

# Database Backup, Restore, and Archive Guide For **Journyx Timesheet** version 5.0 and higher

Document version 2.0 – January 2006

Introduction and Summary.....	1
Purpose.....	1
Backup and Restore Summary.....	1
Warning! .....	2
Backup Techniques.....	2
Recovery Strategy.....	4
Backup Procedures.....	5
Running <i>backupdb</i> .....	5
<i>backupdb</i> options .....	5
Database Archive (Date Range based backup).....	6
Basic Archive Usage.....	7
Archive Limitations and Considerations.....	7
Understanding Archive Date Ranges.....	8
Archive Preparation Checklist .....	8
Archive Options .....	9
Restoring Timesheet Databases .....	12
<i>restoredb</i> options.....	12
Other Backup / Restore Topics .....	13
Automating Backups.....	13
Postgresql Issues .....	14
Native Backup Formats.....	16

## ***Introduction and Summary***

The information in this document only applies to Journyx Timesheet 5.0 and higher except where noted. Please make sure you have the latest maintenance release for your version of Journyx Timesheet as frequently these releases contain backup and restore improvements.

The information about the "Archive" feature (date range limited backups) only applies to Timesheet version 5.5.2 and higher. The rest of the backup and restore options apply to all versions of Timesheet 5.0 and higher.

## **Purpose**

This document describes the Journyx Timesheet database backup and restore utilities including the "Database Archive" feature. These utilities allow you to save your Journyx Timesheet data to a file on disk and restore it at a later time in case of system failure. You should implement a disaster recovery plan and do backups on a regular basis. These backup files should be stored in a safe place separate from your Timesheet server. This document explains the different strategies and options for performing backups. All of the command-line options for these utilities are explained in detail along with general documentation and answers to frequently asked questions about backup and restore.

## **Backup and Restore Summary**

To create a Timesheet backup, run the **backupdb** command and give a filename for the backup file:

```
$ backupdb -v My_Timesheet_Data_January2006
```

## Journyx Timesheet Backup, Restore and Archive Guide

A file with the name `My_Timesheet_Data_January2006.jx` will be created in the current directory. To restore that file at a later time run the `restoredb` command and give the filename for the backup file:

```
$ restoredb -v My_Timesheet_Data_January2006.jx
```

Please be warned that running `restoredb` will **permanently erase** all data in the site and replace it with the data contained in the `.jx` backup file.

The `-v` option on both of those commands enables printing of “verbose” status messages. That gives you a better idea of how long each program will take to finish.

You can also create an “Archive” backup file. An archive takes all the time, expense and mileage sheets that fall within a date range that you provide and writes them to a `.jx` backup file. You then have the option to purge (delete) those archived sheets from your live Timesheet site. The archive file can be saved away in a safe place or restored to an alternate server for reporting on archived data. To create an archive simply give the `-A` (capital A for Archive) option to the `backupdb` command. The program will prompt you for the date range and other options.

```
$ backupdb -A My_2004_Timesheet_Archive.jx
```

The complete set of `backupdb` and `restoredb` options are described in detail later in this document.

### Warning!

Any system administrator responsible for a Journyx Timesheet site should become familiar with the contents of this document and implement an effective data recovery plan. Journyx, Inc. is **not responsible** for the loss of any data for any reason from any system under the care and custody of the customer. Journyx strongly recommends that you backup your Timesheet system on a **nightly basis** and maintain backup media and old backup files going back a minimum of one week.

Journyx Timesheet stores all of your Time and Project data in a RDBMS (relational database management system), such as Microsoft SQL Server, Oracle, or Postgresql. Note: Receipt Images and Saved Reports are stored directly in a filesystem instead of in the database. It is highly recommended that your I.T. staff perform regular backups of the RDBMS hosting your Timesheet data whether you are using the "Internal Database" or an external RDBMS that you have provided, such as Oracle.

In addition you should make regular file-level backups of the Timesheet application server (and the database server, if separate) to ensure the fastest recovery in the event of a total system loss. Remember: in a total loss situation having a copy of the database itself is only one part of the overall recovery strategy. Besides the database itself you'll need a working computer, operating system, network, and an installation of the Timesheet server application. Please consider all of those factors when formulating your recovery plan.

### Backup Techniques

There are two common methods for making a backup of Journyx Timesheet data. One method is to make a direct backup of the RDBMS using the native recovery tools provided by your RDBMS software. The other method is to use the `backupdb` and `restoredb` commands provided by Journyx. The “pros and cons” of each backup solution are described in this table with some specific recommendations for a backup strategy following that.

A third much less common backup option is to use a program such as Norton Ghost to create a backup image of an entire hard disk partition. If you use this technique, it is highly recommended that you make additional backups with the Journyx `backupdb` utility and/or the RDBMS native recovery tools.

