a particular vehicle, MainBoss will create a work order for the job using information from the unit maintenance plan record.

Maintenance Organization: A collection of data describing the operations of a single maintenance organization.

While many MainBoss customers will only have one maintenance organization, some may have more. For example, consider a property management company that maintains properties for several different clients. The company may choose to keep each client's data separate from the others, or may decide to have a single database combining data from all clients.

Storeroom: A place where you store spare parts and other materials.

For each storeroom, you create *storeroom assignments* which specify what items the storeroom should contain and the maximum/minimum quantities for each item.

Note that a "storeroom" doesn't have to be a normal room. For example, if you have service trucks that each are supposed to contain certain quantities of standard work materials, you can treat each truck as a storeroom and can track the contents to make sure the truck has everything it needs.

Expense Category: A way to label costs on a work order.

For example, you might have separate categories for inside labor (your own personnel), outside labor (done by hired contractors), electrical supplies, plumbing supplies, spare parts, and so on. You assign an expense category to each cost on a work order so that you can track and categorize your expenses.

Expense Model: A list of what expense categories are allowed on a particular work order.

For example, a property management company may have a separate expense model for each tenant. This makes it possible to associate different expense categories with different tenants. In future versions of MainBoss, expense categories and expense models will be the basis for more sophisticated account tracking.

Getting the Most Out of MainBoss

Before you begin configuring MainBoss, it's useful to think about some basic principles of computerized maintenance.

1. This is your institutional history: The information in your MainBoss database is what remains after employees leave the company. It's what you can put on the table when upper management asks you to justify your expenses. It's the records you can produce if someone tries to sue you for negligence. It's the repair-cost comparisons you can make when you have complete maintenance histories on similar pieces of equipment. It's the overview you get on suppliers and equipment and downtime and who did what when.

So the start of MainBoss configuration is deciding what information you need to preserve and how your maintenance department will use MainBoss to preserve it.

- **Nothing In, Nothing Out:** If you don't use MainBoss, it can't help you. If your people don't make an effort to record the information you need, your records will give an incomplete and distorted picture of your operations.
- **Garbage In, Garbage Out:** If your people do record information in MainBoss, but do so in an inconsistent manner, you'll still have difficulty getting a return on your investment.

2. Know what you want out of your system, and record the information you need: Establish clear policies for what will be recorded and who will do the recording. For example, when a work order is completed, various information should be recorded as part of the closing process. Who records that information?

- Workers? They know the most about what happened on a job, but the more time they spend on the computer, the less time they spend with their tools.
- Clerical assistants? They're likely more adept with computers, but they have no direct knowledge of what happens at job sites and they may not be knowledgeable about maintenance in general.
- Maintenance managers? They know maintenance and they may like to be hands-on about what gets entered as MainBoss data, but their time may be too valuable to spend on data entry.

There is no universal answer. Typically, workers will be responsible for entering some types of information, clerical assistants will be responsible for other data, and various levels of management will choose to do some jobs themselves. Different organizations will make different decisions, but thinking things through and establishing clear policies is vital.

3. Don't ask people to record information they don't know: If you give someone the responsibility of recording a certain type of information, make sure that person has a good clear way of determining the situation.

For example, consider a car mechanic who performs oil changes. In some organizations, different vehicles are associated with different accounting codes, and the cost of the oil change should be assigned to different accounts depending on the vehicle. If mechanics are required to keep track of these different accounts, they should have an easy way to determine which account is associated with which car; otherwise, they'll make mistakes and the information in your database will become less useful (garbage in, garbage out).

As another example, suppose a plumber has to rip out part of a wall during a plumbing job and then repair the wall afterward. Do your policies say that this whole operation should be categorized as plumbing, or do you write it up some other way (e.g. two separate work orders)? Whoever writes up the job should have a clear understanding of what your policies say and how to decide tricky situations. In addition, your procedures should ensure that the person who writes up the work order has the information he or she needs in order to understand what was actually done.

4. Assign a data-checker: Someone has to be responsible for the way MainBoss is used at your site. This person should ensure that others are using the software according to your policies. We call this person the data-checker.

For example, suppose your maintenance department has decided that workers will enter closing information on work orders. If some worker isn't recording such information or is doing so inconsistently, the data-checker should tell the worker what's required. If the worker still doesn't provide the required comments, the data-checker should inform someone with the authority to enforce that the worker follows policy.

Similarly, the data-checker should ensure consistent use of MainBoss features, including consistency in the format used for entering information. Haphazard record-keeping can cause just as much trouble as no record-keeping at all.

As noted earlier, your MainBoss data is your institutional memory. If the right information doesn't go in, or is entered in an unusable form, something important may have been lost.

