

Chapter 19



In This Chapter

- 19.1 Toolbox Overview 372
- 19.2 ModelDesigner 373
- 19.3 CatWalk 375
- 19.4 WebConsole 377
- 19.5 Widget Browser 378
- 19.6 Admi18n and System Info 379
- 19.7 The `tg-admin` Command 380
- 19.8 Other TurboGears Tools 380
- 19.9 Summary 381

TurboGears includes a number of nice features to make your life as a developer just a little bit easier. The TurboGears Toolbox provides tools for creating and charting your database model, adding data to your database with a web based GUI while you are still in development, debugging system problems, browsing all of the installed widgets, and internationalizing your application.

19.1 Toolbox Overview

The TurboGears Toolbox is started with the `tg-admin toolbox` command. Your browser should automatically pop up when you start the Toolbox, but if it doesn't you should still be able to browse to `http://localhost:7654`, where you'll see a web page with links for each of the tools in the toolbox (as seen in Figure 19.1).

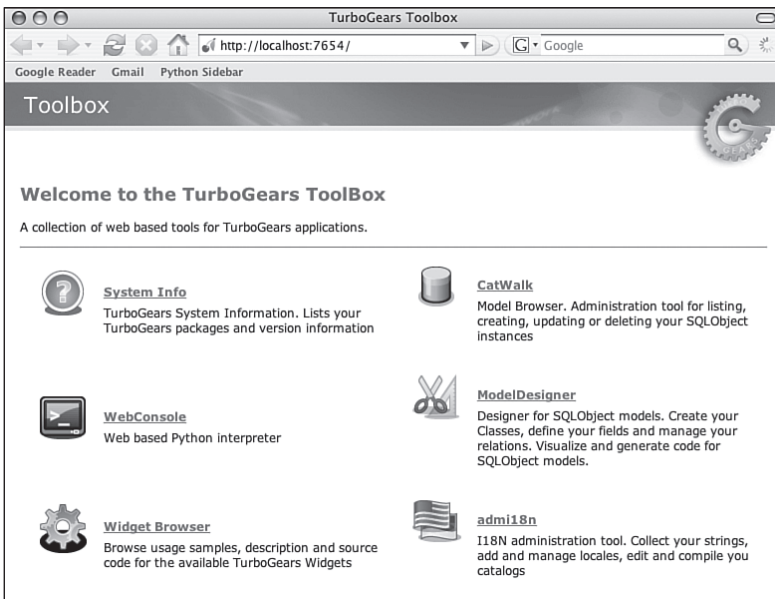


FIGURE 19.1 The TurboGears Toolbox home page

Each of the components in the Toolbox is also a TurboGears application, so you can also look at them as examples of how TurboGears applications are built.

Because there isn't anything in TurboGears that can't be done in code or from the command line, the use of the Toolbox is entirely optional.

19.2 ModelDesigner

The first tool we'll look at is the ModelDesigner. This tool is designed to help you create your SQLAlchemy model classes. It provides a GUI interface that allows you to create new classes by filling out simple web forms. In Figure 19.2 you can see a `Task` class with several columns defined for it.