<b>Timesheet Backup / Restore (Backupdb and restoredb)</b>	<b>RDBMS Native Backup / Restore (also called 'Recovery Tools')</b>
<b>Cross Platform File Format</b>  The backup file can be restored on any operating system that Timesheet supports and with any type of database system. This is also useful in case Journyx Support needs to look at a copy of your database.	<b>Non-portable Format</b>  The backup file may only be restored on the same kind of database system – including the same RDBMS version in most cases.
<b>Automatic Version Migration</b>  Backup files from old versions of Timesheet can be “migrated” into a newer version with some limitations.	<b>No Migration</b>  A “native” RDBMS backup file from an old version of Timesheet may not be restored to the newer version.
<b>Date Range Archive Supported</b>  <b>backupdb</b> allows you to select a date range for the backup and to optionally delete these archived records. (Timesheet version 5.5.2 or higher only.)	<b>No Archive Feature</b>  No easy or automatic way to archive away records that fall under a certain date range.
<b>Saves Objects from Disk</b>  The Timesheet <b>backupdb</b> format also saves Expense Receipt images, Saved Reports, your Professional Services custom programs and configuration as well as your site config file.  All of these objects are stored on the application server filesystem instead of in the RDBMS.	<b>Does Not Save Disk Objects</b>  RDBMS native backups do not include these vital objects stored in the filesystem.  To backup these objects, you must either use the Timesheet <b>backupdb</b> command, or make a separate filesystem backup of the Timesheet installation directory.
<b>Uses Compression</b>  The Timesheet backup format automatically compresses the data with the popular and reliable ZIP format. The <b>.jx</b> files can be opened by most Zip compatible programs and the plain text backup file inside is easily extracted and read by humans.	<b>No Compression by default</b>  The native backup format of RDBMS's are typically not compressed by default though the option will often be available. Compressed backups can take up as little as 10% of the original file size.
<b>Site must be offline to backup</b>  While a backup is being made with the <b>backupdb</b> tool, the Timesheet site is offline and unavailable to your users.	<b>Online (“Hot”) Backups</b>  Most RDBMS's allow you to safely perform a backup while the site is in active use. Sometimes this is called a ‘hot’ backup. This increases site availability.

### Relatively Slow

Timesheet version 5.5.2 improved **backupdb** speed by 20% but a backup of a very large site can still take as much as 30 minutes or more.

The **restoredb** program is very slow compared to most RDBMS native restore programs. The speed is greatly improved in version 5.6 but still relatively slow.

The best way to make the **restoredb** faster is to have fewer records in the first place; in particular, try to avoid activating the “historical extra fields<sup>1</sup> for time records” feature unless absolutely necessary.

It is also a good idea to use the “Archive” feature on a regular basis (see below) to trim out unneeded old data. This will make your ongoing backups and any restore operations much faster.

### Very Fast

RDBMS native backup and restore operations are typically very quick. Usually they are limited only by disk input / output capabilities.

As you can see from the table above, the two major disadvantages of the Journyx Timesheet backup format are that it is slow compared to native RDBMS recovery utilities and that the site must be offline to make a backup. (The site must be offline for a restore as well, even if you are using the RDBMS Recovery Tools.) The slower restore time compared to a native RDBMS restore may mean that your site is unavailable for longer periods of time while restoring.

## Recovery Strategy

If your I.T. staff is capable of understanding and effectively using the native RDBMS recovery tools we recommend that you use these on at least a nightly basis in addition to making weekly backups in the Timesheet native **backupdb** format. The backup process should be completely automated if possible.

If you prefer not to use the RDBMS recovery tools we do suggest that you make a Timesheet native backup at least once a day. Again this will usually be at night or during non-peak hours. Even if you plan to use the RDBMS recovery tools exclusively we recommend that you make Timesheet native backups at least weekly. You may choose a minimally disruptive time to perform the backup such as a weekend night. Automated backups are ideal for that. (See the "Automating Backups" section below.)

One important point to remember when planning your recovery strategy is to never make the current backup overwrite a previous one, especially if that previous backup is your only recent backup. Always backup to a **new** file. Ideally the filename should contain the current date and time as a convenient way of keeping the files distinct.

You can store the backup files on any media that is convenient for your staff whether DAT or other tape formats, recordable CD or DVD formats or network storage that is backed up by a separate process. The Timesheet backup utility cannot normally write directly to a tape or CD-R. You must first create the backup file on disk, then you can copy that **.jx** file to tape or other backup media. Timesheet backups can be written directly to network shares provided that the appropriate access permissions are in place.

If you reuse your backup media, it is recommended that you keep 7 days worth of backups as a bare minimum. It is also highly recommended that you make a full backup in the Timesheet format at least monthly and store that backup in a safe place off-site (away from your place of business) in case of a fire or other disaster at your office. This is good advice for all of your business data, not just the Timesheet database.

---

<sup>1</sup> Also known as “Custom Fields” in version 7.0 and higher.

The rest of this document describes the Journyx Timesheet native backupdb and restoredb utilities. A section at the end of the document does have some additional technical notes about performing native RDMBS backups.

### **Backup Procedures**

Journyx Timesheet provides the **backupdb** command-line utility to perform backups. The files that backupdb creates have the **.jx** extension. A **.jx** file is a bundle of files compressed in the common "zip" archive format. These files contain your Timesheet database tables as well as other objects, such as receipt images, saved reports and your site `config` file.

### **Running backupdb**

To run backupdb on Windows, launch a command window by clicking:

```
Start -> Programs -> Journyx Timesheet -> Timesheet Command Line Prompt
```

Then you may simply type **backupdb** plus any option you need (see below) as well as the name of your new backup file.

On Unix, before you run **backupdb** or any other Timesheet command line utility, make sure you have sourced the Timesheet setup file first. To perform a backup on Unix, you must be logged in as the same account that was used to install Journyx Timesheet.