Coding Definitions

MainBoss set-up is done in the **Coding Definitions** section of the control panel. Basically, you decide which parts of the program you want to use, then fill out the corresponding subsections. For example, if you want to use MainBoss to process work orders, you fill out entries in **Coding Definitions** | **Work Orders**. If you want to use MainBoss for planned maintenance, you fill out **Coding Definitions** | **Unit Maintenance Plans**.

As noted in *The Golden Rules of MainBoss Set-Up*, you don't have to fill out every possible table—just the ones that you'll find useful. You can always expand your use of MainBoss later.

MainBoss Modules: **Coding Definitions** will only show entries that are relevant to the modules you have licensed. For example, if you haven't licensed the **Inventory** module, you won't see the section **Coding Definitions** | **Items**.

Getting Help: Remember, you can press the **<F1>** function key at any time to get online help for whatever window you're looking at.

Recommended Set-Up Order

The order in which you set up your coding definitions depends on which features you'll be using. However, the following list should help you map out how to proceed.

This list doesn't cover everything you might want to set up: just the basics used by most maintenance departments. For more details on set-up, see the full *Configuration* guide.

General Tables

The following tables are relevant to all aspects of MainBoss use:

Coding Definitions | **Users**: Record the login names (and domain names if you use domains) of people who'll be allowed to use MainBoss.

Coding Definitions | **Locations**: Record *postal address* information for all the buildings where you perform maintenance work.

Record *sub location* information for floors and rooms within those buildings.

As desired, record other postal address information (e.g. addresses of suppliers and employees).

Coding Definitions | **Contacts**: Record contact information for employees, suppliers, and other people you might wish to contact during your work.

Remember, you don't have to record everything all at once. For example, you don't have to record full contact information on every person relevant to your operations. If you cover the ones you use most often, you can go back and add others as the opportunity arises during day-to-day operations.

Suppliers and Contractors

Vendors information covers those you do business with outside your organization: suppliers and contractors. This information is used by the **Inventory**, **Work Orders**, and **Planned Maintenance** modules.

Coding Definitions | **Vendors** | **Categories**: Define categories for classifying your suppliers.

Coding Definitions | **Vendors**: Record information about your vendors including addresses and contact people. The **Vendors** list should also include outside contractors (e.g. plumbers and other independent tradespeople) whom you occasionally hire.

Unit Information

Unit information describes the equipment and places you maintain. This information is used by the **Requests**, **Work Orders**, and **Planned Maintenance** modules.

Coding Definitions | **Access Codes**: List any special restrictions relating to access that may affect your work. For example, if some pieces of equipment can only be serviced in off-shift hours, you might create an `OFF-SHIFT` access code. (Access codes are used in unit records and on work orders to describe when jobs can or can't be done.)

Coding Definitions | **Units** | **Usage Classifications**: Defines broad classifications for your units, e.g. distinguishing equipment from places.

Coding Definitions | **Units** | **Categories**: Define categories for grouping your units.

Coding Definitions | **Units** | **Systems**: Define any major *systems* whose maintenance you want to track. (A system is made up of units that are related or linked to each other in some sense, e.g. your HVAC system or your fire/security system: something where a problem with one unit may well affect the whole system. Contrast this with, say, company vehicles, where a problem with one vehicle usually has no effect on others. `Vehicles` is typically a unit category; `HVAC` is typically a system.)

Coding Definitions | **Units**: Record basic information on the units you'll be maintaining, especially their locations (and associated contact people, if useful). Also record access codes, categories and systems, if applicable, and the **Value** section of the unit record (which records purchase and replacement price, if applicable).

Coding Definitions | **Units of Measure** (often abbreviated UOM): List the various units of measurement that you use (e.g. pounds and feet or kilograms and meters). These units are used in equipment meters and in measuring quantities of work materials. For indivisible items (e.g. light bulbs), MainBoss traditionally uses the UOM `EA` standing for `EACH`.

Coding Definitions | **Units** | **Meter Classes**: Types of meters that you'll be tracking (e.g. odometers, kilowatt-hours) and the units of measure each type of meter uses.

Coding Definitions | **Units** | **Meters**: The actual meters whose readings you'll record. For each meter, you state its class and the unit where the meter is found.

Coding Definitions | **Units** | **Service Contracts**: Service contracts that cover one or more units. Note that equipment warranties can be written up like service contracts with the manufacturers; therefore the **Service Contracts** table covers both service contracts and warranties.

Inventory Information

Inventory information describes the materials you use in your work, e.g. spare parts and lubricants. This information is relevant to the **Inventory**, **Work Orders** and **Planned Maintenance** modules.

Coding Definitions | **Items** | **Categories**: Define categories for your spare parts and work materials.

Coding Definitions | **Items** | **Storerooms**: Specify where you store your materials.

