



# Configuration Management

For FastX 3.2

<b>Introduction</b>	<b>3</b>
Audience	4
File Locations	4
Local Configuration	4
Standalone Configuration	4
Cluster Configuration	4
<b>Local Configuration</b>	<b>4</b>
www.json	5
Location	5
Content	5
Notes	5
broker.json	5
Location	5
Content	6
gateway.json	6
Location	6
Content	6
settings.json	7
Location	7
Content	7
session.json	7
static.json	7
debug.json	7
Content	7
FastX.conf	8

Content	8
<b>Standalone Configuration</b>	<b>9</b>
Changing Modes	9
Notes on *.db files	9
Example	10
Correct	10
Incorrect	10
Also Incorrect	10
Cluster Mode	10
Converting Standalone Files to a Cluster	11
Uploading Configuration from a Standalone to a Cluster	11
Downloading Configuration from a Cluster	11
General Configuration	11
Standalone File	11
Cluster Collection	12
Content	12
Ldap Configuration. Used to get extended profile data of users	12
OIDC configuration	12
SSH Configuration	12
Proxy Authentication	13
Bookmark Filter	13
Client	13
Update-web	14
Logins	14
Motd	14
Job Scheduler Filter	14
Session Start	14
Update	16
Use	16
Website	17
Permissions	17
Standalone File	17
Cluster Collection	17
Content	17
Admin Users	17
Admin Groups	18
Manager Users	18
Manage Groups	18
Full-User Users	18
Full-User Groups	18

Disabled Users	18
Disabled Groups	19
Enable Managers	19
Enable Limited Users	19
Manager Permissions	19
Bookmarks	21
Standalone File	21
Cluster Collection	21
Content	21
Events	21
Standalone File	21
Cluster Collection	21
Content	21
Forms	22
Standalone File	22
Cluster Collection	22
Content	22
Http Headers	22
Standalone File	22
Cluster Collection	22
Content	23
Load Balancing Functions	23
Standalone File	23
Cluster Collection	23
Content	23
Metrics	23
Standalone File	23
Cluster Collection	24
Content	24
Notifications	24
Standalone File	24
Cluster Collection	24
Content	24
Profiles	25
Standalone File	25
Cluster Collection	25
Content	25
Themes	26
Standalone File	26
Cluster	26
Content	26

# Introduction

FastX is an extremely configurable product meant to serve the needs of individual users and enterprise clusters alike. FastX ships with a [visual configuration editor in the web interface](#) to configure every parameter of the system. Administrators however may wish to use their own configuration management tools (such as Puppet) in order to automate the configuration and distribute it across multiple instances. This guide provides a deep dive into the automated approach of configuration management.

This document describes the file formats and locations of the configuration files in FastX.

## Audience

This document is intended for system administrators who wish to automatically configure FastX without using the GUI (for example configuration management via Puppet). System administrators who wish to manually configure FastX should use the [GUI configuration](#).

## File Locations

FastX stores its configuration in multiple locations depending on its intended use.

### DotEnv and Environment Variables

The environment variable `FX_CONFIG_DIR` specifies the location of the local configuration files. Its default value is `/etc/fastx`. The environment variable `FX_VAR_DIR` specifies the location of directories maintained by FastX for storage, licenses, certificates, and key files. Its default value is `/var/fastx`. The environment variable `FX_LOCAL_DIR` specifies the location of files and directories that are local to the current system. Its default value is `$FX_VAR_DIR/local`.

These environment variables can be changed from their default values by using the file `/usr/lib/fastx/3/.env`. This file is in [DotEnv format](#), which looks like sh format.

When run as a service, the FastX server also gets environment variables from `/etc/sysconfig/fastx3`. This environment file overrides the values in the `.env` file.

This document specifies the default location of files. When `/etc/fastx` is specified, keep in mind that this directory is actually `$FX_CONFIG_DIR`, if set.

### Local Configuration

Configuration that is ALWAYS local to the current system, whether in standalone mode or cluster mode is located by default in `/etc/fastx`.

These files include configurations for static data, server data, system specific options and debugging among others.

## Standalone Configuration

In a standalone system, configuration that would usually be shared across a cluster is located by default in `/var/fastx/local/store`. This configuration is local to this system only. It is in a standard `*.db` format that can be easily uploaded to a cluster configuration via the upload tool.

