

Улучшение прототипа

Окошки и сценарии использования

Окошки:

- 1) connect to server
- 2) ftp browser
- 3) edit questions
- 4) choose encoding
- 5) license key
- 6) install ATC
- 7) about
- 8) protect script

Для перехода с (1) к (2) используется wizard-style. Чтобы снова получить (1), надо перезапустить программу.

На (2) wizard-style заканчивается. Окошко (2) становится главным окном программы.

(2), (3) и (4) — табы (вкладки).

Примерная последовательность действий:

- Вася запускает программу. Получает окно «connect to server». Вводит что надо, нажимает «connect».
- Программа видит, что на сервере нет ATC, поэтому выводит окно «install ATC». Чтобы продвинуться дальше, Вася должен установить ATC.
- Появляется ftp-браузер. Вася находит скрипт, который надо защитить и кликает на нём. Говорит, что сделал бекап (врёт, наверное), и программа защищает файл.

Через месяц Вася решает поменять вопросы.

- Он запускает программу, проходит окно «connect to server».
- Программа видит, что ATC уже установлен, поэтому считывает настройки и вопросы из файла bbas_config.php на сервере.
- Появляется ftp-браузер, но Вася переключается на редактирование вопросов. Изменяет их.
- Он пытается сохранить изменения, но программа требует вначале

зарегистрироваться.

- Поэтому Вася находит keygen на astalavista.box.sk, подцепляет очередной троян, но тем не менее получает свой ключ, который и вводит в табе «license key».
- Наконец-то, Вася сохраняет изменения. Перед тем, как завершить программу, он смотрит в окошко «about».

Разбор файла „bbas_config.php“.

Возможно, самая важная и сложная функциональность. Нужно много юнит-тестов. И описать, как я могу добавлять свои собственные тесты.

Большое пожелание: код не должен быть завязан на остальную программу и на библиотеки wxPython. Я хочу использовать его и в других программах.

Формат файла:

```
...что-то...
$questions = array(
...вопросы...
);
...что-то...
    'CHARSET' => 'кодировка',
...что-то...
$license_key = 'ключ';
...что-то...
```

Необходимо и достаточно, чтобы программа всегда могла распарсить то, что сама сохранила. Если программа подозревает, что распарсила файл неправильно или сталкивается с сюрпризом, пусть она сообщает об этом и завершает работу.

Нельзя портить текст bbas_config.php вокруг изменяемых данных.

Надо учитывать кодировку при считывании и записи файла bbas_config.php.

Вопросы хранятся в виде обычного PHP-шного ассоциативного массива массивов. Например:

```
$questions = array(
    'Are you human?' => array ('yes', 'ja', 'oui'),
    'Say "Hello"'    => array ('hello', 'hi', 'hallo')
);
```

Строки только single quoted. Экранирование специальных символов описано тут: <http://www.php.net/manual/en/language.types.string.php>.

Строки могут быть как с html-разметкой, так и html-safe. Признак html-разметки: строка начинается с «<tag» и заканчивается на «</tag>». Если это не так, то строка должна быть html-safe.

Строки могут содержать численные entities, например: «é» (десятичный код) или «ï» (шестнадцатеричный).

Интерпретация текста вопросов и ответов при загрузке из конфига:

- Раскодировать escape-последовательности php-строк
- Численные entities превратить в символы
- Если строка html-safe, то раскодировать её

Обратный процесс при сохранении:

- Если строка html-safe, то закодировать её
- Если надо, заменить некоторые символы их численными entities
- Заменить спецсимволы php на их escape-последовательности

Когда надо заменять символы на их численные entities:

- Если код символа меньше кода пробела (кажется, 32)
- Символ отсутствует в выбранной кодировке (например, кодировка windows-1251, а вопрос содержит немецкую букву «ß»)

Лицензионный ключ

Лицензионный ключ завязан не на саму программу bbAntiSpam, а привязывается к серверу и хранится на сервере в файле bbas_config.php. Ключ считывается и интерпретируется:

- после установки связи с сервером и чтения bbas_config.php, и
- после ввода пользователем в окне регистрации.