Coding Definitions | **Items**: Record basic information on your spare parts and materials.

Coding Definitions | **Units** | **Parts**: Record which items serve as spare parts for which units.

Coding Definitions | **Items** | **Storeroom Assignments**: Record which items are stored where. For each item in each storeroom, you record the maximum quantity allowed in the storeroom and a minimum quantity below which you want to restock the item (i.e. purchase more from a supplier).

Coding Definitions | **Items** | **Adjustment Codes**: Adjustment codes are used to record changes to inventory records. While you may define multiple adjustment codes for all the reasons you may need to adjust your inventory, at this point, all you need is one for *physical counts*—recording how much of an item is actually in inventory.

Record physical inventory: At this point, you're ready to record what's in your materials inventory. To do this, go to your list of storeroom assignments (**Coding Definitions** | **Items** | **Storeroom Assignments**). For each record, click **Edit**, then go to the record's **Activity** section and click **New Physical Count**. At the same time, you can record the value of each item.

Request Information

Request information let you create work requests. This information is used in connection with the **Requests** module.

Coding Definitions | **Requests** | **Priorities**: If you wish to assign priorities to requests, record your priority classes.

Coding Definitions | **Requests** | **Requestors**: If you will only accept requests from certain people (especially in connection with the **@Requests** module), list the names and e-mail addresses of the authorized requestors. (Alternatively, you can set up @Requests so that it adds people to the list of authorized requestors as you receive e-mail from them. However, your license may only allow a limited

number of requestors, and if you reach this number, requests from new e-mail addresses will be rejected.)

In a **Requestors** record, you specify an entry in the **Contacts** table. The **Contacts** record should contain the person's e-mail address, *in the format used by the person's mailer software*. (In some organizations, a person might have multiple aliases that all go to the same mailbox. Mail sent from that box is always sent from a specific e-mail address; that's the address which should go into the **Contacts** table.)

Work Order Information

Work order information let you create work orders. This information is used in connection with the **Requests** module.

Coding Definitions | **Work Orders** | **Priorities**: If you wish to assign priorities to work orders, record your priority classes.

Coding Definitions | **Work Orders** | **Work Categories**: Define categories for your work orders.

Coding Definitions | **Work Orders** | **Closing Codes**: Define codes to be used when marking a work order as closed. Many companies use these codes to specify the original reason for the work (e.g. normal wear and tear, planned maintenance, operator error, vandalism, etc.). For a deeper discussion of the value of closing codes, see the *Configuration* guide.

Coding Definitions | **Work Orders** | **Labor** | **Trades**: List your employee classes (if any).

Coding Definitions | **Work Orders** | **Labor** | **Employees**: List your employees. (Each employee record contains a reference to the employee's information in the **Contacts** table.)

Coding Definitions | **Work Orders** | **Labor** | **Hourly Inside**: Record how much you charge per hour for work by each employee. In most cases, there will only be one "hourly inside" record for each worker. However, some organizations may pay different rates depending on what the worker does, and may also have special rates for overtime, holidays, etc. For example, Joe Smith may have two different "hourly inside" records: `Joe Smith Regular` and `Joe Smith Time-and-a-Half Overtime`.

Coding Definitions | **Work Orders** | **Labor** | **Per Job Inside**: This is like "hourly inside" except for situations where workers are paid by the job rather than by the hour. For example, you might hire a car mechanic to do oil changes and pay by the car rather than by the time taken. Record any such arrangements.

Coding Definitions | **Work Orders** | **Labor** | **Hourly Outside**: This is like "hourly inside" but the records state hourly rates for outside contractors. (Often, you

record this information “on the fly” each time you hire a contractor for a job, but if you have a standing price arrangement, you can create a standard “hourly outside” record ahead of time.)

Coding Definitions | **Work Orders** | **Labor** | **Per Job Outside**: This is like “hourly outside” but for cases where you pay by the job. Again, you usually record this information on the fly, but you can create an appropriate record ahead of time if you have a standing arrangement with some contractor.

Planned Maintenance Information

Planned maintenance information let you schedule inspections and other types of preventive maintenance. This information is used in connection with the **Planned Maintenance** module.

Coding Definitions | **Unit Maintenance Plans** | **Tasks**: Create descriptions of what should be done in a standard planned maintenance job (e.g. a standard oil change or a standard HVAC inspection). Your task list can also include descriptions of standard jobs that aren’t done on a specific schedule, e.g. a standard muffler replacement; you can then use the task description to produce a quick muffler work order if needed.

Coding Definitions | **Unit Maintenance Plans** | **Maintenance Timing**: Create schedules for planned maintenance jobs, either by date (“every three months”), by meter (“every 3000 miles”), or both.