For example, if Timesheet runs under the **journyx** user, and is installed in the `/home/journyx/jt55` directory, log in as **journyx** and do this:

```
~:$ cd /home/journyx/jt55
```

That directory is also know as **\$WTHOME** since it is your installation directory.

```
~/jt55:$ source pi/bin/setup  
~/jt55:$ backupdb -v MyBackupFile.jx
```

The **.jx** backup file is always written to the current directory, unless you specify the full path to the output file, such as:

```
~/jt55:$ backupdb -v  
/netshares/BackupVolume/2005/September/Timesheet_20050905_backup.jx
```

Please note: when you run **backupdb**, it will automatically take your Timesheet site offline. If any of your users attempt to access the site while you are making a backup, they will get a "Site is Offline for Maintenance" message until the backup completes. Please note that your Timesheet service must be running in order to make a backup if you are using the "Internal Database" on Unix **or** are using a version of Timesheet older than version 5.5. Use **wstart** to start the service on Unix or use the Services control panel under Windows.

If the backup fails due to lack of disk space or for any other reason it will prompt you at the console whether or not to remove the database busy flag. Saying "yes" will put the Timesheet site back online. If you are running the backupdb command through an automated process such as a timed "at job" or "cron job" this may render your site offline if your script is unable to answer the prompt about the database busy flag. See the section below about automated backups.

### **backupdb options**

These options may be given to the **backupdb** command. In addition to the options listed here, you must provide a filename for the output backup file. If you do not give the filename a **.jx** extension, it will be added automatically. For instance, if you type:

## Journyx Timesheet Backup, Restore and Archive Guide

```
$ backupdb myfile
```

A backup file named `myfile.jx` will be created.

The "Archive" options for date range based backups are listed separately in the next section.

```
-?          Show the usage screen
```

This option displays a short screen describing the rest of the options.

```
-v          Show verbose output
```

If you do **not** give this option, the `backupdb` command is silent. It will not print any output or progress indication. If you give the `-v` option, `backupdb` will keep you informed about its progress. In most cases we recommend that you use the `-v` option as it will give you a rough idea of how fast the backup is proceeding and how long it might take to finish.

```
-l          List what would be backed up
```

When you give this option, `backupdb` does **not** actually perform the backup. It simply lists the objects that would have been backed up as well as the source locations for those objects such as the directory for receipt images.

```
-d          Backup only the database schema and data
```

When this option is given the backup only includes the actual data in the RDBMS system. It will **not** include receipt images, saved reports or professional services files. This option is rarely used.

```
-p (lowercase) Backup only the professional services scripts
```

When you give this option, the backup will **only** have the Professional Services files. It will not include the actual Time data, receipt images or anything else. This option is useful if you wish to move your professional services scripts to another system without moving the actual Time data. Please note that this only backs up Professional Service files in the "cgi" directory. To ensure a complete backup of all Professional Services deliverables please do a filesystem backup of the entire Timesheet install directory, such as `C:\Program Files\Journyx`.

```
-O          Allow backupdb to overwrite existing files
```

Timesheet `backupdb` will not overwrite an existing backup file unless you give this flag. For safety it is never wise to overwrite your only good backup while in the process of making a new backup, so we do not recommend the use of this option.

```
-x          Ignore database busy flag
```

Normally the `backupdb` command will not perform a backup if the "database busy" setting is active on your Timesheet site. "Database busy" is turned on by certain maintenance utilities such as other instances of `backupdb` or `restoredb` or the `guilockout` command line tool. When you specify `-x` it will ignore this setting and perform the backup anyway. This option does **not** turn off the prompt about the database busy flag that comes up if the `backupdb` program encounters an error. There is currently no way to disable that prompt.

### ***Database Archive (Date Range based backup)***

Timesheet version 5.5.2 introduced the capability of archiving only a specific range of dates with the `backupdb` program. This creates a regular `.jx` backup file except that the time, expense and mileage sheets in the file will be constrained to the given date range. There are a number of reasons you might want to do this.

## Journyx Timesheet Backup, Restore and Archive Guide

The most important reason to use the archive feature is to get rid of old data from prior years to improve system performance. If you no longer need to report on this data getting rid of it with Archive is a good way to trim the overall size of your database and greatly improve performance. You can still report on the data by putting it on an “archive” server away from your main production Timesheet server.

Another reason to perform an Archive is in order to send a recent copy of your database to Journyx Support for troubleshooting without including all your old data from prior years. If you plan to do this, please see the note below about the `-N` option to disable the inclusion of the entire database in the archive file.

### Basic Archive Usage

To make a date range archive of your Timesheet data, just run the `backupdb` command with the `-A` (capital A for Archive) option. The program will prompt you for the range of dates and whether or not to delete the archived records. In most cases, to create a date range archive you do not need to provide any additional options other than the `-A`. The additional options listed below are for certain specialized options, or to avoid being prompted for the archive dates.

The program will create a regular `.jx` Timesheet backup file. The backup file will contain all the time, expense and mileage sheets that fall within the given date range. No other data will be constrained by the date range; your archive file will always have a complete version of the Project tree and the Field Values (such as Pay Types and Bill Types.)

The archive program will prompt you near the end whether you want to delete the archived data from your live site. If you say **yes** to this question the archived sheets that fall within the date range will be *permanently deleted* from your live production Timesheet site. The only copy of that data will be in the newly created archive file.

To review or report on this archived data you will need to restore the archive to a non-production “test” or “archived” Timesheet site. This requires a completely separate Timesheet install. If your Timesheet server is running under Microsoft Windows you will need a separate machine for your “test” install as multiple Timesheet installs cannot coexist on the same Windows machine. If running under Linux or Unix you can create a separate Timesheet install on the same machine as long as you do it as a different system user. If you are using an external database you must also use a separate database user account.