In standalone mode, FastX works as an isolated system. All configuration is local. FastX does not attempt to connect to any other installation.

## Cluster Configuration

Configuration that is shared across all cluster members is located in a shared database. [\(See Clustering Guide for more details\)](#) Administrators configuring the cluster will typically store the configuration in the `*.db` format of the standalone configuration, and then use the `upload-store` tool to upload the configuration into the shared database. This only needs to be done **once**. Administrators can also get the information from the shared database using the `download-store` tool.

For more information see [converting standalone files to cluster](#)

# Local Configuration

Configuration that is private to the local system is stored in the config directory (default `/etc/fastx`, set by `FX_CONFIG_DIR`)

These files are standard JSON files. Comments are added for description ONLY. In an actual file, there should be no comments. Make sure the file is [valid JSON](#)

## www.json

This file holds the information needed to start the web server including port, certificate paths, and other options.

This file must be world readable as sessions use this file to attempt to reconnect with the web server

## Location

`/etc/fastx/www.json`

## Content

```
{
  "ca_file": "/path/to/ca/file", /* Path to the certificate's CA file */
  "cert_file": "/path/to/certificate/file", /* Path to the certificate file */
  "key_file": "/path/to/key/file", /* Path to the certificate's private key file */
}
```

```
"pfx_file": "/path/to/pfx/file", /* Path to the PFX file (if ca, cert and key not
used) */
"port": 3300, /* WWW port */
"https": {}, /* HTTPS Options Object */
"required": false, /* require the web server to start sessions */
}
```

## Notes

If the desktop client is used to connect via SSH, and the FastX web server (“fastx3” service) is not running, then the connection will fail if “required” is true. Otherwise, the client will show a warning message about the server not running. The www.json file must be publicly readable for this to work.

## broker.json

This file holds the information needed to connect to a broker which is used in clustering. This file is typically created by the cluster manager and copied from the cluster manager to the FastX server.

FastX uses a microservices architecture called Molecular. Cluster communication is done through a broker, [with different transporters available](#). Molecular is highly configurable. See the [documentation for broker options](#)

## Location

/etc/fastx/broker.json

## Content

Default Transporter Type (standalone usually omitted)

```
{
  "namespace": "fastx-cluster", /* cluster namespace */
  "broker": {}, /* any other broker options */
}
```

Transporter Type Redis (cluster)

```
{
  "transporterType": "redis",
  "host": "server.example.com", /* redis host */
  "port": 6379, /*redis port */
  "password": "ABCDEFGH" /* redis password */
  "options": {}, /* transporter options */
  "tls": {}, /* TLS options */
}
```

```
"rejectUnauthorized": false, /* make sure certificate is valid */
"clientCertFile": "/path/to/cert", /* client certificate file*/
"clientKeyFile": "/path/to/key", /* client key file */
"clientCaFile": "/path/to/caFile", /* client ca file */
"namespace": "fastx-cluster", /* cluster namespace */
"broker": {}, /* any other broker options */
}
```

## gateway.json

This file holds the information for connecting to an external gateway when the server is on a private network. This file is typically created by the gateway installer and copied from the gateway to the FastX server

### Location

/etc/fastx/gateway.json

### Content

```
{
  "url": "https://server.example.com:33300", /* url of the relay */
  "key": "SECRET_KEY", /* secret shared key */
}
```

## settings.json

Local settings of the web server

### Location

/etc/fastx/settings.json

### Content

```
{
  "disable": ["login", "start"], /* Array of settings to disable */
  "sshport": 22, /* SSH port to use when connecting via ssh */
  "authorizedKeysDir": "STRING_VALUE", /* directory where to store
authorized keys in a CLUSTER */
}
```

```
"hostname": "STRING_VALUE", /* Override hostname of this system */
}
```

## session.json

When the web server IS NOT running, this is the [profile of the session](#). If this file contains invalid JSON, an attempt to start a session will fail with an error.

When the web server IS running, this file is ignored.