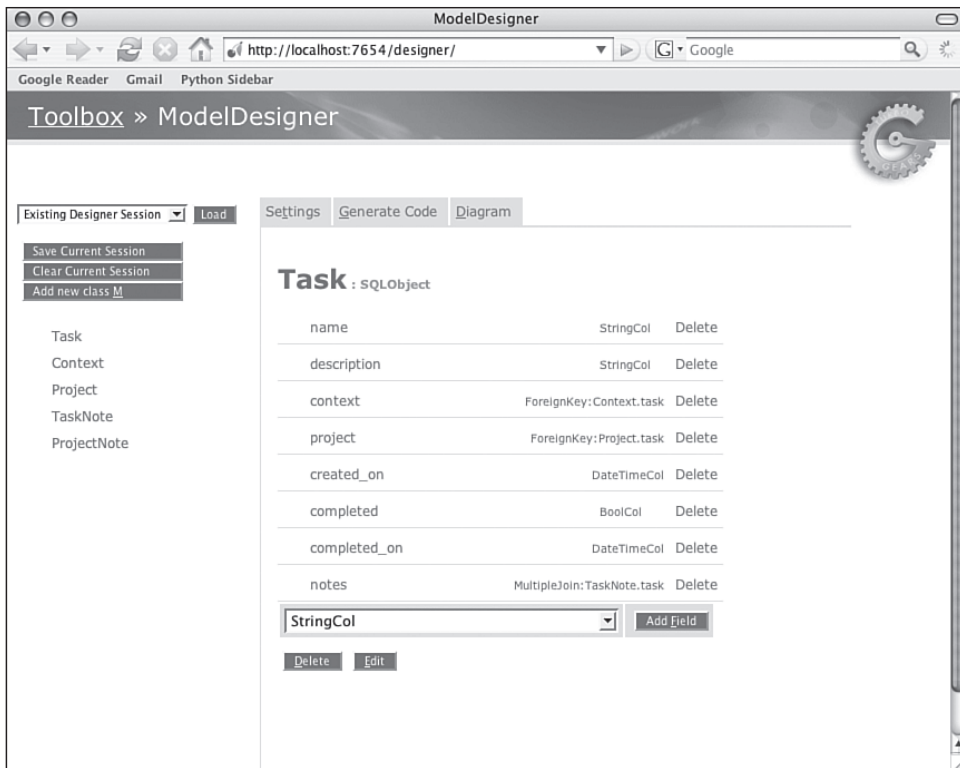


FIGURE 19.1 Default Widget Attributes

The main advantages of using the ModelDesigner are:

- You get a drop-down list of options for most fields so you don't have to remember things like all the possible `column` types.

- ModelDesigner gives you a form with all the column options so you don't have to remember those either.
- ModelDesigner creates both ends of joins at once so you don't have to make sure everything matches up manually.
- ModelDesigner provides you with a graphical representation of your model “for free,” as you can see in Figure 19.3.

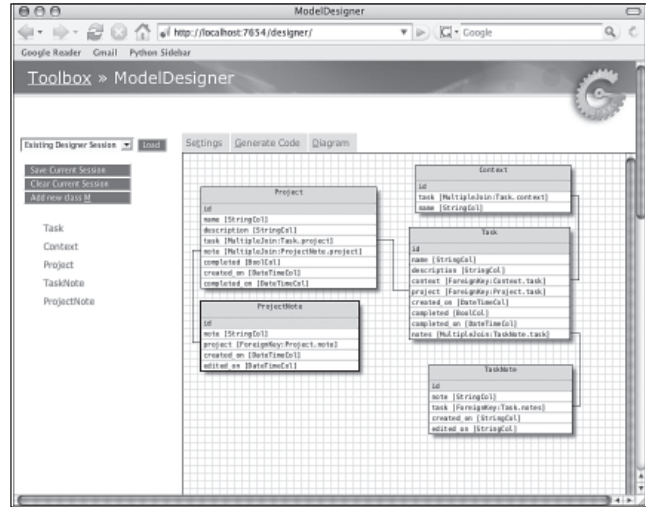


FIGURE 19.3 ModelDesigner diagram

All of that sounds so good that you might think you always want to use ModelDesigner, but there are a couple of limitations to remember when using the ModelDesigner:

- You can't start out defining your model classes by hand and then move to ModelDesigner when things get more complicated. ModelDesigner can write a new `model.py` file, but it doesn't parse existing files. This also means you won't be able to use ModelDesigner with the automatically generated Identity tables that `tg-admin` can give you.
- You have to remember to save your ModelDesigner session or you'll have to start all over again if you need to add a column to a table.
- ModelDesigner has no way of tracking custom methods that you might add to your table classes, which means that you'll have to re-add those methods whenever you update your database schema with ModelDesigner.