To access the archived data simply restore the archive file to your “test” archive reporting site with the **restoredb** command. You may then log into the test site and run reports.

Please note that in most cases your separate “test” or archive reporting server will need its own license key. Please contact the Journyx Sales department if you need a separate key for archive reporting; often these keys can be obtained at a significant discount.

### Archive Limitations and Considerations

The archive tool has some limitations and other issues that are important to consider when making decisions about archiving old data.

The most important limitation is that it is not currently possible to merge two separate archive files into the same test / reporting site. For example, if you make an archive of your 2002 data and restore that to a test site, and then later make an archive of the 2003 data, you cannot merge the 2003 data into the same test site that has the 2002 data. You would have to either set up another test site or manually switch between the data sets on the one site. Using the native RDBMs backup/restore tools will make the switching process much faster.

It is also not possible to move data from an archive reporting site back onto your main production site. Make sure you do not enter any *new* time data on your archive reporting site. You may wish to configure the login screen of the test / reporting site to have a message warning users that it is an archive site and not to enter any new data.

The primary cause of large database sizes and slow archive operations is using an excessive number of “historical extra fields.” (Also called historical Custom Fields in version 7.0 and higher.) Every custom field that has the “historical for time records” option set generates a large amount of extra data that must be dealt with by the backup, restore and archive code. These options also affect the overall site performance during peak usage hours. Disable the “historical” feature on any extra fields unless absolutely necessary. As of version 5.6 Timesheet built-in reports can report on the “current” value of extra fields as opposed to the “historical” value.

The archive operation itself can be slow in some circumstances. If you have version 5.6 or 7.0 of Journyx Timesheet, please make sure you obtain the “Archive patch” from Journyx Support. When this patch is installed the archive operation will be significantly faster.

If you have a large volume of data to purge (delete) after archiving, this can take a very long time in your database server. Experiments have suggested that to delete 1,000,000 records takes approximately 1 hour of database time. A site which has 1,000,000 time records to archive plus 15 “historical extra fields” associated to time records may take up to 16-20 hours just to delete the records. That is in addition to approximately 1 hour of time to perform the archive / scanning process. Please be patient and do not cancel the archive process prematurely. If you are concerned whether the process has “hung up” check your database server for activity. RDBMS's such as Oracle have management tools which will show you progress in very long running requests such as a massive delete transaction.

See also the notes below about the `-z` option and transaction / rollback space in your database.

### Understanding Archive Date Ranges

It is important to understand how **backupdb** selects which Time, Expense, and Mileage sheets will be included in the archive. For example if you create an archive of all Pre-2002 records do not be surprised if some records from early January 2002 are included in the archive. That is because a user's 1-week time sheet may run from Sunday, December 29th 2002 to Saturday, January 4th 2003. If you have selected 20021231 (December 31st 2002) as your archive cutoff date some "overlap" sheets may contain January 2003 days.

Normally these overlap sheets will be included in the archive but they will **not** be deleted even if you choose to purge out the archived sheets. In other words, by default Time/Expense/Mileage sheets will only be purged if **all** of the dates of the sheet fall within the selected archive range. The `-y` option overrides that feature and makes it purge all archived sheets including overlapping ones.

This lets you create archives based on any time period that you choose without worrying about splitting up Time / Expense / Mileage sheets that have already been approved by your staff.

By default overlap sheets will be included in your archive. You can use the `-x` (capital X) option to **eXclude** overlap sheets. This means that a sheet will not be included in the archive unless **all** of the dates in the sheet fall within the given archive range.

### Archive Preparation Checklist

When you are considering doing an archive or purge operation please run through the following checklist to make sure that you are fully prepared.

- Have you determined the exact date range that you wish to archive? You can choose either a specific date range (such as January 2002 to December 2002) or you can choose an open ended range such as “all sheets before December 2002.”
- Have you determined what you will do with the archived data? Will you simply keep the file for backup purposes or do you plan to set up the data on a separate legacy reporting site?
- Do you need to purge (delete) the archived sheets from your live production site? Using the purge option to delete archived sheets will greatly extend the total amount of time to complete the archive on very large sites. The speed of the purge operation depends entirely on the speed of your database system in deleting a large amount of data under a transaction. Usually this in turn depends on the

speed of your disk(s) and the amount of data involved. It is highly recommended for the best performance that your RDBMS uses separate physical disks or arrays for data, indexes and transaction rollback space. This applies to general Timesheet performance as well as the Archive operation.

