

# NEARLY THERE FRONTKICK MANUAL

1.0.6  
April 11, 2005

---

# Table of Contents

1	Preface.....	4
2	Quick Start: For the Impatient.....	5
3	Pre-requirements.....	6
3.1	Hardware and Software Requirements.....	6
3.2	Frontkick is built on aMember.....	6
3.3	aMember License Key.....	6
3.4	IonCube encoding.....	6
3.5	Transferring the files to your server.....	7
3.6	Certain Directories must be Writable.....	7
3.7	Apache mod_rewrite.....	7
3.8	PHP Curl Extension is Required .....	7
4	Getting Around Frontkick.....	8
4.1	General Overview.....	8
4.2	Directory Structure.....	8
4.3	Customizing your site.....	10
	Introduction.....	10
	How do the /site and /performers areas work?.....	10
	Customizing templates.....	11
	Customizing PHP pages.....	15
4.4	Frontkick, Sidekick, Activeshows, private session & tokens management.....	17
	Overview.....	17
	Shows.....	17
	Sessions.....	17
4.5	Payments & Credits for private sessions.....	18
4.6	Images management and Frontkick.....	18
	Download & store.....	19
	Viewing images.....	19
4.7	Caching of the who is online page.....	19
5	Frontkick Admin Area Examples.....	21
5.1	Hooking up a payment system.....	21
	Enabling new payment plugin.....	21
	Configuring the products.....	21
5.2	Changing your purchase options.....	21
5.3	Mapping product to Frontkick credit.....	22
5.4	Frontkick/Sidekick services.....	22
5.5	Performer Accounts.....	22
	Adding performers.....	22
	Sync performers.....	22
	Status & expiration date.....	23
	Extending fields of a performer's account.....	23
6	Upgrading Frontkick.....	24
6.1	Overview.....	24
6.2	Keeping your modifications.....	24
7	Operations.....	25
7.1	Overview.....	25
7.2	Log & errors files.....	25
7.3	Cleaning up old records.....	25
7.4	Recovering after a server crash.....	25

7.5 Security Issues.....	26
8 Frequently Asked Questions (FAQ).....	27
I Some customers pay with a wire transfer. How is this entered?.....	27
II I launch a show and it gets immediately terminated!.....	27
III Can I use multiple payment systems?.....	27
IV Can Sidekick and Frontkick be on different servers?.....	27
V Does Frontkick support paid memberships?.....	27
VI I use multiple payment systems. How can I track all of them using Frontkick's tracking system?.....	27
VII How do I give free credit to a member?.....	27
VIII I want to email a single performer or member. Why is there no form for that?.....	27
IX I want to temporally disabled a performer, how do I do that?.....	27
X I don't like PHP. Do you have a Perl version of Frontkick?.....	27

# 1 Preface

---

Frontkick is a PHP/MySQL system which provides all the features needed to build a pay per view site. Frontkick allows your clients to register, buy time and apply their time to various shows. A sophisticated administrative area allows you to view reports, update the configuration and handle accounting. The following major components are part of Frontkick:

- User account management. Users can register and login. They will be able to buy time and Frontkick will manage their funds.
- Ecommerce. Support is built-in for over 30 billing systems which allow you to easily connect with your chosen billing solution.
- Administrative Area. You will be able to set up and configure Frontkick, view reports such as total sales . manage the user and performer account.
- Performer account management. Performers can apply, and login when approved by the admin. They have access to menu to personalize their show, setup their images gallery and to some reports showing their activity.
- Sidekick interoperability. Frontkick uses Sidekick as the video platform.

With Frontkick, you will be able to set up and configure it via easy web pages. You can then customize it to your needs, such as modifying html pages, adding images. After this, it will be ready for testing and going live.

## 2 Quick Start: For the Impatient

---

Here's our absolute minimum Quick Start Instructions which will bring up Frontkick on your Red Hat Linux server. Later on, be sure to peruse the other sections of this manual for advanced settings and features.

1. Install Sidekick. Frontkick uses Sidekick to manage videochat shows, so installing Sidekick first will make the installation more straight-forward.
2. Extract the files from Frontkick-1.0.6.tar.gz and move the three directories extracted under newly created docRoot directory (/docRoot/ms /docRoot/site and /docRoot/performers) into the public directory of your web server.
3. Make sure that for the installation time, /ms, /ms/data and /ms/data/new\_rewrite is writeable by the webserver.
4. Using phpMyAdmin, or any other tool your hosting solution provide you to manage database, add a new database to Mysql. We recommend it be named Frontkick.
5. In your browser, go to <http://www.myDomain.com/ms/setup.php> and follow the directions. You will be prompted for system configuration details.
6. In your browser, go to <http://www.myDomain.com/ms/adm> and login as an administrator. Click on **Verify Sidekick Connectivity** and see that all the tests pass. If any fail, you will need to configure Sidekick as instructed.
7. Click on **Sync Performers** and verify that your performers are defined in Frontkick and Sidekick. Frontkick and Sidekick must both have the same list of performers.
8. In the admin area, under Configuration , add your merchant billing service using the Plugins tab and modify any other settings to match your needs.

Now, you can go to Frontkick's home page by browsing to <http://www.yourDomain.com/site> for the user part or <http://www.yourdomain.com/performers> for the performer site.

# 3 Pre-requirements

---

Installing Frontkick is meant to be a easy, step-by-step process. The "Quick Start" chapter lays out the steps. Please review this chapter for further details.