With all of that said, using the ModelDesigner to rough out your schema at the beginning of the project can help you get started much more quickly. You can always move over to the old-fashioned method of defining your tables by hand whenever the limitations of ModelDesigner start feeling painful.

Using ModelDesigner is pretty simple, especially if you know how `SQLObject` works and have created a few tables by hand. About the only thing that trips people up is the load/save interface. You can save your project with whatever name you want, but it will always show up as “Existing TurboGears Session.” Once you've defined

your model, you can write it out to `model.py` from the “Generate Code” tab (as seen in Figure 19.4).

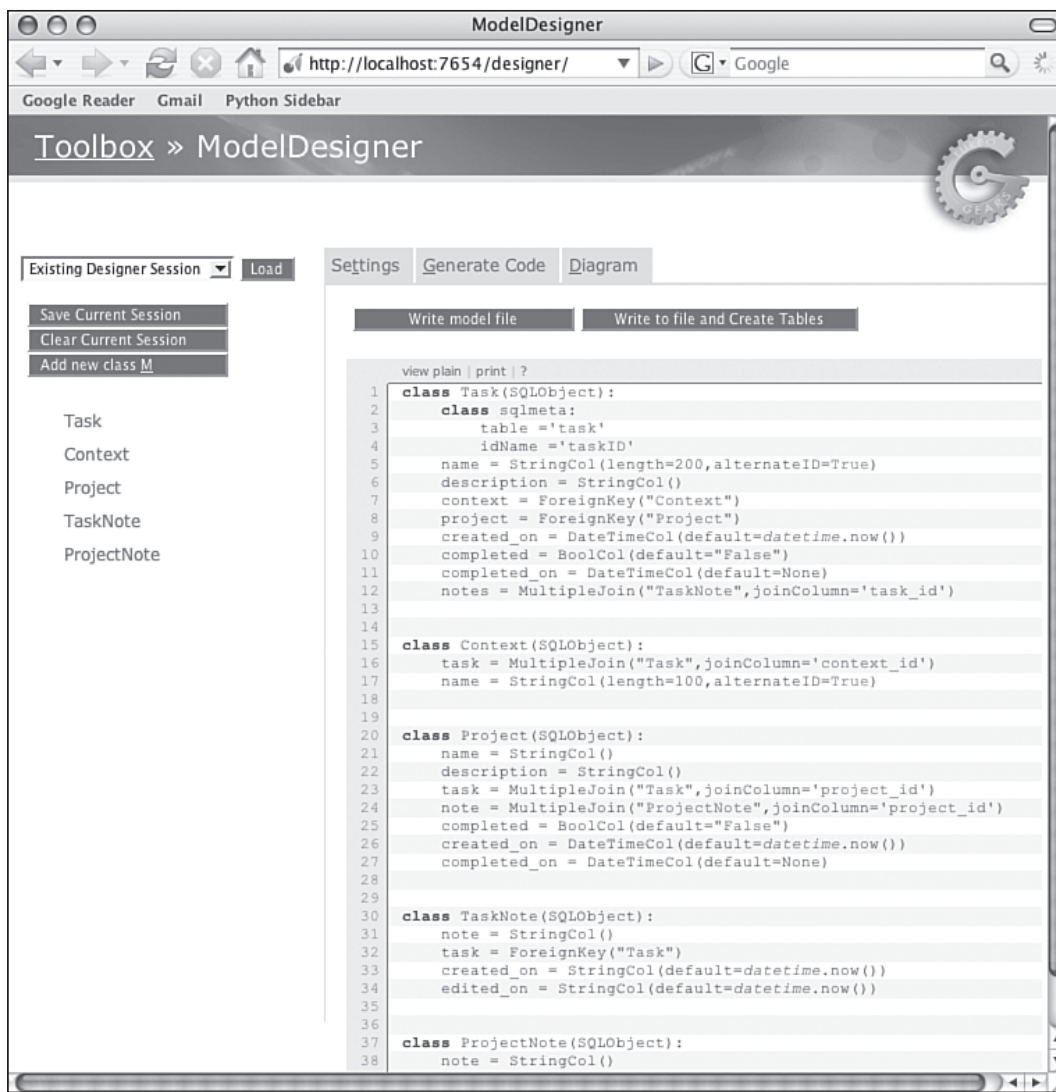


FIGURE 19.4 ModelDesigner “Generate Code” tab

19.3 CatWalk

Once you’ve created your database schema, you generally need at least a couple records worth of data in order to be able to start writing code and verifying that it works.