## static.json

Custom user defined JSON object that gets sent with each server update

```
{
  /* add custom JSON parameters */
}
```

## debug.json

Turn on web server debugging options

### Content

```
{
  "fastx:request": false, /* log HTTP requests */
  "fastx:log": false, /* misc logging */
  "fastx:error": false, /* misc error logging */
  "fastx:api:request": false, /* log api requests */
  "fastx:api:process": false, /* log processing of the request */
  "fastx:link:messages": false, /* log web server to link messages */
  "fastx:link:auth": false, /* log web server to link connections */
  "fastx:link:netstring": false, /* log low level webserver to link
messages */
  "fastx:cluster:comm": false, /* log cluster communication */
  "fastx:cluster:sse": false, /* log cluster server sent events */
  "fastx:cluster:db": false, /* log cluster database communications */
  "fastx:oidc": false, /* log OpenId Connect */
  "fastx:ssh": false, /* log SSH connection */
  "fastx:ws:connection": false, /* log session protocol connections */
  "fastx:ws:messages": false, /* log all protocol messages */
}
```



```
}
```

## FastX.conf

This file sets the following variables, which are used to define session directories and paths for other files:

Variable	Default Value	Description
FastXPath	@HOME@/.fastx_server	History and user bookmarks are saved here in failsafe mode.
HOST	From hostname() library call	Used in various places.
SessionPath	@HOME@/.fastx_server/@HOST@/sessions	Where session directories are created.
TMPDIR	/tmp	Where link sockets are created.

## Content

```
# This file sets variables one line at a time.
# The # sign starts a comment, unless it is enclosed in @ signs: @#@
# Variables are expanded between @ signs, including environment variables.
# If variables do not appear in this file, they get their default values.
FastXPath=@HOME@/.fastx_server
# The FastXPath and the SessionPath variables can be set to somewhere other
# than @HOME@ if the user's home directory is not mounted read/write
# before the session starts.
SessionPath=@HOME@/.fastx_server/@HOST@/sessions
```

## Standalone Configuration

FastX 3.2 and later create distinct separations from the modes of operation. Take into account how you want your installation to function as these are mutually exclusive modes.

## Changing Modes

The way a FastX server operates is defined at startup and stays that way for the duration of the process. The mode itself is defined by the existence of a file called [broker.json](#) located in `/etc/fastx/broker.json`

More specifically, within the file, the parameter `"transporterType"` defines the transport mechanism that is used for clustering. If `transporterType` is a falsy value, FastX will operate in Standalone mode.

## Notes on \*.db files

The standalone system files are based off of JSONL. They are all located in the `"store"` directory and the name of each file ends in `".db"`. A [JSONL](#) file is a file that contains one line per JSON object. The object is terminated by a newline and may be surrounded by whitespace.

In the following document, the JSON objects are pretty printed for readability, and includes comments, which is not valid JSON. In an actual standalone file, this would cause an error as the JSON object is not contained on the same line. **Make sure each JSON object in a \*.db file is on one line.**

Each object in the file will contain an `"_id"` field. If `"_id"` is missing, that object will be ignored. If the object has the same `"_id"` as a previous object in the file, the later object will override the previous one. The FastX server will periodically "compress" the file, meaning earlier objects with duplicate `"_id"` fields will be removed, as well as objects without an `"_id"` field. Modifications via the GUI will cause the new object to be appended to this file, which will then be periodically compressed.

## Example

Correct

```
{ "_id": "id1", "hello": "there", "json": "on_one_line" }
{ "_id": "id2", "hello": "world", "json": "correct" }
...
```

Incorrect

In this example, lines are pretty printed

```
{
  "_id": "id1",
  "hello": "there",
  "json": "on_one_line" }
```

```
{ "_id": "id2",  
  "hello": "world",  
  "json": "correct"  
}  
...
```

## Also Incorrect

```
{ "_id": "id1", "hello": "there", "json": "on_one_line" },  
{ "_id": "id2", "hello": "world", "json": "correct" }
```

## Cluster Mode

In clustered mode, the configuration is stored in the shared database on the cluster manager. Once the cluster has been configured, new FastX installations will automatically pick up any shared configuration.