## 3.1 Hardware and Software Requirements

Frontkick/Sidekick runs on a server with the following capabilities:

- Pentium 2G IV processor or better (The AMD equivalent is fine);
- At least 1G of memory;
- At least 1G hard disk space (More space may be required if the show archiving feature is enabled);
- Supporting software: Apache 2.0 with mode\_rewrite, Mysql 4.1, PHP 4.3

## 3.2 Frontkick is built on aMember

aMember (from <http://membership.cgi-central.net>) provides user management, reporting capabilities and ecommerce features. The version of aMember that Frontkick uses is aMember 2.3.4PRO. Some questions you have might really be aMember questions. In this case, please consult their manual, a copy of which is included in the Frontkick distribution.

## 3.3 aMember License Key

aMember requires a license key to run. Frontkick asks for this during the setup process, when you enter `www.myDomain.com/setup.php`. We provide this key to you and it can be found by logging in to your Nearly There client account at <http://www.nearlythere.net/clientManager>.

## 3.4 IonCube encoding

aMember is supplied in PHP source except for two files. These two files provide database access and allow aMember to validate its license key. These two files are encoded with the IonCube PHP Encoder (<http://www.ioncube.com/>). Frontkick comes with the decoder and it should work correctly on installation. If you should have any problems, IonCube has instructions which might help.

Frontkick includes the IonCube decoder files for Linux in a directory called `ms/ioncube`. (In fact, the IonCube decoder files for Windows are also included for completeness.) Normally, Frontkick is able to find these files and will load them automatically. If you have any troubles, you can try reviewing the included README file, `ms/ioncube/README.txt`. It will explain how you can run the included `ioncube-loader-helper.php` PHP script to determine the actual problem.

In particular, if you this an error like this, **The file `ms\rconfig-ioncube.inc.php` has been encoded with the ionCube PHP Encoder and requires the free ioncube\_loader ionCube PHP Loader to be installed**, then PHP doesn't know how to decode these files.

This means that the encoder can't be found. You might have to download the required dll from <http://www.ioncube.com> and add a line such as this to your `php.ini` file:

```
zend_extension_ts = /path_to_ioncube/ioncube_loader_file
```

and make sure that the following line is commented out:

```
;extension=eaccelerator.dll
```

## **3.5 Transferring the files to your server**

When transferring the files to your server, make sure you FTP them in binary mode, as the encrypted files will be corrupted if the transfer is in text mode.

## **3.6 Certain Directories must be Writable**

During setup, Frontkick will check that it can write to certain directories. If it can't, it will inform you of this and you must correct this before continuing. The following directories need to be writable:

- /ms
- /ms/data
- /ms/data/new\_rewrite
- /cache/whois

In order to correct this, you would probably do something like:

```
# chmod 0707 ms ms/data ms/data/new_rewrite
-or-
# chmod 0777 ms ms/data ms/data/new_rewrite
```

in your Linux shell for each directory.

After you have completed the setup, you should change /ms back to being non-writable. It only needs to be writable during setup so that a configuration file, `config.inc.php` can be written out. Make sure to type:

```
# chmod 0705 ms ms/data
-or-
# chmod 0755 ms ms/data
```

## **3.7 Apache mod\_rewrite**

Frontkick requires your Apache installation have the Apache module, `mod_rewrite`, enabled. `mod_rewrite` is a standard module that comes with Apache and can be enabled by ensuring that the following line in `conf/httpd.conf` is not commented:

```
LoadModule rewrite_module modules/mod_rewrite.so
```

## **3.8 PHP Curl Extension is Required**

If after installing Frontkick, you get an error like this: **Fatal error: Call to undefined function curl\_init() in ms\Frontkick\sk.inc.php on line 39**, this means the the PHP Curl extensions are not enabled. Look for a line like

```
;extension=php_curl.dll
```

in your `php.ini` file and uncomment it.

# 4 Getting Around Frontkick

---

## 4.1 General Overview

Frontkick works hand-in-hand with Sidekick. Frontkick provides the user-interface, account management and ecommerce system, whereas Sidekick provides the videochat system.

You can think of Frontkick as a layer on top of Sidekick. In technical language, Frontkick is the master and Sidekick is the slave. What this means is that Frontkick is controlling the entire process. For example, whenever a session is active, it was initiated by Frontkick and Frontkick tracks the session progress.

Frontkick is written in PHP and is built on a membership system called aMember (see <http://membership.cgi-central.net>). aMember is a very stable system which provides user management, reporting capabilities and ecommerce features.

While PHP can be not trivial for non-programmers, FK always display pages using the notion of templates, which present an easy way to customize your site using some pseudo HTML code, easy to modify.

There are three key areas to Frontkick:

- Site: what your prospects and clients will see. This is the public view.
- Performer: where your performers can go to sign up and see reports.
- Administrative: where you can go to manage your operation.

Each of these three areas is for a different kind of user and they are separated on your site by naming conventions:

- Use `/site` to access the Site area.
- Use `/performers` to access the Performer area.
- Use `/ms/adm` to access the Admin area. ("ms" stands for membership.)

When configuring your system, you will configure most of it in Frontkick through its Configuration screens. There are aspects that need to be configured in Sidekick, but Frontkick will tell you what they are! In Frontkick control panel (<http://www.yourdomain.com/ms/adm>), there is a "Sidekick check" which verifies that Sidekick is configured as required by Frontkick. Using this screen, you can easily fix any problems in Sidekick so the two communicate correctly.