That's where CatWalk comes in. CatWalk provides an automatically generated GUI for editing sample data during development. The basic editing form, shown in Figure 19.5, is very easy to use for smaller data sets. If this database had a thousand projects in it, the drop-down list would be unmanageable. But, for adding a bit of data into your database during development, it couldn't get much easier than this.

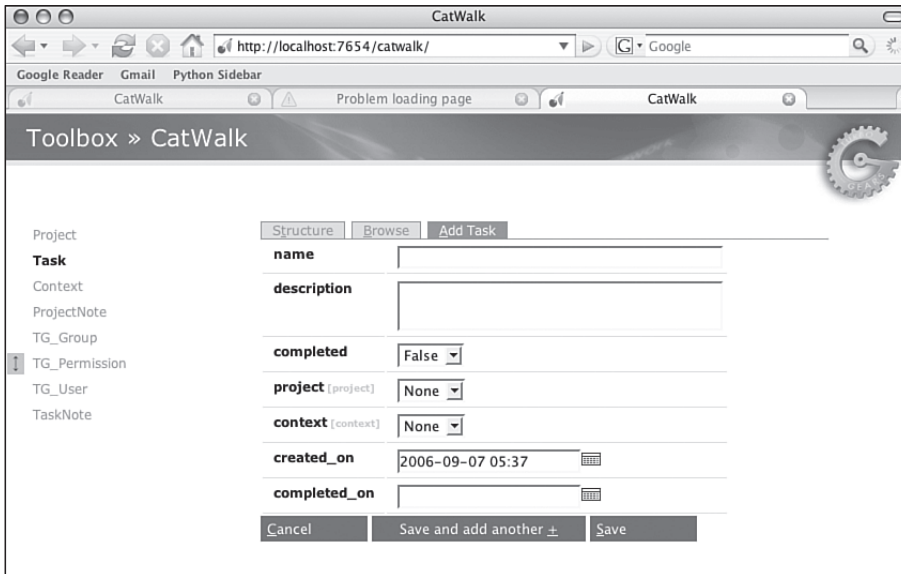


FIGURE 19.5 CatWalk editing

The one area that people sometimes get confused about with CatWalk is adding many-to-many join relationships. All you have to do is add the rows to the two tables, edit either of the rows you want to connect, and click the triangle next to the join “column.” Then a selection form that lets you manage relationships will appear and you can set up new relationships to your heart’s content.

In Figure 19.6, you can see another common use of the CatWalk tool—browsing data.

There are any number of ways you can customize CatWalk. You can have it display only some fields; you can change the way fields are ordered on your form; and you can even edit what appears in the drop-down lists so that you get actual project names rather than non-intuitive id values.

To do this all you have to do is click on the little edit icon in the browse view, and you’ll get something like what appears in Figure 19.7. You can uncheck columns to have them not be displayed, or drag-and-drop them to change their order.