Coding Definitions | **Unit Maintenance Plans**: Create plans that say which units will be serviced with which tasks on which timing schedules.

Select Scheduling Basis: In order to start a unit maintenance plan, you have to give it a starting date or meter reading. For example, in order to start scheduling oil changes on a car, you have to tell MainBoss when the car last got an oil change. MainBoss can then figure out when the next oil change should be. In order to set this basis, go through your list of unit maintenance plans (**Coding Definitions** | **Unit Maintenance Plans**) and click **Select Scheduling Basis** for each.

The above list should help you get started with the basic functions of MainBoss. You can add more as time goes on. For example, you can use **Coding Definitions** | **Units** | **Service Contracts** to record service contract information on your units; however, you start using MainBoss productively without recording service contract information.

Division of Duties

In order to use MainBoss, you have to decide who does what: which maintenance personnel will be responsible for recording different types of information.

To help you make this decision, this chapter lists a number of *roles* that should be filled in order to use MainBoss productively. In most maintenance departments, one person will likely fill several roles; in a small maintenance department, a single person might do everything; in large departments, on the other hand, several people might all do the same role. Furthermore, there is no precise dividing line between roles. Even so, it is still useful to recognize the possible divisions of labor between personnel.

The roles are:

Implementation Committee: Decides how MainBoss will be used at your site and lays the ground rules (e.g. what naming conventions you'll use). The committee should establish a phase-in plan and get buy-in from everyone who has a stake in the outcome (management, workers, customers, etc.).

SQL Server Specialist: Ensures that SQL Server 2005 is installed and configured correctly for use by MainBoss. This person will also ensure the ongoing correctness of the configuration and make regular backups of the MainBoss database.

Installer/Administrator: Installs MainBoss software on all the computers where it will be needed and creates an initial database. This person must have Windows Administrator privileges in order to the install the software. The installer also records license keys for the database and make sure that the Configuration Specialist (described next) is in the **Users** table. [Uses MainBoss's **Administration** features]

The Installer/Administrator is the only person who needs Windows Administrator privileges. Other MainBoss users need no special Windows privileges.

Configuration Specialist: Creates the initial MainBoss set-up. This person should be very familiar with all aspects of MainBoss, and with the policies of your maintenance department. Once the initial configuration is finished, this person's job is done (except for occasional tweaks to the system). [Only uses **Coding Definitions**]

Technical Advisor for Configuration: Advises the Configuration Specialist on technical matters, particularly on planned maintenance tasks. This person should have extensive expertise in maintenance so that the task descriptions properly describe what's required for your existing units. [Advises Configuration Specialist on **Coding Definitions**, especially **Unit Maintenance Plans**]

Configuration requires knowledge of MainBoss and knowledge of maintenance. We've split these into two separate roles because they're two different types of expertise.

Equipment Specialist: Records information about units, or advises someone who actually types in the information. This person will be significantly involved in the configuration process, deciding what information should be recorded about each unit and gathering that information—specifications, spare part lists, warranty information, and so on. Someone must continue to fill this role after configuration, as new equipment is purchased and old equipment is retired. In particular, the equipment specialist should record planned maintenance procedures for new equipment and put those procedures into the schedule. [Uses **Units** and **Coding Definitions** | **Units**]

Help-desk: Receives problem reports and creates requests based on the information. Help-desk personnel don't need much expertise in maintenance or MainBoss use—just enough to ask relevant questions about the problem and to write an understandable description of what's gone wrong. [Only uses **Requests**]

Dispatcher: Creates work orders from requests. This may require obtaining more information about the problem, assigning personnel, reserving materials, and so on. The dispatcher generally needs a good knowledge of maintenance and of MainBoss. [Only uses **Work Orders**]

Workers: Report back information from the job site; either they enter the information into MainBoss themselves, or they write out the information by hand for someone else to transcribe. Workers must know what information they're expected to record (e.g. time spent, materials used, closing codes, etc.). If they're asked to use MainBoss directly, they must have a basic familiarity with the software. [Only use **Work Orders**]

Chargeback Administrator: Writes up chargebacks after a job is finished. This person must be familiar with contract agreements between your maintenance department and the chargeback customer—what the customer should and shouldn't be charged for, as well as the applicable rates. [Only uses **Work Orders** and **Coding Definitions** | **Work Orders** | **Billable Requestors**]

Storeroom Personnel: Record the receipt and use of materials, take physical inventory, and ensure that supplies are restocked as required. [Only uses **Items**]

Analyst: Prepares reports on various aspects of your maintenance operations. This person is usually a manager or a manager's assistant. [Uses **Reports** in various sections of the control panel]

Data-Checker: Maintains the consistency of the database and advises all other MainBoss users on use of the software. [May use any part of the software]