## 4.2 Directory Structure

Frontkick uses the following directory structure:

- `/ms`: This is the key directory! It includes all of the aMember functionality plus modifications and extensions for Frontkick.

We have tried to limit file changes as much as possible. The following files have been extended to incorporate Frontkick's functionality: `amember.sql` (responsible for creating the additional tables used by Frontkick), `common.inc.php` (To support additional capabilities for Smarty templates) and `site.inc.php` (to extend aMember without effecting their files).

The `/ms` directory branch includes a number of subdirectories, each of which is briefly explained below:

- `/ms/data`: This directory is used by aMember as a test directory during setup, and as a working

directory to keep track of user and performer logins (also related is /ms/data/new\_rewrite, which should remain writable for proper operation).

- /ms/docs: Various aMember documents.
- /ms/ioncube: Contains the ioncube (<http://www.ioncube.com/>) PHP encryptor plugin. Ioncube is a system to protect PHP code by encrypting the PHP file. aMember use ioncube to protect only two files. The two files can be found by looking at /ms/rconfig.inc.php and /ms/plugins/db/mysql.inc.php. This means that all the other aMember files are completely readable. Of course, all the Frontkick files are also readable.
- /ms/smarty: Smarty is a template engine (<http://smarty.php.net/>) which processes html templates with special directives into html pages. Frontkick uses it to generate the web pages according to smarty templates that are defined in the *templates* directory.
- /ms/coresite: These PHP pages are used by the client site, and include the following: login, logout, profile, signup, myaccount and buy more time. These pages shouldn't be accessed using the /ms/coresite path, but rather through the /site path as explained on the 'Customizing your site' chapter.

The templates used to display all pages are contained in /site/templates.

The php pages are explained in the next section: Customizing your site.

- /ms/performers: These PHP pages are used by the performers area, and include the following: login, logout, profile, signup, myaccount and earnings reporting.

The templates used to display all pages are contained in /performers/templates.

The php pages are explained in the next section: Customizing your site.

- /ms/frontkick: Contains all the common php libraries, including new database functions, extensions to aMember, the Sidekick/PHP interface, the callout page used by SK to notify FK and the Snapshot page. You might note that it does not expose its templates since they are basically a programatic interface.. A more detailed list can be seen below:
  - callout.php : This page is called by SK whenever a significant Sidekick event occurs. It will handle events as simple as keeping the Frontkick list of who is online up to date, and also more complex events such as processing the accounting for private sessions. In addition, callout.php will also handle the caching of the whois page. Triggered by certain events, it will delete the current whois page so that it will be recreated the next time it is requested..
  - common.inc.php: Contains the PHP library where all the common functions are gathered. For example, it includes the debug() function which is used to add log lines to the debug files. Another example is various utilities to convert seconds to HH:MM:SS format. A few functions are highlighted here:
    - skcheck(): checks Sidekick connectivity. It will verify that Sidekick and Frontkick are correctly configured to interoperate with each other. It will also raise and explain any potential problems.
    - update\_time\_on\_payment(): updates a member account and the current private session when a new payment has been confirmed.
    - clear\_FK\_log(), clear\_FK\_inactivetoken() and restamp\_credential(): used to purge, reset or change files and contents based on daily or hourly criteria.
  - db.mysql.php: This script contains all Frontkick operations on the database. This script extends amember's core database operations to handle the new tables used by Frontkick, surfacing access for tokens, performers, images, profiles, shows, sessions and activeshows tables. All the included functions rely of access to the *GLOBAL\$db*; object, which is defined by *aMember*,

and will all start with **db\_** as prefix.

- `performers.inc.php`: This script, which is based on the original aMember `member.inc.php`, replicates “members” as a new structure and implements what is a performer in FK. Uses `db.mysql.php` to implement any database-related operation.
- `sk.inc.php`: This script represents Frontkick's direct interface to Sidekick. It allows Frontkick to request many kinds of information from Sidekick, such as retrieving a list of services. It also allows Frontkick to change Sidekick, such as removing a performer. In essence, it maps Sidekick's RemoteAccess HTTP requests as a PHP layer.
- `SnapshotUpload.php`: This script is used to upload and store in the database a new snapshot (or image) from the Sidekick's performer program.
- `/ms/adm`: This directory is responsible for the Frontkick Admin Area. This will be described further in the Frontkick Admin Area example chapter.
- `/ms/templates`: This directory contains templates that you shouldn't change! They are used by core functions of Frontkick, and by the Frontkick Admin Area.

That said, if you should change them, it will make upgrading to a newer version of Frontkick more involved.

The last two directories, `/site` and `/performers`, are described more in detail in the next paragraph.

## **4.3 Customizing your site**

### **Introduction**

Frontkick offers two levels of customization.

The first and recommended way is through a template-based system called Smarty (<http://smarty.php.net/>) that Frontkick uses to process and display pages. All the templates that used by Frontkick have an `.html` extension. These pages will look like html files that you are used to working with. They contain standard html tags and can be edited in any standard editor. In addition, they contain special Smarty directives which allow Frontkick to insert dynamic content when generating the page.

The second level of customization is through PHP. Frontkick is distributed with complete PHP code, to allow programmers to customize Frontkick behavior. This way is much more involved and requires a much higher skill set. Please note that Nearly There support is not included for this level of customization.

### **How do the /site and /performers areas work?**

Both customer and performer areas are built using the same approach. If you look at these directories, you will see very few files!! As shipped, no php pages are in the site directory – it contains only a `.htaccess` file (used by Apache's `mod_rewrite` module), a `shared.inc.php` file and a `/templates` directory.