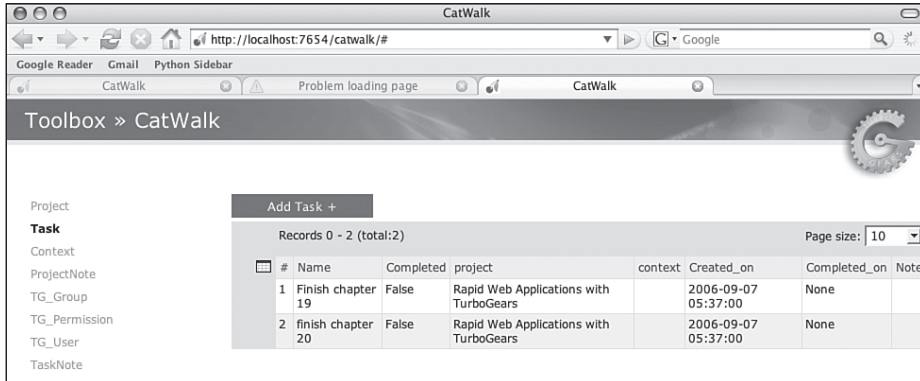


FIGURE 19.6 Browsing database data

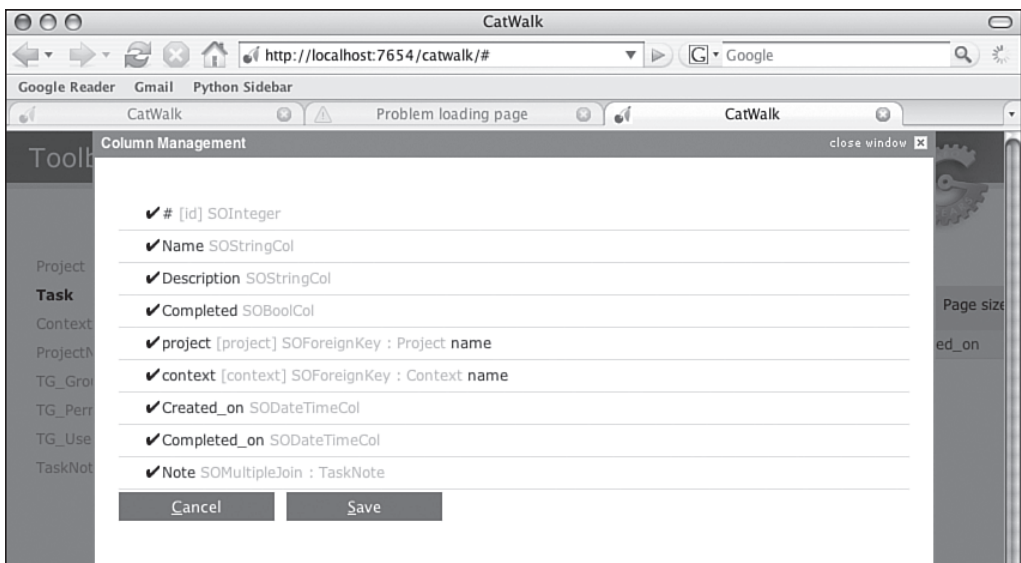


FIGURE 19.7 Customizing CatWalk

19.4 WebConsole

TurboGears also provides a web-based version of the interactive Python prompt. You can play with your model objects, add data, and even run adhoc tests from the WebConsole.

The Console also imports your model and sets up your database connection automatically so you can just get down to work. Admittedly, there's nothing you can do in the Console that you couldn't do from the `tg-admin` shell.

As you can see in Figure 19.8, the Console gives you the opportunity to create multi-line Python statements, or quick one liners.

19.5 Widget Browser

The Widget Browser is one of the most useful tools in the TurboGears Toolbox. We mentioned it in Chapter 16, “TurboGears Widgets: Bringing CSS, XHTML, and JavaScript Together in Reusable Components;” let’s take a little bit more time to look at it in depth here. The Widget Browser makes use of a special feature of `setuptools`, which allows eggs to define an