To set up a cluster manager using a configuration manager, first install and start the cluster manager using the cluster setup script. Then create the desired configuration files in JSONL format as described above in “Standalone Mode”. For this example, we will put the files in a directory stored in `$FASTX_CONFIG_STORE`. Then the following command will be run with the following variables:

- `FASTX_CONFIG_STORE` -- As described above.
- `FASTX` -- The directory that includes the fastx “microservices” directory.
- `NODE` -- The path to the “node” executable. Node version at least 10 is required.
- `BROKER_JSON` -- The “broker.json” file created by the cluster manager setup script.

```
$NODE $FASTX/microservices/tools/upload-store.js config --bjf $BROKER_JSON --dir  
$FASTX_CONFIG_STORE
```

The above command does not have to be run on the cluster manager itself, but it must be able to connect to the cluster manager. If the system has FastX installed, then `$FASTX` will usually be `/usr/lib/fastx/3`, `$BROKER_JSON` will usually be `/etc/fastx/broker.json`, `$NODE` will usually be `/usr/lib/fastx/3/nodejs/bin/node`, and `$FASTX_CONFIG_STORE` will often be `/var/fastx/local/store`.

## Converting Standalone Files to a Cluster

## Uploading Configuration from a Standalone to a Cluster

Once your cluster manager is set up, you may want to migrate your configuration into the cluster. FastX ships an upload tool to take a standalone configuration format and upload it into a running cluster.

1. Make sure the cluster manager is running.
2. Run the command: **`/usr/lib/fastx/3/nodejs/bin/node /usr/lib/fastx/3/microservices/tools/upload-store.js config --bjf /etc/fastx/broker.json --dir /var/fastx/local/store`**

## Downloading Configuration from a Cluster

The configuration will be downloaded into separate JSONL files that are readable by the standalone installation.

1. Make sure the cluster manager is running.
2. Run the command: **`/usr/lib/fastx/3/nodejs/bin/node /usr/lib/fastx/3/microservices/tools/download-store.js config --bjf /etc/fastx/broker.json --dir /var/fastx/local/store`**

## General Configuration

General Configuration for FastX. This file is used for configurations that are stored as a single object defined by the `_id`. Configurations that can contain multiple objects (like bookmarks, profiles etc) are located in their respective stores.

### Standalone File

`/var/fastx/local/store/config-store.db`

### Cluster Collection

`config-store`

### Content

Ldap Configuration. Used to get extended profile data of users

```
{
  "_id": "ldap",
  "url": Url, /* Url of the Ldap Server*/
  "bindDN": "STRING", /* BindDN */
```

```
"bindPassword": "STRING", /* Bind password*/
"searchBase": "STRING", /* Search Base of searches */
}
```

## OIDC configuration

```
{
  "_id": "oidc",
  "enabled": Boolean, /* Enable/disable OIDC authentication */
  "redirectUrl": URL, /* Url to redirect after success:
https://your.example.com/auth/oidc/callback */
  "issuer": "STRING", /* OIDC Issuer */
  "clientId": "STRING", /* Client Id*/
  "clientSecret": "STRING", /* Client Secret */
  "logoutRedirectUrl": URL, /* Redirect URL after Logout,
https://your.example.com/auth/oidc */
}
```

## SSH Configuration

```
{
  "_id": "ssh",
  "disable": Boolean, /* Enable/disable SSH authentication */
  "disablePublicKeyAuthentication": Boolean, /* Enable/disable SSH Public
Key authentication */
  "hideBackground": Boolean, /* Hide background image on web page */
  "hidePassword": Boolean, /* Hides Initial password field on web page
so only username is visible */
  "passwordAuthenticationOnly": Boolean, /* Disables all SSH
authentication methods except username/password */
  "theme": "light" | "dark" | "classic", /* Sets the web page theme */
  "serverFilter": {
    "script": "script to filter out servers */
  }
}
```

## Proxy Authentication

```
{
  "_id": "auth-proxy",
  "secret": "SECRET_KEY", /* Secret key used to sign the proxy JWT */
}
```

## Bookmark Filter

```
{
  "_id": "bookmark-filter",
  "script": "STRING", /* Bookmark filter script */
}
```

## Client

```
{
  "_id": "client",
  "disableMenu": Boolean, /* Hide the Menu */
  "disableModal": Boolean, /* Hide any Modal popups */
  "disableTokenValidation": Boolean, /* Do not verify the token on
connection */
  "keyboardLayout": 1033, /* Default Keyboard layout if none currently
exists on client */
  "keyboardType": 4, /* Default Keyboard type if none currently exists
on client */
}
```