When you request a page from this directory, let say for example `/site/whois.php`, the `.htaccess` file will intercept the request and will rewrite it on fly as necessary. First, it will set up variables so that Frontkick knows from where to retrieve the templates. Then it will test to see if the requested page is present in the current directory. If it exists, the request will be passed to it. If the page is not present in the current directory, it rewrite the request to look for the file in the `/ms/coresite` or `/ms/performers` directory.

Both core php implementation under `/ms` will then use the variables passed by the `.htaccess` file to register the templates directory and the site URL as part of the PHP session. Why? The customer and performer areas use different templates. By doing this, Frontkick can manage each area's templates separately.

Want to see something special about this architecture? It's easy to present multiple 'skins' of the same site, without duplicating PHP pages, which most of the time doesn't change. With the default installation for example, you can duplicate the /site directory to /newsite and modify the /newsite/templates to provide another look and feel of your site. This approach allows you to concentrate on the look and feel and not have to be concerned with PHP code.

## **Customizing templates**

For both areas, the templates are in the corresponding templates directory. They are written in Smarty (<http://smarty.php.net>). The following table lists out the different templates used and describes their usage:

<i>/site/templates</i>		
<i>Name</i>	<i>Description</i>	<i>Available Variables</i>
<i>~all pages~</i>		<p>\$membername: the member name.</p> <p>\$expired: true if the membership has expired.</p> <p>\$screenname: the screenname of the member, if he has changed it</p> <p>\$credit: Available credit for this member.</p>
add_field.inc.html	Frontkick internal use only.	N/A
add_field_ro.inc.html	Frontkick internal use only.	N/A
agreement.html	Displays the agreement for the membership, if enabled in Frontkick's Admin Area.	N/A
bio.html	Displays a performer's bio.	<p>\$profile: the performer's bio row from the perfprofile table.</p> <p>\$cred: credential to view the image.</p> <p>\$perf: the performer's database row.</p>
buy.html	Displays the 'buy-more-time' page.	
chat.html	Displays the public show for an online performer.	<p>\$performer: The performer's database row.</p> <p>\$params: The parameters needed to launch the viewer applet.</p>
email_verify.html	When email verification is checked for membership registration, this page will be displayed.	\$error: an array of strings representing potential errors to display.
email_verify.txt	Used with email_verify.html above: the text email message which will be sent to the user for confirmation.	<p>\$user: the user database row</p> <p>\$url: the validation URL</p>
fatal_error.html	Internal use only: when an unrecoverable error happens.	
footer.html	The footer included on all pages.	

<i>/site/templates</i>		
gallery.html	Displays a performer's image gallery.	\$colwidth: the width of the table's col. \$images: an array() or array (). Each row represent a database row of images. \$perf: the performer's database row.
header.html	The header included in all pages. Includes the navigation menu.	
login.html	The login page.	
lperf.html	Lists all performers.	\$colwidth: the width of the table's col. \$lperf: array() of performers' rows, with image credential.
member.html	Displays the membership renew page. Not currently used.	
notallowed.html	Displays when trying to enter a show and is not allowed to.	
notonline.html	Displays when the performer is not online	
pchat.html	Same as chat.php, but for private show	
profile.html	Displays the member profile (email, password, etc) and allow specific fields to be changed.	
profile_saved.html	When the profile was successfully saved	
redirect.html	Wrapper to redirect a page somewhere else.	
sendpass_failed.html	Cannot send a forgotten password	
sendpass_ok.html	Password was sent ok	
sendpass.txt	Text email sent when a member forgets his password	
signup.html	The signup page	
signup_mail.txt	Email sent after signup	
signup_multi.html	Not used	
sname.html	To change your screenname. Note: the screenname is part of the PHP session.	\$fksname: the screenname extracted from the PHP session.
sname_saved.html	When screenname has been changed successfully.	
thanks_error.html	Used by many pages to display 'thanks...' when there is an error	
thanks.html	Used by many pages to display 'thanks...'	

<i>/site/templates</i>		
uniq_login_exists.html	Not used	
uniq_login_free.html	Not used	
whois.html	Displays the whoisonline page.	<p>\$open, \$private and \$stage: are each an array() of current show, for each service classes.</p> <p>\$countperf: is the number of performers online.</p> <p>(other variables are more internals)</p>

As for the previous table, the following table describe the performer's templates:

<i>/performers/templates</i>		
<i>Name</i>	<i>Description</i>	<i>Available variables</i>
<i>~all pages~</i>		\$performername: the performer login name.
agreement.html	Not used. Reserved for future use.	
bio.html	Edit performer's bio.	\$profile: the performer's bio DBrow.
chose.html	Generic page to choose one image from the performer's image gallery.	<p>\$images: array() of array() of images with text and credential, where each row will be shown as a table row.</p> <p>\$scope: <i>gallery</i> or <i>preview</i> to indicate what size you want to display the image since this page is used both to choose the whoisonline image and as the profile page.</p>
email_verify.html	Not used. Reserved for future use.	
email_verify.txt	Not used. Reserved for future use.	
fatal_error.html	Same as for customer's site.	
footer.html	Same as for customer's site.	
gallery.html	Edit the performer's images	<p>\$images (enable images) and \$dimage (disable images): array() of array() of images with text and credential, where each row will display as a table row.</p>
header.html	Same as for customer's site.	
img_add.html	Form to download a new image.	