- Do you have enough time available? Archiving and purging several years worth of data from a large site with thousands of active users can potentially take up to 48 hours or more in extreme cases. For these large sites consider starting the archive on a Friday morning and letting it run over the weekend.
- Can you do a test run on a separate test site using a copy of the database? Doing this will give you a better idea of how long it will take in production. The time will be proportional to the overall size of your database and the size of the date range being archived and purged. Doing a test run will also help you be aware of any problems before they affect your production site such as running out of database transaction space.
- If you have done a test run have you restored the archive to a separate test site and verified that the correct data is archived and reportable?
- Have you notified your users that the Timesheet application will be offline for a certain period of time? You should allow them a final chance to get into the system before taking it offline.
- Are you using the interactive mode (**-A**) or are you setting the archive start and end dates manually with the **-S** and **-E** options? If the latter then you must use the **-P** option to purge (delete) records. You will only be prompted to delete records if you use the **-A** interactive Archive mode.
- Have you considered using the **-S**, **-E** and **-P** options instead of **-A** interactive Archive mode? Using these options let you avoid the prompt about purging (deleting) records. This prompt only comes up after the main part of the archive operation has finished; this may happen late at night when no one is around to answer the prompt. If you want to delete archived records you should use the **-P** option to avoid the prompt. This will reduce your total amount of downtime by eliminating time wasted waiting on the prompt.
- Have you checked with your database administrator (DBA)? Have you informed him or her that the purge operation will require heavy database resources and may tie up the database for a period of hours or days in some cases?
- Does your RDBMS system (such as Oracle) have a large amount of free transaction space? (Also known as RBS, rollback segments, or just rollback.) The space you will need is related to the size of the purged records compared to the total size of the database. For instance, if your Timesheet site has 5 years of data and you are archiving 2 years, you will need at least two-fifths (40%) of the data-tablespace size available as rollback space.
- Have you made an RDBMS native backup “just in case”? For instance, have you used the Oracle EXP utility to make a copy of the database?
- Have you double-checked your archive plans with members of your organization who are involved with Timesheet administration, configuration and support?
- Have you studied this document in detail? Have you read about all the different command line options?

### Archive Options

This section details all of the **backupdb** command options related to the Archive feature. Note that all of the Archive related options for **backupdb** are capital letters. These options are only available in Timesheet version 5.5.2 and higher.

**-A**                      prompt for specific dates to Archive

This is the main option to invoke Archive mode. If you give this option you do not need to give the options below. With the **-A** option the backupdb utility will prompt you for the start and end dates of the archive. It will also prompt you at the end whether or not you want to purge (delete) the archived sheets from your live Timesheet site. However if you use the **-S** and **-E** options (see below) to set the date range of the archive, “Archive mode” is implied and you do not need to use the **-A** option itself. Please note that in that case you will not be prompted to delete records and must use the **-P** option to force deleting of archived records.

## Journyx Timesheet Backup, Restore and Archive Guide

When prompted for the archive start and end date you must enter the date in YYYYMMDD form (4 digit year first, then 2 digit month then 2 digit day. Use a 0 if necessary, such as 01 for January.) Use the - character (dash) to indicate "no limit." The program will prompt you for a new date if the date you gave is not a valid YYYYMMDD date.

If you give - (no limit) for both the start and end dates of the archive, then a regular backup is performed.

```
-S YYYYMMDD    specify a start date for Archive (or - for no limit.)
```

Specify a start date for the archive in YYYYMMDD format. You may also use the - (dash) character to mean "no limit." If you specify -S but do not specify -E, the program will assume you mean the end of the archive should not be limited.

For instance, this command:

```
$ backupdb -S 20020101 myarchive.jx
```

will produce an archive starting on January 1<sup>st</sup>, 2002 and containing all records up to the current date, since -E is not specified.

Please note that if you use either the -s or -E option then Archive mode is assumed and you do not need to give the -A option. More importantly if you use either -s or -E then you should be aware that you will **not** be prompted to delete any records. If you use -s or -E then you **must** use -P if you want the archived records to be purged (deleted) from the site.

```
-E YYYYMMDD    specify an end date for Archive. (or - for no limit.)
```

This specifies the end date of the archive, or - (dash) for no limit. Similar to -S, if you specify -E but do not specify -S, then the program assumes the archive has no limit for the start. For instance, this command:

```
$ backupdb -E 20020101 myarchive.jx
```

will produce an archive including all sheets from the past up to January 1st, 2002.

```
-P (capital)   purge (delete!) archived Sheets from the live site without  
prompting. (Use caution with this option!)
```

Use this option to force the Purge (deletion) of all archived sheets without prompting. Normally you will be prompted whether or not to purge the archived sheets. This option skips the prompt and always purges.

```
-Y            Purge archived sheets with some days outside the archive range
```

As noted above, by default, a sheet is only purged if **all** of the dates in the sheet fall within the archive range. This option forces it to purge even these overlap sheets, assuming that purge is enabled. This option has no effect when combined with -X (see below.)

```
-X (capital)   eXclude sheets whose dates do not fall entirely within the given  
date range. The default is to include all sheets that have at least 1 day within  
the archive date range.
```

This option excludes 'overlap' sheets -- those sheets where at least one of the days falls outside of the specified archive date range. When -X is enabled, these overlap sheets are **not** included in the archive, and they are **not** purged.

The default is "inclusive mode" -- a sheet is included in the archive if **any** of its dates are within the archive range, even if some of the dates are outside the range.

## Journyx Timesheet Backup, Restore and Archive Guide

`-N` Do not include 'full' backup in Archive (saves space.)

Normally backupdb (even in Archive mode) produces a file that contains a **full** copy of your entire Timesheet database. If you have used the Archive option and try to restore an archive `.jx` file, the **restoredb** program will ask you whether you want to restore just the archive date range or the full database. This is done for your convenience and the safety of your data.

However, sometimes you may wish to produce an archive that does **not** contain the entire database, and you can use the `-N` option to do this. One reason to do that is to make the resulting `.jx` backup file much smaller, since it only needs to contain the data for the date range you specified. This is often useful when creating an archive to send to Journyx Support for troubleshooting. Often, Support will only need the last 2 months of data in order to diagnose a problem.