## Update-web

```
{
  "_id": "update-web",
  "script": "javascript function"
}
```

## Logins

```
{
  "_id": "logins",
  "disbleKeepalives": false,
  "linkDaemonKeepalive": true,
  "maxApiTokenAge": 10080,
  "shutdownClusterDaemonsOnLogout": false,
  "shutdownLinkDaemonOnLogout": false,
  "enableFileDownload": true,
  "fileShareRootDir": "$HOME"
}
```

## Motd

```
{
  "_id": "motd",
  "motd": ""
}
```

## Job Scheduler Filter

```
{
  "_id": "job-schedule-filter",
  "fun": "STRING_VALUE", /* The job scheduler function */
}
```

## Session Start

```
{
  "_id": "session-start",
  {
    "forms": {
      "script": `async function(input) {

        // input.data -- input sent from the api call
        // input.user -- user login object
        // input.isAdmin -- is the user an admin
        // input.isManager -- is the user a manager
```

```

// input.userGroups -- Linux User Groups of the user
// input.forms -- array of form object

//return a falsy value to continue
//return a form object to pop up a form
//return the string 'SERVERLIST' to allow the user to choose from the
available servers (dynamically generated form from the serverList function)

//if there are no forms go to the next step
if(!input.data.forms)
{
    return;
}
//if the input.data.forms is a string return the form it found or
null
if(typeof input.data.forms === 'string') {
    if(input.data.forms === 'SERVERLIST')
        return 'SERVERLIST';

    return input.forms.find(c => c.id === input.data.forms);
}
//if input.data.forms is not an array return
if(!Array.isArray(input.data.forms)) {
    return;
}
//if input.data.forms is empty return
if(!input.data.forms.length){
    return;
}

//cycle through the forms until you find a valid one to return
//or return go to the next stage
do{
    let f = input.data.forms.shift();
    let myForm;
    if(f === 'SERVERLIST')
        return 'SERVERLIST';

    else if(myForm = input.forms.find(c => c.id === f)) {
        return myForm ;
    }
} while(input.data.forms.length);

```



```

    return false;

}
},
"selectServer": {
  "script": `async function(input) {
    // input.data -- input sent from the api call
    // input.user -- user login object
    // input.servers -- array of server objects
    // input.isAdmin -- is the user an admin
    // input.isManager -- is the user a manager
    // input.userGroups -- Linux User Groups of the user

    //return a filtered list of servers for the user specified by
    //username, or none to disable server selection
    return;
  }` }
}

```

## Update

```

{
  "_id": "update",
  "url": 'https://www.starnet.com/auto/upgrade32',
  "disable": false,
  "channel": 'stable'
}

```

## Use

```

{
  "_id": "use",
  "theme": "theme_id",
  "form": [],
  "loadBalanceLogin": [],
  "loadBalanceStart": []
}

```

## Website

```
{
  "_id": "website",
  "auth": "auth",
  "internalError": "HTML_CONTENT",
  "permissionDenied": "HTML_CONTENT",
  "notFound": "HTML_CONTENT",
  "welcomePage": "HTML_CONTENT",
  "homePage": "session" | "shortcut",
  "showTimedOutSessions": false
}
```

## Permissions

Configures the permissions of the users in FastX

### Standalone File

/var/fastx/local/store/permissions-store.db

### Cluster Collection

permissions-store

## Content

### Admin Users

```
{
  "_id": "admin-user",
  "data": [], /* array of admin user names */
}
```

### Admin Groups

```
{
  "_id": "admin-group",
  "data": [], /* array of admin linux groups */
}
```

## Manager Users

```
{
  "_id": "manager-user",
  "data": [], /* array of manager user names */
}
```

## Manage Groups

```
{
  "_id": "manager-group",
  "data": [], /* array of manager linux groups */
}
```

## Full-User Users

```
{
  "_id": "full-user-user",
  "data": [], /* array of full user names */
}
```

## Full-User Groups

```
{
  "_id": "full-user-group",
  "data": [], /* array of full user linux groups */
}
```

## Disabled Users

```
{
  "_id": "disable-user",
  "data": [], /* array of users who are disabled */
}
```

## Disabled Groups

```
{
  "_id": "disable-group",
  "data": [], /* array of linux groups that are disabled */
}
```

## Enable Managers

```
{
  "_id": "manager-enable",
  "data": true, /* enable manager permissions */
}
```

## Enable Limited Users

```
{
  "_id": "limited-user-enable",
  "data": true, /* enable limited user permissions */
}
```