<i>/performers/templates</i>		
img_form.html	Form to edit the text associated with an image.	\$img_info: the text database row associated with an image.
index.html	Home page	
login.html	Displays the login page.	
payments.html	Report on the performer's payments.	Mainly \$payments: array of payments to be displayed.
profile.html	To edit the performer's profile (email, screenname, password, etc)	
profile_saved.html		
redirect.html	Same as customer site.	
report_select.html	The report main selection page.	
shows.html	Show setup: edit the keyword associated with a show and if allowed, the rate.	
signup.html	To sign up or register as a member.	
signup_mail.txt	Same as for customer's site.	
thank_register.html		
thanks_error.html	Same as for customer's site.	
thanks.html	Same as for customer's site.	

All of these templates can be easily edited in a text editor and modified to reflect your individual requirements.

## **Customizing PHP pages**

As said, Frontkick offers an further level of customization, since it includes the PHP source code for all its pages. Nearly There includes this information but does not provide technical support in this area.

To customize the behavior of the PHP pages, while reducing the work of upgrading Frontkick in the future, we strongly recommend you to proceed as follows:

Let's say you want to modify the whois.php, accessed from the /site directory. As shipped, and as explained in a previous paragraph, this php page do not exist in the /site directory, but is retrieved from the /ms/coresite directory.

The first step will be for you to copy /ms/coresite/whois.php to /site/whois.php.

That done, the '.htaccess' will no longer rewrite the request as the file now exists in the current directory.

You can now edit the /site/whois.php page, and make the modifications you want.

This method insure that the core directory /ms and any underlying directory is not modified, and that Frontkick will be easier to upgrade at future time. Of course, that said, we are making no guarantees that the functions used in these PHP scripts, such as whois.php, will not change in the future.

The following table provide more details about some of the common pages used on the customer site. These pages are all displayed in a browser using the HTTP GET method, which is standard.

<i>Page</i>	<i>Parameters</i>	<i>description</i>
chat.php	perfsname=YYY	Starts the applet for the public show of performer YYY, where YYY is the performer's screenname.  If the performer is not online, or if the show has no public view, an error page will be displayed using the template.
pchat.php	perfsname=YYY	Starts the applet for the private show of performer YYY, where YYY is the performer's screenname.  If the performer is not online an error page will be displayed using the template.  Sidekick will generated a token corresponding to the maximum time available for the session, according to the customer's available credit and the show rate.
bio.php	perfsname=YYY	Displays the bio of performer YYY, where YYY is the performer's screenname.
gallery.php	perfsname=YYY	Display the image gallery of performer YYY, where YYY is the performer's screenname.  Only images that have been 'enabled' by the performer will be displayed. Also, the order chosen by the performer will be honored.
whois.php		Displays all the performers that are currently online.  Also refreshes the cached page if it doesn't exist.
	service=XXX	Will restrict the list to only those that shows that belong to XXX, where XXX is a class of service, such as public/private/stage
	keywordssel=YYY	Will restrict the list to only those shows that include the keyword YYY.

## **4.4 Frontkick, Sidekick, Activeshows, private session & tokens management**

*This part is the core of the integration between Sidekick and Frontkick. While much details are given here, it is **strongly** recommended not to modify this part.*

### **Overview**

Sidekick and Frontkick communicate using two channels.

The first one is event driven and is triggered by any change of state by Sidekick. Sidekick will generate a call (from itself) to Frontkick (/ms/frontkick/callout.php) using the callout. Based on the information received, Frontkick will update its state.

The second communication channel is through Sidekick's RemoteAccess() HTTP request and is used by Frontkick when it needs to query Sidekick for more information, or to apply changes.

You might be interested to understand that Frontkick always acts as master while Sidekick is always considered the slave. If some information differs between Frontkick and Sidekick, Frontkick's notion will always take precedence – that is, Frontkick has priority and will try to re-sync Sidekick according to its own information.

### **Shows**

Sidekick events are used to keep the activeshows table up to date. The activeshows table represents all the shows that are currently online. All the information associated with each show is stored in this table, including the performer's name, the rate and description of the show, and other information which is used internally.

In particular, the whois page will use this table to generate a page with a list of the currently available shows.

### **Sessions**

Sidekick will generate events for all sessions – they are generated when a customer, member or not, joins a showroom, in private or public.

As previously said, Frontkick always acts as the master in relation to Sidekick. For any session, public or private, Frontkick will expect Sidekick to restrict access to it using Sidekick's "token restriction." Frontkick will call upon Sidekick to handle the token creation, as well as destroying tokens that are no longer in active use.

For public view of a show, Frontkick (through chat.php) will, after checking if the show is open to public, allocate a temporary token with an allowance of zero seconds. This token doesn't represent any value but is just here to grant the client access to the chatroom, and to centralize the responsibility of determining who can access a show to Frontkick. Those temporary tokens have a relatively short life, and will be cleaned up automatically by Sidekick.

For private sessions, Frontkick (through pchat.php) will allocate a token with an allowed time based on the available credit in the member's account and the show's rate. This token will be recorded in the tokens database, associated with the member ID and other information.

Once the token is allocated, Frontkick can launch the customer applet so the client can enter the private view of the show.

If the applet launch is successful, Sidekick will produce an event to indicate a new private session has start, and Frontkick upon receiving it, will update the specific token database record to show that the session has successfully started.

