Journyx WebTime: Proving Python's Commercial Readiness

Andrew Reutter
journyx

Introduction

journyx WebTime (JWT) is a commercial, Web-based, time tracking product written in Python. It works with multiple operating systems, databases, and web browsers. It has user access control, customizable graphical interface, and both matrix and item reporting.

JWT is a prime example of the rapid development possible using Python. The typical release cycle is three months, with minimal development staff. Defect reports are few, and Python's effectiveness often allows patch delivery the same day a defect was reported.

1. Built on Free Software

JWT relies on free tools. Python, in particular, provides the brains and glue of the entire system. Other free software components include the Apache webserver, the PostgreSQL database, Robin Friedrich's HTMLgen, and D'Arcy J.M. Cain's PyGreSQL.

2. Architecture

The code base is split into libraries and interfaces.

2.1. WebTime libraries

These modules provide all necessary functionality for JWT, excepting specific user interfaces. They fall into the following distinct categories.

2.1.1. WebTime Database API

Wtdb provides homogenous access to PostgreSQL, MS SQL Server, JetDB, and most ODBC databases, all through the configurable layering of modules with identical APIs. Caching modules implement the same API as DB modules, and can be inserted or removed dynamically. This architecture greatly eases the expansion of our growing list of supported databases.

Our database API and its implementation modules are undergoing preparation for release to the public domain. Keep your eye on comp.lang.python.announce!

2.1.2. One stop configuration data

Configuration data is stored in both database tables for global and user options, and a configuration file with data for which the database is unsuited. Developers can ignore the source of this information. The wtconfig module exposes it all as a read/write dictionary, retrieving from and storing to the appropriate location.

2.1.3. Operating System independence

The wtos module is our extension of the os module, affording the developer OS invisibility. It provides such information as the username, hostname, and OS descriptors. Internally, these functions call their OS-specific versions based on the contents of os.name.

2.1.4. Security

Users are granted different levels of authority and are assigned to groups. Wtacl is our mechanism for tracking this data, allowing the developer to use the DB API without regard for the current user's access level.

2.1.5. Web based user interface

Wtdoc extends the functionality of many HTMLgen classes. The WtDocument class provides context-sensitive help screen access and button bars. The WtTable class prettifies the HTML table, alternating row colors to improve readability and automating text coloring based on the background color of a cell.

2.2. WebTime interfaces

A small Python module provides the gateway between Apache's CGI mechanism and the GUI interface pyc files shipped with JWT. Many of these user screens are highly customizable in appearance.

Python modules, wrapped by simple shell scripts, implement command line access to such common operations as adding database records, generating reports, and creating new users. This allows for external scripting of JWT. Use your OWN interface.