## Manager Permissions

```
{
  "_id": "manager-permissions",
  "data": { /* the permissions for the managers */
    "session": {
      "info": true,
      "connect": true,
      "disconnect": true,
      "params": true,
      "exec": true,
      "log": true,
      "purge": true,
      "shortcut": true,
      "terminate": true
    },
  },
}
```

```
    "user": {
      "profile": true,
      "sshKeys": true,
      "logins": true
    },
    "server": {
      "info": true
    }
  }
}
```

### Limited User Permissions

```
{
  "_id": "limited-user-permissions",
  "data": { /* the permissions for the limited user */
    "start": {
      "bookmark": true,
      "custom": true
    },
    "bookmark": {
      "system": true,
      "user": true,
      "history": true,
      "favorites": true
    },
    "session": {
      "info": true,
      "connect": true,
      "disconnect": true,
      "shortcut": true,
      "params": true,
      "exec": true,
      "log": true,
      "purge": true,
      "terminate": true
    },
    "user": {
      "profile": true,
      "sshKeys": true,
      "logins": true
    },
  },
}
```

## Bookmarks

### Standalone File

/var/fastx/local/store/system-bookmark-store.db

### Cluster Collection

system-bookmark-store

## Content

List of objects of the format

```
{
  "_id": "STRING_VALUE", /* Bookmark ID. */
  "command": STRING_VALUE, /* Command to execute */
  "icon": "STRING_VALUE", /* Optional icon to display */
  "name": "STRING_VALUE", /* Bookmark name */
  "profile": "STRING_VALUE", /* optional id of Profile that will be used
when launching session */
  "tags": ["tag1", "tag2", ...], /* Optional array of tags */
}
```

## Events

### Standalone File

/var/fastx/local/store/system-api-store.db

### Cluster Collection

system-api-store

## Content

List of objects of the format

```
{
  "_id": "EventID",
```

```
"data" {
  "_id": "EventID",
  "script": "function(o) {}", /* Javascript function */
}
}
```

## Forms

### Standalone File

/var/fastx/local/store/system-forms-store.db

### Cluster Collection

system-forms-store

### Content

List of objects of the format

```
{
  "_id": "STRING_VALUE", /* Form Id */
  "data": {
    "name": "STRING", /* Display name of the form */
    "html": "<div></div>", /* HTML of the form */
    "allowUnsafeHtml": false, /* allow unsanitized HTML/javascript in the
form */
    "static": {}, /* Static data json object used in filter functions */
  }
}
```

## Http Headers

### Standalone File

/var/fastx/local/store/system-http-headers-store.db

### Cluster Collection

system-http-headers-store

## Content

List of objects of the format

```
{
  "_id": "STRING_VALUE", /* Header Id */
  "data": {
    "header": "STRING_VALUE", /* HTTP Header */
    "value": "STRING_VALUE", /* HTTP Header Value */
    "path": "^.*$", /* Regular expression defining where to set the header
*/
  }
}
```

## Load Balancing Functions

Standalone File

/var/fastx/local/store/system-load-balance-store.db

Cluster Collection

system-load-balance-store

## Content

List of objects of the format

```
{
  "_id": "STRING_VALUE", /* Load Balancing ID */
  "data": {
    "name": "STRING_VALUE", /* Name of the load balancing script */
    "script": "STRING_VALUE, /* Load balancing function */
  }
}
```

## Metrics

Standalone File

/var/fastx/local/store/system-metrics-store.db



## Cluster Collection

system-metrics-store

## Content

List of objects of the format

```
{
  "_id": "STRING_VALUE", /* Metric Script ID */
  "data": {
    "metric": "STRING_VALUE", /* Metric Id */
    "command": "STRING_VALUE", /* Metric Function */
  }
}
```