Формат ключа: server_name/xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, где

- "x" -- это шестнадцатеричные буквы в количестве 32 штуки,
- server_name -- это имя сервера, для которого сгенерирован ключ.

Если ключ не подходит под этот формат, то ключ *неправильный*. Если ключ подходит под этот формат, то он *условно правильный*. Если к тому же `server_name` и имя сервера, к которому приконнектилась программа, совпадают, то ключ *правильный*. В целях usability, совпадение определяется так:

- если есть, откидываем префикс "www." у `server_name` (назовём результат "xname")
- имя сервера, к которому приконнектились, должно заканчиваться на `xname`.

Если ключ неправильный, то программа ограничивает свою функциональность: не даёт сохранять вопросы. Если ключ правильный или условно правильный, то программа полностью функциональна.

Единственная разница между правильным и условно правильным ключами — это реакция программы на ввод ключа в окне регистрации. Если ключ правилен только условно, программа должна то предупредить пользователя об этом: `The key you've entered seems invalid: FTP server name and the key server name are different. Using an invalid key completely disables spam protection. Continue anyway?` Если пользователь нажимает на "No", то попытка ввода ключа отменяется, и программа должна использовать старый ключ.

Что такое установка Advanced Textual Confirmation

Установка АТС — это копирование файлов «`bbantispam.php`» и «`bbas_config.php`» на сервер в директорию «`bbantispam_dir`». Эта директория вычисляется так:

- Если файл настроек содержит непустую переменную «`bbantispam_dir`», то использовать значение этой переменной.
- Иначе это поддиректория «`bbantispam`» в домашней директории ftp-пользователя. Путь должен быть абсолютным, не относительным. Насколько я знаю, абсолютный путь домашней директории выдаёт команда «`pwd`», если её дать сразу после захода на ftp-сервер.

Если файлы АТС уже есть на сервере, но каким-то образом программа дошла до режима установки, то файлы надо перезаписать.

Установлен ли Advanced Textual Confirmation?

Простая проверка: в директории «`bbantispam_dir`» (см. Что такое установка Advanced Textual Confirmation) должны присутствовать оба файла

«bbantispam.php» и «bbas_config.php», и они должны быть ненулевого размера.

Что такое подключение Advanced Textual Confirmation к скрипту

В самое начало файла скрипта надо вставить строчку:

```
<?php include_once('bbantispam_dir/bbantispam.php'); ?>
```

Вместо «bbantispam_dir» должна быть директория, в которую установлен АТС (абсолютный путь, см. Что такое установка Advanced Textual Confirmation).

В скриптах могут быть разные концы строк: «\r», «\n» или «\r\n». Программа должна определять, какой тип используется в скрипте, и использовать его для строки подключения АТС.

Надо проверять, подключен ли уже АТС, и если да, то второй раз подключать не надо. Вместо этого надо сообщить пользователю, что защита уже подключена.

Защищать надо только PHP-файлы. Признак PHP-файла: наличие обоих сочетаний символов «<?» и «?>». Если их нет, то вместо подключения АТС сообщить пользователю, что он собирается защищать не скрипт.

Надо отказываться защищать системные файлы «bbantispam.php» и «bbas_config.php».

Файл настроек

Файл настроек лежит в той же директории, что и программа. Называется «bbas.ini». Формат простейший, файл состоит из строк вида «параметр=значение».

Программа только считывает файл настроек, но ничего туда не сохраняет. Настройки:

ftp_user и ftp_password не нужны. Но если удобно их иметь, то можно оставить.

ftp_proxy, ftp_proxy_user и ftp_proxy_password — параметры FTP-прокси.

ftp_mode (active или passive) — режим использования FTP.

bbantispam_dir — директория, куда установлен скрипт Advanced Textual Confirmation.