It is also possible to open the `.jx` backup file with a Zip utility such as Winzip to remove the 'full' data from an archive backup. Simply delete the `database.backup.trimmed` file within the `.jx` zip archive. **Do not touch** the `database.backup` file as this contains your primary archived time data. It is best to make an extra backup copy of the `.jx` backup file before you attempt to modify it.

Please note that when you give the `-N` option the `database.backup.trimmed` file inside the `.jx` archive is still created but it will be very small in size and will not contain your time data.

`-Z` Do not use a database transaction when purging records.

Normally you will not need this option. This option is provided in case your RDBMS does not allow very large transactions. Do not use this option unless you have been told to do so by Journyx Support.

Purging archived records normally happens within a single database transaction. This is in case there is any kind of error during the delete process; in case of error, the database will be quickly restored to the state it was in just prior to the backup. In other words, either all of the archived records are safely purged or none of them are.

When you are purging a year's worth of Time and Expense data from your system it often requires a very large transaction. Occasionally a RDBMS such as Oracle can run out of transaction log space. That means you are not allowed to purge your archived records in a single transaction. Even without transaction space problems, a very large delete of millions of records can take 12 to 24 hours in some extreme cases so please be patient. (See notes above about large deletes.)

The best way to handle a transaction space situation is to have your DBA (database administrator) increase the transaction log space in the Timesheet database and then re-run the backup/archive.

This temporary disk space for transaction information is called "rollback segments" or RBS in Oracle. In Microsoft SQL Server, it is just called the transaction log file. Normally, SQL Server adjusts the space for this file automatically. Depending on its configuration, Oracle does **not** adjust RBS space automatically and your DBA must make manual adjustments. The most common Oracle error indicating too little RBS space is `ORA-01555 "snapshot too old"` error.

On Oracle you must have at least one free RBS that is large enough by itself to contain the entire transaction. It will not work to have lots of different small rollback segments. The size required obviously depends on your particular data. The best way to tell is to look at the (uncompressed) size of the `database.backup` file within the `.jx` archive that you just created. In general, you must have a free RBS with at least 80% of the size of your uncompressed `database.backup` file to do a purge.

If it is not possible to increase your transaction log space you can specify the `-Z` option to do the purge in multiple transactions. This avoids the problem of needing one single large transaction logfile. The downside and danger to this option is that if something goes wrong during the purge, some of the archived records may

have been deleted while others were not. This may leave your database in an inconsistent state and lead to unpredictable and unsupported errors.

In the unlikely event that this happens it is recommended that you do a full restore of Timesheet from your most recent backup. In most cases you can use the archive file that was just created prior to the purge error.

### **Restoring Timesheet Databases**

The **restoredb** program allows you to restore the backup files that you created with the **backupdb** command. If you are on Unix, be sure to source the Timesheet `pi/bin/setup` file before running **restoredb** as described above in the section for **backupdb**.

In general, you do not need to provide any options to **restoredb** other than the name of the `.jx` backup file. The program will prompt you for any information that it needs. However, a number of options are provided to tweak the restore behavior and in some cases avoid the prompts.

#### **restoredb options**

```
-R          Do not restore receipt images
-S          Do not restore saved report files
-P          Do not restore professional services scripts
```

The three options above should be self-explanatory. You may use any combination of these options.

```
-A          Clean restore of special Database Archive format (this will
overwrite any current data)
```

This option skips the prompt when you attempt to restore an Archive (date range limited) backup. The prompt asks you whether you want to restore the specific date range or the entire backup if available. If you specify this `-A` option, it will skip the prompt and just restore the archived (date-range) data set. This option is only available on Timesheet 5.5.2 and higher.

```
-r          Restore receipt images without restoring database
```

Just restore a set of Expense Receipt Images without touching the actual database.

```
-s          Restore saved report files without restoring database
```

Just restore a set of Saved Report files without touching the actual database.

```
-p          Restore professional services scripts without restoring database
```

Just restore a set of Professional Services scripts without touching the actual database.

```
-F          Do not force foreign key constraints after migration
```

Do not use this option unless instructed to do so by Journyx Support.

If Timesheet is unable to activate a foreign key (FK) constraint due to corrupted records it will automatically delete and log the corrupted records. This option overrides that behavior; unless the FK is mandatory it will be disabled in case of any corrupted records.

```
-u          Create an undo file
```

This option creates a backup file of the current Timesheet database before restoring the file named on the command line. You will be prompted for the name of the 'undo file'. The undo file is just a regular Timesheet `.jx` backup.

```
-k [license] Specify license key
```

## Journyx Timesheet Backup, Restore and Archive Guide

Allows you to specify a new license key for the site on the command line without being prompted.

If you do not specify this option, you will be prompted for a new license key. At that time you may simply press Enter if you wish to keep the existing license key that is currently on your site. The existing key will not be displayed.

`-x` Ignore database busy flag

Restore the database even though the 'database busy' (system maintenance) flag is already set. Be careful when using this flag as some other administrator on a different account may actually be doing maintenance work at the same time.

`-q` No output or warnings

If this option is set **restoredb** will not print any status output to the console.

`-D` Use default dirs for receipt images and saved reports

Normally the restore program will prompt you for the filesystem location to save Receipt Images and Saved Reports. If you use this option the prompt is suppressed and the default locations are used. These are based on the Timesheet install directory.

`-d` Turn debugging on

Normally, "extended tracebacks" are turned off automatically after a restore. This option turns on extended tracebacks (also called 'debugging' in the Logging Preferences screen) after a restore.