## Notifications

### Standalone File

/var/fastx/local/store/system-notifications-store.db

## Cluster Collection

system-notifications-store

## Content

List of objects of the format

```
{
  "_id": "STRING_VALUE", /* Notification ID */
  "data": {
    "title": "STRING_VALUE", /* title */
    "summary": "STRING_VALUE", /* summary */
    "content": "STRING_VALUE", /* content */
    "public": false, /* is this in the public notifications */
  }
}
```

# Profiles

## Standalone File

/var/fastx/local/store/system-profile-store.db

## Cluster Collection

system-profile-store

## Content

List of objects of the format

```
{
  "_id": "STRING_VALUE", /* Profile Id */
  "data": {
    "name": "STRING_VALUE", /* Display name */
    "description": "STRING_VALUE", /* Display description */
    "session": {
      "clientcomm": ["fx-url", "fx-notify" ],
      "clientClipboard": true, /* Enable Client Clipboard support */
      "serverClipboard": true, /* Enable Server Clipboard support */
      "cpuPercentage" 50, /* Max CPU the FastX Compressor will use */
      "disconnectTimeout": 0, /* Terminate session after N minutes
disconnected. 0 disables timeout */
      "enableVirtualGLDetection": true, /* try to detect virtualgl
detection */
      "iglx": false, /* Enable indirect GLX for older OpenGL libraries */
      "listenTCP": true, /* Enable X server to listen on a TCP port.
Also sets the $DISPLAY */
      "logLevel": 3, /* Set the logging for X server */
      "multisurface": "auto", /* ["auto", true, false] enable
multisurface */
      "offlineSharing": false, /* Enable Offline sharing. Master does
not have to connected for users to log in */
      "sharing": false, /* Enable Session Sharing */
      "saveXauthDefault": false, /* Save the Xauthority file to the
default location */
      "videoRam": 1280000, /* Maximum RAM for video buffer. Higher
numbers allow for larger max resolution */
      "virtualGLOptions": "", /* Custom virtualGL options */
      "clipboardMax": 0, /* Maximum number of byte that can be sent from
server to client */
    }
  }
}
```

```

    "debug": {
      "license": false,
      "clipboard": false
    }
  },
  "events": {
    "start": { "script": "/* custom shell commands */" },
    "connect": { "script": "/* custom shell commands */" },
    "disconnect": { "script": "/* custom shell commands */" },
    "params": { "script": "/* custom shell commands */" },
    "terminate": { "script": "/* custom shell commands */" },
  }
  "static": { /* custom static data object */ }
}
}

```

## Themes

### Standalone File

/var/fastx/local/store/system-theme-store.db

### Cluster

system-theme-store

## Content

List of objects of the format

```

{
  "_id": "STRING_VALUE", /* Theme ID */
  "data": {
    "name": "STRING_VALUE", /* Display name of the theme */
    "description": "STRING_VALUE", /* Description of the theme */
    "background": { /* See CSS Background Property for more details */
      "image": "STRING_VALUE", /* Background Image */
      "attachment": "STRING_VALUE", /* Background Attachment */
      "clip": "STRING_VALUE", /* Background Clipping region
      "color": "STRING_VALUE", /* Background Color */
      "origin": "STRING_VALUE", /* Background Origin */
      "position": "STRING_VALUE", /* Background Position */
      "repeat": "STRING_VALUE", /* Background repeat */
    }
  }
}

```

```
    },
    "brand": {
      "logo": "STRING_VALUE", /* Logo Image (usually a dataurl) */
      "tagline": "STRING_VALUE", /* Display text to be used with the theme */
    },
    "colors": { /* All colors are CSS Color Values in strings */
      "primary": "STRING_VALUE",
      "secondary": "STRING_VALUE",
      "info": "STRING_VALUE",
      "success": "STRING_VALUE",
      "warning": "STRING_VALUE",
      "danger": "STRING_VALUE",
      "light": "STRING_VALUE",
      "dark": "STRING_VALUE",
    },
    "sass": "STRING_VALUE", /* Custom SASS configuration to add to the theme */
  }
}
```

StarNet Communications Corp.

4677 Old Ironside Drive, Suite 210, Santa Clara, California 95054-1825 • USA  
408.739.0881 • 408.739.0936 • sales@starnet.com • www.starnet.com