Окошко «connect to server»

Сообщать о том, что программа соединилась с сервером, не надо. Сразу открывать ftp-браузер.

Help ID: connect.

Окошко «ftp browser»

Не писать отладочную информацию. Или лучше писать только в режиме «debug» -- тогда надо сделать ещё один параметр в ini-файле.

По умолчанию должна открываться домашняя директория пользователя, а не корневая директория.

Добавить текст: «to protect a file, double-click or right-click on it».

Должен работать right-click.

Help ID: browser.

Окошко «edit questions»

Help ID: edit.

Окошко «choose encoding»

Нужна кнопка «Help». Help ID: encoding.

Нужен полный список кодировок. Вроде бы это как-то должно делаться через модуль питона «codecs». Идеально было бы сделать выбор кодировок как в firefox'e (internet explorer'e) — с категориями.

Окошко «license key»

Help ID: register.

В окне писать статус регистрации: зарегистрирован, условно зарегистрирован, не зарегистрирован.

Нужна кнопка или ссылка «buy a license». В веб-браузере открывается страница «<http://bbantispam.com/goregister/>».

Нужна кнопка «Paste a key». По нажатии на неё, программа ищет ключ в буфере обмена (clipboard) и вставляет в поле ввода ключа. Как искать ключ: взять текст из буфера обмена и найти первую строку, которая подходит под формат ключа.

Кнопка «Save license key». Реакция программы на ввод ключа в окне

регистрации описана в «Лицензионный ключ». Дополнительно, содержимое окна должно обновиться: например, надо исправить статус регистрации.

Окошко «about»

Добавить ссылку на сайт: «<http://bbantispam.com/>».

Писать статус регистрации программы. Если не зарегистрирована, то вывести кнопку «Register». По её нажатию открыть страницу «<http://bbantispam.com/goregister/>»..

Окошко «protect script»

Нужна кнопка «Help». Help ID: mount.

Окошко «install ATC»

Нужна кнопка «Help». Help ID: install.

В диалоге «Choose language» языком по умолчанию должен быть английский.

Хороший вопрос, откуда брать список языков и соответствующие файлы «bbas_config.php». Думаю, проще всего так: в директории программы помимо файла «bbas.ini» есть поддиректория «lang». В ней куча поддиректорий: для каждой из них имя есть название языка, а внутри лежит файл «bbas_config.php».

Файл справки

Файл справки лежит в инете по адресу <http://bbantispam.com/onlinehelp/index.html>. Когда пользователь нажимает «Help», программа открывает этот адрес в обозревателе.

Но если у активного окна или кнопки есть Help ID, то надо открывать другую страницу: <http://bbantispam.com/onlinehelp/HelpID.html>, где вместо подстроки «HelpId» подставлено значение Help ID.

Строковые ресурсы

Все сообщения программы должны быть обёрнуты в механизм gettext: вместо "строка"

должно быть

_("строка")

Насколько я понял, так уже сделано.

Py2exe

Нужно разобраться, как превратить программу в обычный исполняемый файл. Наверное, следует использовать py2exe.

Мне нужен файл проекта для py2exe, или что-то, что позволит пересоздавать exe-файлы самостоятельно.

Для исполняемого файла нужны иконки, тексты и прочие добавки. Используй заглушки, а мне пришли список этих заглушек с указанием, как их менять.

Меню

Линейка меню содержит «File» и «Help».

В меню «File» есть только «Exit».

В меню «Help» есть «Help» и «About».

Особенности работы с ftp-сервером

Раз в 30 секунд давать какую-нибудь безобидную команду типа «cwd», чтобы сервер не разорвал соединение по таймауту.

В процессе работы, если соединение с сервером потерялось, попробовать его восстановить. Если не получилось, сообщить об этом пользователю и выйти.

Если возможно, надо сделать безопасную перезапись файлов — чтобы не было больших проблем в случае неожиданных ошибок. Например, можно делать так: изменения писать не сразу в файл, а создавать новый файл и затем переименовывать его в старый.