FIGURE 19.8 WebConsole

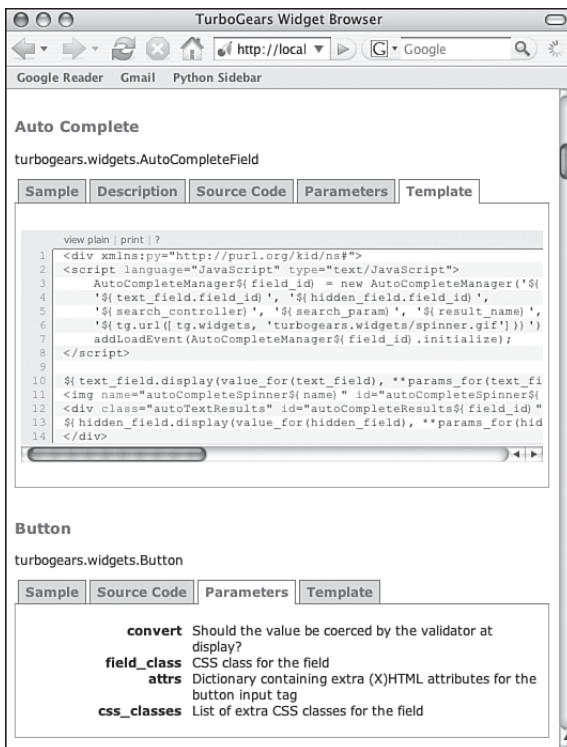


FIGURE 19.9 Widget Browser

entry-point. Every widget package in the Cog Bin (www.turbogears.org/cogbin) has a specially-defined entry point. This allows the Widget Browser to dynamically grab information for every widget you have installed. The information is always up-to-date, and everything works.

There is one exception to this rule. Some third-party widgets do not define a widget description class and will not show up in the Widget Browser. If you’re wondering what a widget description does, it isn’t at all complicated. It defines a default instantiation of the widget, which is used to display the widget in the Toolbox.

As you can see in Figure 19.9, each widget is displayed with five tabs:

- A sample of what the widget looks like in use

- A description of the Widget (taken from the widget’s docstring)
- A listing of the widget’s source code, which is invaluable when you want to know exactly how something is done
- A listing of each of the widget’s parameters and a brief description of what that parameter is for, which is taken from a `params_doc` dictionary in the widget itself so the description will always be up-to-date

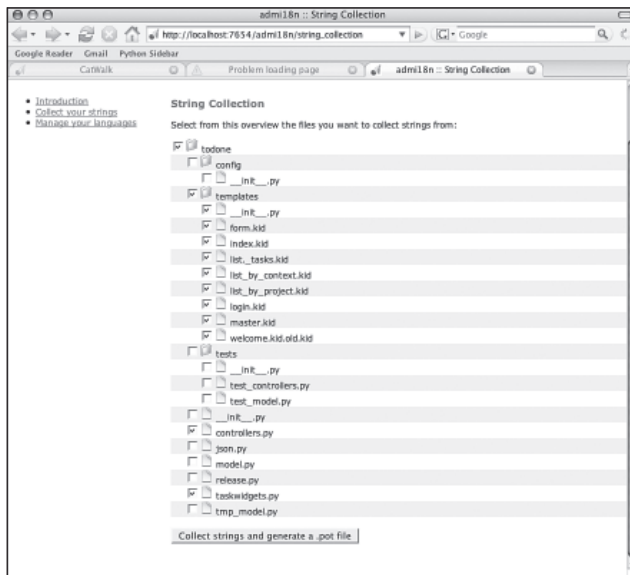


FIGURE 19.10 Admi18n set up

- A listing of that particular widget’s template code that can be cut and pasted by using the “view plain” link, allowing you to customize the way the widget is displayed