At the end of a private session, Sidekick will produce another event to indicate the session has ended, and its duration. Based on the duration and on the stored token information, Frontkick can update the member information, thus reducing his available credit based on the duration and the show's rate.

## **4.5 Payments & Credits for private sessions**

*This part is the core of the integration between Sidekick and Frontkick. While much details are given here, it is **strongly** recommended not to modify this part.*

Frontkick maintains for each member the amount of credit they have available to buy time for private sessions. There are really several parts in this credit management: how to link with an external payment system; how to define the relationship between the payments system products and the corresponding number of credits; and the implementation of this auto-update on the Frontkick side.

The link with the external payment system is describe in the next chapter, in the Frontkick Admin Area examples. This section is meant to explain Frontkick internals on how an external payment will result in an update of the associated member record and adds additional credits to it.

Frontkick relies on aMember's implementation of payments management. For aMember, a payment will have differents states, as explained here:.

First, based on the buy page of Frontkick, a new payments record is created when a customer selects a time product and clicks on 'Buy'. This payment has not yet completed, and is marked as such in the Frontkick database. Frontkick then forwards the customer to the external payment system, allowing him to complete the payment.

Once the payment is completed and validated by the payment system, the payment system will use IPN to call back Frontkick and record the fact that the payment transaction was sucessfully handled, and therefore that Frontkick should validate the associated payment record.

Those IPN pages are provided by aMember, as part of aMember support for the various payments plugins. Each aMember IPN will change the state of the associated payment to be completed, and will call any plugin function which have been register to be triggered on this change of state.

Frontkick adds one function on aMember list of plugins associated with accepted payment: it is the `update_time_on_payment()` function defined in `/ms/frontkick/common.inc.php`.

For each newly accepted payment, this function will first check to see if the payment corresponds to a product defined as a Frontkick 'credit' related product. If so, it will add the corresponding credit to the member account.

Also, as Frontkick allows member to buy additional credit while in private session, this function will check to see if the member is currently in a private show, and if so, will, using the Sidekick RemoteAccess request, add the correct amount of time to the session.

## **4.6 Images management and Frontkick**

To create galleries for the performers and thumbnails for the whois page, Frontkick offers a way to upload, store, manage and display images.

## Download & store

Images can be uploaded in one of two different ways:

1. Using the performer's site (/performers), the image gallery offers a upload page
2. Using the Sidekick performer program, 'SnapShot upload' feature.

In the case the performer is uploading from the performer's gallery page, the size of the file is limited to the MAX size defined in the control panel -> Configuration -> Images tab.

Also, the same control panel tab will enable you to limit the Max number of images that each performers can store.

When the upload has completed through the Sidekick's SnapShot feature, Frontkick will ensure that the call originated from a valid Sidekick program and performer by validating the request against its performer's database.

Once an image has been uploaded, Frontkick will store it in a specific table in its database. It will pre-process the image to store a thumbnail and a medium size view of it, along with the original image. The size of the thumbnail and medium view are defined on the same configuration tab in the Frontkick Admin Area.

Newly uploaded images are stored as disabled by default. The performer can use the gallery page to enable desired images, or even to delete or re-order them. As you can imagine, only enable images are shown to the customer. Disabled images continue to be stored by Frontkick, perhaps for future display.

## Viewing images

Frontkick implements a sophisticated yet straightforward restriction around displaying images to avoid people bookmarking, linking to or viewing stored images directly.

To display an existing stored image, you must use the /ms/performers/views.php page and pass it three parameters:

- The id of the image to display
- The size -- preview (thumbnail size), gallery (medium size) or view (original size).
- A credential

The credential is computed using a one way hash scheme called "Shal" from the display order field stored with the image. To display image \$my\_id, the following PHP code can be used:

```
$pri = db_get_info_img($my_id);  
$cred = shal($pri['displayorder']);  
$t->assign('my_id', $my_id);  
$t->assign('cred', $cred);
```

The associated Smarty template can now display the image using the following code:

```
<img src="/ms/performers/view.php?action=gallery&id={$my_id}&cred={$cred}>
```

In addition, there is a background, daily task that will change the value of the 'displayorder' field for all stored images, and therefore reset all the credential so that any client attempt to bookmark will fail within 24 hours.

## 4.7 Caching of the *who is online* page

For performance reason, Frontkick is configured to keep a cached copy of the who is online page and update the cache only when a performer goes offline (a show is terminated) or when a new one come online

(a show is started).

The caching method is done as follows: When a start or end show event is received from Sidekick, Frontkick (in callout.php) will remove the cached page. Then the next time a client requests the whois.php page, it will be to regenerated and then displayed.

*Note: The whois.php page currently doesn't cache pages when filters (on keywords or services) are applied.*

# 5 Frontkick Admin Area Examples

---

The Frontkick Admin area is designed and laid out to be self-explanatory. You should have little trouble working through the menus and understanding what is happening. With this in mind, this chapter will highlight on real life examples.

## **5.1 Hooking up a payment system**

Often, the first thing to do after installing Frontkick, is to get it hook up to whatever payment system you are planning to use, so that your 'buy' page will work. This is done through the Frontkick Admin Area.

Frontkick will provide the connection to your payment system but it is up to you to establish an account. You will need to contact these companies and apply for an account.

### **Enabling new payment plugin**

1. Log on as admin on /ms/adm.
2. Choose Configuration on the left menu.
3. In the configuration page, click on the Plugins link in the top menu.

Select from the list the payment plugins, which you wish to enable. Make sure to keep the Free payment – it doesn't mean you'll be giving away any time, but the Free payment is needed for proper functioning

Click Save to save your choice.