`-l` Turn debug logging on

Usually "diagnostic logging" (`debug.log`) is turned off automatically after a restore. This option turns on diagnostic logging for the restore itself, and leaves the option on after the restore.

You should not leave diagnostic logging on for extended periods on your production server unless instructed to do so by Journyx Support. The log file (`debug.log` in your Timesheet `tmp` directory) can quickly fill up all available disk space on a busy site. Leaving Diagnostic Logging turned on will also slow down overall site performance.

`-?` Show this usage screen

Shows a short help screen listing the **restoredb** options.

## ***Other Backup / Restore Topics***

### **Automating Backups**

These instructions only apply to Microsoft Windows 2000 and higher. It is also possible to schedule automated backups on various Unix systems using the "crontab" command. Please consult your Unix System Administrator or documentation for details.

To set up a scheduled task in Windows 2000 or higher, click `Start -> Control Panel -> Scheduled Tasks`. Scheduled Tasks are not available in NT 4.0. (The "at" command is available in NT 4.0, but it is more difficult to use and will not be covered here.)

Double-click "Add Scheduled Task." Click Next.

It will ask you to select the program to run. The Timesheet `backupdb` will usually not be on the default list. Therefore, click Browse. Select the `backupdb.bat` file. Assuming you installed Timesheet to the default location, that will be:

## Journyx Timesheet Backup, Restore and Archive Guide

```
C:\Program Files\Journyx\jwt\bin\backupdb.bat
```

Next it will ask you to select a name for the task. Name it something like "Timesheet Nightly Backup."

Normally you will want to run the task every night, so select Daily and then click Next.

It will then ask you more questions about when to run the task. Choose a time for the task. We strongly recommend that you test the backup task at least once during normal business hours, and then after you confirm that it works set it to run at an off-peak time such as 4 a.m. After you have selected a run time click Next. After you have created the scheduled task, you can test it at any time by right-clicking on it and selecting 'Run.'

It then will ask you which Windows user should run the task. It is critical that you choose the same user account that you used to install the Journyx Timesheet software. If you use a different account here, it will not work. Be sure to give the appropriate account password for that account and then click Next.

Now you must be sure to set some options on the `backupdb.bat` command line, so be sure to check "Open advanced properties" on the final screen of the wizard.

In the Properties screen, change the Run: line as follows:

(Original)

```
"C:\Program Files\Journyx\jwt\bin\backupdb.bat "
```

(New)

```
"C:\Program Files\Journyx\jwt\bin\backupdb.bat" -0x ..\..\NightlyBackup.jx
```

Note: you cannot use a path with spaces when editing the command line for a Scheduled Task; therefore we use the `..\..\` relative path as above. An example of a path with spaces is "C:\Program Files\Journyx"

It is important to note that the scheduled task as described above will cause it to overwrite the previous night's backup every time. This is not a good idea. If something goes wrong with the backup, you have just deleted your previous (good) backup. We recommend that you create a separate batch file which saves the old backup file before running `backupdb`.

For instance, you can create a new file in the `%WTHOME%\jwt\bin` directory called `autobackup.bat`. Be sure to create the file with a plain text editor.

Put commands like this in the `autobackup.bat` file:

```
c:  
cd "c:\Program Files\Journyx"  
copy NightlyBackup.jx PreviousNightlyBackup.jx  
cd jwt\bin  
backupdb.bat -0x "C:\Program Files\Journyx\NightlyBackup.jx"
```

Then you can change the Windows Scheduled Task to run the `autobackup.bat` file instead of `backupdb.bat`. Be sure to test the Scheduled Task by right-clicking on it and selecting "Run."

## Postgresql Issues

Postgresql is the "Internal Database" that Journyx includes with the Unix Timesheet product. It is also possible to use an external Postgresql instance that is maintained by your staff. In either case, there are some issues to be aware of when working with the Timesheet **backupdb** and **restoredb** commands.

**"Enabling referential integrity" is slow.**

## Journyx Timesheet Backup, Restore and Archive Guide

The main issue to be aware of is that Postgresql is slow when it comes to "Enabling referential integrity." That is something that happens at the very end of a restore. "Referential Integrity" (RI) refers to a set of rules in the database that prevent anyone from modifying the database in a way that would corrupt certain relationships.

For example, there is a rule that prevents anyone from deleting a Pay Type code if there are any Time records that use that Pay Type.

The problem is simply that Postgresql takes a lot longer to activate these rules than the other databases supported by Journyx. The amount of time depends on the number of records in the database (as well as the speed of your system.)

This problem is present whether you use the Timesheet restoredb utility or the Postgresql native backup format (see below.) However, since restoring from a native Postgresql backup is overall much quicker, the problem may not be as noticeable.

Currently there is no solution or workaround to this problem other than to use a different database system with Timesheet, such as Oracle or DB2. You may experience this issue whether you are using the (Unix) Internal Database provided by Journyx or an external Postgresql database. The Timesheet Internal Database for Windows is based on Microsoft SQL Server and does not have this issue.

Journyx has raised this issue with the Postgresql developers. Postgresql is an Open Source database solution, developed by volunteers all over the world. So far, Journyx has not received any information about when this issue might be fixed in Postgresql. We will keep our customers posted on developments in this area.

### **Run VACUUM FULL periodically**

Journyx recommends that you periodically issue a "VACCUM FULL" command in your Postgresql database to improve performance. You should do it on a weekly basis if you have a busy site. You should also do a VACUUM FULL if you have just created an archive of old sheets and purged them from your main Timesheet site.