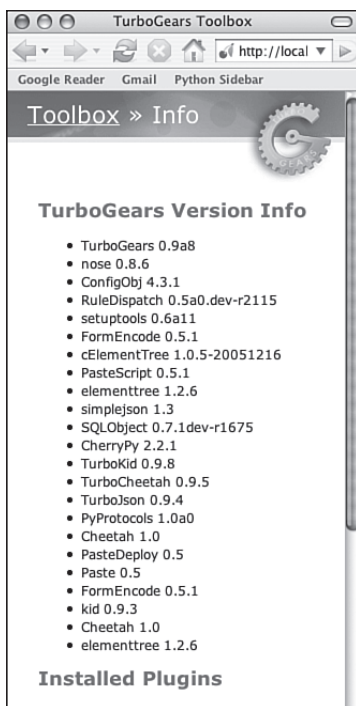


FIGURE 19.11 System Info

19.6 Admi18n and System Info

We’ll cover the Admi18n tool (as seen in figure 19.10) for translation administration in depth in Chapter 20, “Internationalization.” For now, we’ll just mention that TurboGears includes a full-fledged tool for creating and maintaining translation files so that you can easily create multilingual TurboGears applications. For a few applications, where you don’t have any strings coming from Python, you can actually just use this interface. For most applications, you’re going to have to do a little bit of prep work before you can get started with translation. While it’s not hard, there’s enough going on under the covers that it’s worth a whole chapter to explore.

Unlike the rest of the Toolbox applications, there's not much to do with the System Info tool (as shown in Figure 19.11). It just displays a single dynamically-generated page that displays which version of TurboGears and all its associated dependencies are installed on your system. This information is very useful if you ever have to submit a bug report on the TurboGears bug tracker (trac.turbogears.org), or when asking for help with a framework problem on the TurboGears mailing list (groups.google.org/group/turbogears).

19.7 The `tg-admin` Command

We've already seen the `tg-admin` command, but there's a lot more that it can do than just quickstart a new project, and even the `quickstart` option has some features we haven't touched upon yet. The core options of `tg-admin` are:

- `quickstart`—Creates a new directory based on the project and package names provided; has switches to use SQLAlchemy (`-s`) or use alternate project templates (`-t tgbig`)
- `shell`—Opens a new interactive shell with our model imported (When using SQLAlchemy your database changes should autocommit)
- `i18n`—Internationalization is covered in depth in Chapter 20
- `sql`—Wraps around the SQLAlchemy `sql-admin` command with your connection parameters pre-defined based on the information in your project's `config` file
- `toolbox`—Starts up the web-based Toolbox (You can use `-n` to not automatically open a browser window and `-c` to allow other specific IP addresses beyond the default localhost to connect to the Toolbox instance)
- `info`—Lists current versions of all installed TurboGears components

19.8 Other TurboGears Tools

TurboGears includes a number of optional tools for you to use if your application has specific requirements. There are also third-party tools like TurboMail. We can't cover all of these here, and new tools are popping up every day. But here's a list of a few tools that might save you some time if you need them:

- **Scheduler**—TurboGears includes support for a cron-like scheduler based on Kronos. You can find more information about the scheduler at <http://docs.turbogears.org/1.0/Scheduler>.
- **Configuration**—TurboGears uses `ConfigObj` to parse configuration files. You can find out more about how this works at <http://docs.turbogears.org/1.0/Configuration>.
- **logging**—TurboGears reuses the `logger` module from Python’s standard library. You can learn more about TurboGears logging at http://docs.turbogears.org/The_Logging_system, and more about the logger at: http://docs.turbogears.org/The_Logging_system.
- **paginate**—When you need to paginate tabular results, TurboGears provides a simple decorator-based pagination mechanism that you can read more about at <http://docs.turbogears.org/PaginateDecorator>.

19.9 Summary

- TurboGears provides several graphical tools to help you get started more quickly, and to maximize your productivity.
- The `ModelDesigner` provides a quick and easy way to define your database structure—and it even gives you free database diagrams.
- `CatWalk` makes it easy to add sample data to your database before you start work on your user interface!
- The `WebConsole` and `SystemInfo` tools make it easy to test your application through a web interface, and get a list of the versions of all the libraries included in your TurboGears install.
- The `Widget Browser` gives you an up-to-date look at all the widgets installed on your system, along with descriptions, documentation, and sample code.
- The `tg-admin` command gives you access to a variety of features like `quckstart`, `sql create`, and `shell`, that we’ve been using throughout the book to start new projects, manipulate the database, and create database tables.