Clicking Save will display new link in the top menu, one per payment plugin selected. Next, go through each of these new payment links and configure each payment system with the right information.

### **Configuring the products**

The next step, now that you have setup your payment system(s) is to link your products with the payment system. So, depending on how each payment system works, you will establish the link between your products (or purchases options) and their corresponding products in the payments system. For that, you will have to edit each of your purchase options and fill in the additional fields added by the payment system. Read the note at the bottom of each payment system plugin page in the configuration menu -- it should provide you wall the details of what to do.

Also, while editing the your purchase option, you can assign a specific payment system to each of them.

Lastly, and as a side note, if you have multiple payment systems enabled and haven't assign a specific one to each of your purchase options, the buy page will display to the customer a list of available payment systems to use for this transaction.

## **5.2 Changing your purchase options**

As shipped, Frontkick offer four purchase options, namely bronze, silver, gold and platinum. Using the menu of the Admin Area, you can easily change their names and descriptions. You can also change their prices, or reduce the number of available options by enabling or disabling them.

You should note that the price defined in Frontkick may be overwritten by the payment system, depending on the features offered by it. On a payment systems where you have to duplicate each Frontkick products,

changing the Frontkick side of the price is informative only: you must also change the respective payment system's product price to stay in sync.

## **5.3 Mapping product to Frontkick credit**

The previous two paragraphs showed you how to enable customer to buy predefined purchase option, and how to change their price.

Now, to map each of these purchase options to the amount of credit Frontkick has to add to the member account, you should use the 'Purchase Options credits' on the *Frontkick* Admin Area top menu.

On this page, just enter the amount of Frontkick credit each product corresponds to. By default, there is a one to one mapping between the default product price in \$ and the amount of credit added.

## **5.4 Frontkick/Sidekick services**

Frontkick has the notion of three classes of services, to which Sidekick services are mapped to:

- *open*: this service is open to the public. Frontkick will allow non-members to join.
- *private*: this service is open to the public. Frontkick will restrict it so that only members can join.
- *stage*: this service is not available to the public. Frontkick will restrict it so that only members can join.

Sidekick defines services which specify the video/audio/chat characteristics of a session, and if it allow a public view of it or only a private view. Frontkick will request Sidekick to restrict all the view of any service to use the token restriction, so it can be sure access to those view will be controlled by the chat and pchat pages.

Frontkick control panel, under the SK service link in top of the Configuration menu, offers you a way to configure each services defined in Sidekick. The configuration choises includes the default (site wide) rate of this service when in private session, the option of allowing or not non-member to enter the (free) public view of it, when such view exist and the option of delegating the control of the rate to each individual performer.

## **5.5 Performer Accounts**

### **Adding performers**

There are two main ways that a performer account can be established. First, as an administrator you can click on the **Add Performer** link and enter her information directly. Second, a prospective performer can go to the performer's area and click on **Signup**, where she can register.

When a performer self-registers, her account is not approved. As an administrator, you have to go in, edit her account and approve it. If you add the performer yourself, you can select "Yes" in the Approve field to have her account immediately approved.

### **Sync performers**

Frontkick and Sidekick should both have the same list of performers. This normally happens automatically -- for example, when you add a performer account in Frontkick, this information is automatically

propagated to Sidekick. You can verify that Frontkick and Sidekick are "in synch" by going to the admin area and clicking on **Sync Performers**. This screen will display a list of performers that are known to both and also those that are in one but not the other. There are choices that allow you to easily correct any discrepancies.

You might notice that Sidekick has a way to add performers. We suggest that you don't add performers through Sidekick since Frontkick won't be aware of this -- unless of course, you use **Sync Performers**.

## **Status & expiration date**

The performer status indicates if the performer is enable or not. Any change of its status will automatically generate synchronization with Sidekick, to update Sidekick performer account. When a performer status is changed to disable (or not approved), Frontkick will remove the associated Sidekick performer account. When the status is changed from disable to enable (or approved), Frontkick will create the associated Sidekick performer account.

Aside this status field, performer's account in Frontkick also have an expiration date, which will trigger change of status and therefor synchronization with Sidekick. This will happen two different ways. The first one will be during edition of the performer account by an admin. If the expiration date is changed to a past date, Frontkick will automatically set the performer status to disable and remove the Sidekick corresponding account. Also, a daily task will run through all performers account and automatically disable the one who reached their expiration date.

## **Extending fields of a performer's account**

The pre-defined performer account record was defined to hold the most common information related to a performer, however it is very easy to add additional field to it so that you can customize it to your needs.

To do so, use the Additional fields for performers menu in Frontkick control control panel. You will also have the option to make those fields visible or not and editable or not to the performer, while the admin will always have full access to those fields through the performer menu of the control panel.

# 6 Upgrading Frontkick

---

## 6.1 Overview

Frontkick can be easily upgraded from one version to another. Generally, three steps are needed:

- Back up your Frontkick directories (/ms, /performers and /site) for example, with FTP, and your database (using Frontkick/Backup). This step is IMPORTANT!

Copy new files into the /ms, /performers and /site directories.

- Run [http://yourDomain.com/ms/adm/upgrade\\_db.php](http://yourDomain.com/ms/adm/upgrade_db.php) from your browser to update the database with any new fields or tables. You may want to change that

## 6.2 Keeping your modifications

If you have modified any files, you will have to take additional steps to ensure that you don't lose your customizations and these files continue to function correctly.