Postgresql is the only database that implements a manual "VACUUM" command. The other databases do this automatically behind the scenes. The VACUUM command tells the database to clean up old data and compute statistics about your data, which can make your site run faster.

Timesheet does automatically run "VACUUM" (without the FULL) on a nightly basis when running under Postgresql. In addition to this, we recommend you do a "VACUUM FULL" on a weekly basis. When you add "FULL" to the VACUUM command, it will take much longer to run, but it will be more aggressive in cleaning out old data and compacting your database. Note that new installs of Timesheet version 5.6m2 or higher will automatically do a VACUUM FULL nightly at 3:30 am local time by default.

Please note that the VACUUM FULL command can take a long time to run (up to an hour on a very large and busy database) and it will prevent anyone from making changes to the database while it runs. (The regular VACUUM without the FULL does not lock anyone out.) Therefore, we recommend you run VACUUM FULL on a weekend or during non-business hours.

We also recommend that you add the word ANALYZE to the end of the command as illustrated below, but this is optional. It will cause the database to collect additional statistics that can help query performance.

If you are using the Timesheet for Unix internal database, you can type this simple command:

```
$ psql -c "VACUUM FULL ANALYZE" template1
```

(Make sure you 'source the setup file' first as described above in the "Running backupdb" section.)

There is no need to restart the Timesheet service after you perform a VACUUM (with or without the FULL.)

## Native Backup Formats

In this section we describe briefly how to perform "native" database backups for the two types of "Internal Database" provided by Journyx. Please see the "Backup Techniques" chart above to determine whether making native backups is the right strategy for your organization.

On Windows, the Timesheet Internal Database is provided by MSDE, which is based on Microsoft SQL Server 2000. On Unix, the internal database is provided by the Postgresql database system. In this section we describe how to do backup and restores on both of these systems. Journyx also supports external Oracle and IBM DB2 databases, but the recovery tools for those systems are not described here. Please consult with your DBA (database administrator) if you have Oracle or DB2.

### MSDE (Windows Internal Database)

The Timesheet internal database on the Windows platform is called MSDE, and it is based on Microsoft SQL Server 2000. If your organization already has a (full) copy of SQL Server 2000 you can use the Enterprise Manager to perform backup and restore actions on your Timesheet MSDE database. In that case please consult the documentation that came with your edition of SQL Server. If you do not have Enterprise Manager available, you must run a command line utility called OSQL and type in several commands. MSDE does not come with any GUI (graphical user interface) based tools.

The general idea here is that you must first create a backup device (pointing to a file on disk) and then you can backup your database to the device.

First get to a command line:

Click Start -> Programs -> Journyx Timesheet -> Timesheet Command Line Prompt.

At the prompt in the command window, type this command as one line:

```
c:\Program Files\Journyx> "c:\Program Files\Microsoft SQL
Server\80\Tools\Binn\OSQL.EXE" -U sa -S GANDALF\JOURNYXMSDE
```

You must replace GANDALF with the name of your local machine whatever that is.

It will then prompt you for the sa (administrator) password. If you are a Journyx customer with an up to date support contract then you can obtain this password from Journyx Support. Please note that it is possible to change the sa and journyx account passwords at any time with the changedbpw utility. Run the utility (in the bin directory of Timesheet) without any options to show its help screen.

First, create a backup device:

```
1> sp_addumpdevice 'disk', 'jxDump', 'C:\Program
Files\Journyx\MyTimesheetDatabaseSQLBackup.data'
2> go
```

That creates a backup device named 'jxDump' at the given filename. You can change the output filename to whatever you like. This command actually performs the backup operation:

```
1> backup database journyx to jxDump
2> go
```

That may take a few moments to complete depending on the size your database. It should say:

```
BACKUP DATABASE successfully processed (plus some additional statistics.)
```

You may now quit the OSQL program by typing "quit":

```
1> quit
```

## Journyx Timesheet Backup, Restore and Archive Guide

If at some other time you need to restore that file, start OSQL again and run this command:

```
1> restore database journyx from jxDump with replace
2> go
```

Usually you can only restore the backup onto the same machine that created it. It is possible to move a SQL Server backup to a different machine, but it involves a great deal of extra work.

If you get this error: "Exclusive access could not be obtained because the database is in use"; be sure to stop the "Journyx Timesheet" service in the Services control panel (usually under Control Panel -> Administrative Tools) before running the restore command.

### **Postgresql (Unix Internal Database)**

It is quite simple to make a "native" backup with Postgresql (PG), whether it is the version provided by Journyx or your own external PG site.

The **pg\_dump** command creates the backup. Normally the data is dumped to the console (screen) so you will want to redirect it to a backup file. In this example, assume that your database is named `timesheet` and you wish to backup to a file named `my_dump.sql`. (Please note that when the Internal Timesheet Postgresql instance is used, the database is named "template1", so you would use `template1` in this example instead of `timesheet`.)

This command creates the backup from the 'timesheet' database:

```
$ pg_dump -c -O timesheet > my_dump.sql
```

The `-c` makes a 'clean' backup including commands to drop old data. The `-O` option removes connection information and is usually necessary to make it easier to move data between different Unix accounts.

This command restores the backup to 'timesheet':

```
$ psql timesheet < my_dump.sql
```

After that completes, simply restart your timesheet site (`wstop; wstart`) to access the newly restored data.

**(End of Document)**