Unless you are a serious hacker, we recommend that your modifications be restricted to the /performers and /site directories. By doing so, you can probably simply copy the /ms directory and things will continue to function.

# 7 Operations

---

## 7.1 Overview

Frontkick was designed as a zero-maintenance product. This means that on a day-to-day basis, absolutely no administrative effort should be needed to keep the system running.

For various reasons, the following sections are meant to provide a better understanding of some of the operational aspects of Frontkick. For example, these sections help answer questions such as “Is Frontkick running?”

## 7.2 Log & errors files

Frontkick writes significant events to the database and provides screens to display them. These log messages can be very useful for ensuring an on-going high-quality operation. For example, scanning these messages periodically for errors and warnings, might allow you to proactively eliminate a problem.

Frontkick use two types of messages: debug logs and errors messages, stored both in different tables.

Debug messages enable you to have an inside view of Frontkick. All the main function will output trace information on it. Debug log level can be changed in the control to be less or more verbose.

Errors logs are serious conditions reported by Frontkick. They should be look at carrefully, and will most of the time require imediate action from the admin. The only exception is the IPN trace of the payment system, when enable, which will be logged as an error msg.

The debug log can be automatically purged: use the Configuration menu of the control panel to enable this option, and set up the maximum number of days you want to keep track off.

## 7.3 Cleaning up old records

Frontkick records in log files information related to its operation. It will record key events in log files. Over time some of this information which may not be relevant any more. These logs might be useful for tracking down bugs, or for understanding the history behind a client issue. As a specific example, consider this: The information might be an old yet not completed payment -- which will occur when a client creates the payment and so is forwarded to the external payment system, but the transaction is never completed).

Also note that Frontkick doesn't write any data that is considered critical to your on-going operation in log files. It maintains all critical data in other database tables so that other activities, such as reporting, will continue unaffected by deleting any or all log files.

To clean up the database and recover the used space, you can use the Delete old records menu in Frontkick's Admin Area.

## 7.4 Recovering after a server crash

Under exceptional conditions, Frontkick might get into a state which is internally inconsistent. For example, Frontkick might believe that a show is still running when it isn't. Frontkick could become inconsistent if a service such as Apache stops running. Since the services that Frontkick relies on are very reliable, this potential inconsistency is considered exceptional, but this section explains how to handle it.

The cleanup menu in the Frontkick Admin Area will handle the recovery, by removing all records from the

activeshows and tokens tables. If it encounters a problem during this operation, it will log any problem as an error which cannot be automatically cleaned up, and which will require manual intervention.

As an example, take a server crash while a client is in a private session. The session end event normally sent by Sidekick callout will be lost, so the administrator will have to manually go through the logs and re-set this client's credit to the correct amount.

## **7.5 Security Issues**

There are several areas to review to ensure that your installation is secure. Here is a list of important areas:

- For sensitive data, the `https` should be used. This can be enabled during setup.
- As explained in the section called "Certain Directories must be Writable", the `/ms` directory should not be writable. Note: The `ms` directory has to be writable at the install time
- The database should be configured with a login and password that is difficult to guess. This login and password can be entered during setup.

Of course, securing a web site operation involves many steps and this list is only meant as a short list which is specific to Frontkick issues.

# 8 Frequently Asked Questions (FAQ)

---

## **I. Some customers pay with a wire transfer. How is this entered?**

When you receive a wire transfer from a customer, you can enter it manually into Frontkick. Use Browse Users to display the User Payments/Subscriptions page. At the bottom, specify the item purchased (e.g., Bronze), that it is a "Manual Payment" and the amount. Then click **Add**.

## **II. I launch a show and it gets immediately terminated!**

Why you launch a show, Frontkick will only allow it to continue if the following conditions are satisfied:

- The performer must be defined in Frontkick. This condition will fail if the performer is only defined in Sidekick. Under admin, use Setup/Sync Performers to resolve this.
- The performer is not approved in Frontkick.
- Frontkick doesn't have a service defined for this show. If the show is launched, Frontkick will ask Sidekick for the service with which it was launched.

## **III. Can I use multiple payment systems?**

Yes. Just select multiples payments plugins and configure them according to their instructions. You can also assign specific payment systems to individual purchase option.

## **IV. Can Sidekick and Frontkick be on different servers?**

Yes. All their communication is through an HTTP based API it is possible to host Frontkick and Sidekick on two different servers even located physically apart. This topology can be implemented for various reasons, one of which might be to increase the scalability of the system.

## **V. Does Frontkick support paid memberships?**

Yes, edit the Membership option in Frontkick's Admin Area. You will be able to assign a price and a time duration, such as one year. By default, the price is set for \$0.00, so membership is free.

## **VI. I use multiple payment systems. How can I track all of them using Frontkick's tracking system?**

You can't. Frontkick currently only supports a single payment system in its tracking system.

## **VII. How do I give free credit to a member?**

The easiest way is to edit the member account, where the credit field can be changed by the administrator.

## **VIII. I want to email a single performer or member. Why is there no form for that?**

You can just click on the performer or member email when you display his or her profile.

## **IX. I want to temporally disabled a performer, how do I do that?**

Edit the performer's account and change Approved to No. Two things will happen: First, the performer will be deleted from Sidekick so she can't launch a show. And second, she won't be able to log on Frontkick.

## **X. I don't like PHP. Do you have a Perl version of Frontkick?**

No. :-p And don't bother asking for one!