

Jarvis Web Gateway

Installation Instructions (Also Tracker & Demo Applications)

Jonathan Couper-Smartt

jarvis@nsquared.co.nz

Abstract: The Jarvis Web Gateway is a lightweight web-service designed to give Rich Internet Applications a rapid, powerful and secure mechanism to access and update server-side databases. Jarvis can be configured to use Apache's "mod_perl" for better performance than simple CGI.

Jarvis supports fetch, create, update and delete with additional support for custom plug-in features. Jarvis is based on the RESTful approach to http web services. It provides JSON, XML interfaces, with additional support for CSV and MS Excel downloads.

Jarvis works excellently with JavaScript web apps using ExtJS, Dojo and many other JS toolkits, and is also ideal for richer client applications such as those written in Adobe Flex.

Jarvis supports any database for which a Perl DBI driver exists. This include Oracle, PostgreSQL, MySQL, SQL Server, SQLite and more.

Table of Contents

1 Licence.....	3
1.1 GNU Lesser General Public License.....	3
1.2 Clarification of Intentions.....	3
1.3 Jarvis Tracker Icons.....	3
2 Installation – Debian.....	4
2.1 Package.....	4
2.2 Install Additional Perl Modules.....	4
2.3 Apache Configuration – Without mod_perl.....	5
2.4 Apache Configuration – With mod_perl.....	5
2.5 Apache Configuration – With mod_perl & Drupal.....	6
3 Installation – Solaris.....	7
3.1 Package.....	7
3.2 Install Additional Perl Modules.....	7
3.3 Apache Configuration – Without mod_perl.....	8
3.4 Apache Configuration – With mod_perl.....	8
3.5 Apache Configuration – With mod_perl & Drupal.....	8
4 Apache Configuration & Testing.....	9
4.1 Web-Server.....	9
4.2 Apache Configuration – Without mod_perl.....	9
4.3 Apache Configuration – With mod_perl.....	9
4.4 Test.....	10
5 Relocating Jarvis.....	11
5.1 Relocation.....	11
6 Tracking Application.....	12
6.1 Tracking Requests & Errors.....	12
6.2 The Tracker Database.....	12
6.3 The Tracker Display Application.....	12
7 Demo Application.....	15
7.1 Introduction.....	15
7.2 Complete Basic Jarvis Configuration.....	15
7.3 The Demo Database File.....	15
7.4 Configure Demo Web-Service.....	15

Version Tracking (Since v5.2.0):

Version	Date	By	Comment
5.2.0	21-Feb-2012	JXC	Add Solaris support. Separate Install/Demo/Tracker docs.
5.3.0	03-Apr-2012	HK	Tracker uses postgres database.

1 Licence

1.1 GNU Lesser General Public License

This documentation is part of the Jarvis WebApp/Database gateway utility.

Jarvis (including documentation) is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

Jarvis is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with Jarvis. If not, see <<http://www.gnu.org/licenses/>>.

This software (including documentation) is Copyright 2011 by Jonathan Couper-Smartt.

1.2 Clarification of Intentions

The intention of the releasing under the LGPL (rather than the GPL) is to provide additional freedom to allow you to develop Exec and Plugin modules.

Such Exec and Plugin modules are considered to be part of your “Application” and not part of Jarvis, and are thus excluded from the “Minimal Corresponding Source” under the terms of the agreement.

1.3 Jarvis Tracker Icons

The Jarvis Tracker icons, as can be found in the 'tracker/htdocs/style' directory are copyright FamFamFam (<http://www.famfamfam.com/lab/icons/silk/>), and are used under the Creative Commons Attribution License v3.0 (<http://creativecommons.org/licenses/by/3.0/>). We extend our thanks to the creator of these icons for their quality and availability.

2 Installation – Debian

2.1 Package

Jarvis has support for Debian packages, suitable for Ubuntu and other popular Debian distributions. The Jarvis Debian package can be installed with the `dpkg` command, e.g:

```
# dpkg --install jarvis_5.0.1-1_all.deb
```

Note: If pre-requisite Perl modules are not installed, this may fail to complete successfully. If this is the case, install the dependencies (which would be listed in the output of the “`dpkg --install`” command) via `apt-get`:

```
# apt-get -f install
```

This will install the necessary packaging automatically, along with completing the install of Jarvis. The following directory locations will be populated:

Path	Notes
<code>/usr/share/jarvis/cgi-bin</code>	This contains <i>agent.pl</i> , which is the CGI script which accepts requests over http/https when not using <code>mod_perl</code> . This is a simple bootstrap program. Most of the functionality is in the <code>lib/Jarvis/*.pm</code> files in the <code>lib</code> directory.
<code>/usr/share/jarvis/demo</code>	Contains demo files which comprise a simple application showing how to use Jarvis with ExtJS and SQLite.
<code>/usr/share/doc/jarvis</code>	Documentation including this file.
<code>/usr/share/jarvis/htdocs</code>	This contains a helper JavaScript include file for use with ExtJS.
<code>/usr/share/jarvis/lib</code>	Contains the majority of the Jarvis functionality as Perl modules.
<code>/var/lib/jarvis</code>	Working directory for SQLite database files for Tracker and Demo applications, also recommended location for CGI session files.
<code>/etc/jarvis</code>	Your application top-level configuration XML files will go here.
<code>/etc/jarvis/apache</code>	Apache configuration files.

2.2 Install Additional Perl Modules

Jarvis is written in Perl, and relies on some additional Perl modules being installed. You can install these via CPAN, or by the following `apt-get` commands.

REQUIRED

```
sudo apt-get install apache2 \
libcgi-session-perl \
libxml-smart-perl \
libjson-perl \
libtext-csv-perl \
libio-string-perl \
```

```
libmime-types-perl \  
libdbi-perl \  
libdbd-sqlite3-perl \  
libsoap-lite-perl
```

REQUIRED (if using mod_perl)

```
sudo apt-get install libapache2-mod-perl2 \  
libapache-dbi-perl \  
libapache2-request-perl
```

OPTIONAL

```
sudo apt-get install libdbd-sybase-perl      (DB = SQL Server)  
sudo apt-get install libdbd-pg-perl         (DB = PostgreSQL)  
sudo apt-get install libnet-ldap-perl       (Login = LDAP or ActiveDirectory)  
sudo apt-get install libcrypt-eksblowfish-perl  
                                           (Login = Database, Authentication = 'eksblowfish')
```

OPTIONAL (required to use format=xlsx on requests)

```
CPAN Excel::Writer::XLSX  
http://search.cpan.org/~jmcnamara/Excel-Writer-XLSX-0.46/
```

2.3 Apache Configuration – Without mod_perl

If performance is not a significant factor, you can simply run Jarvis as a CGI script.

The Debian package performs this configuration automatically, with a soft link:

```
# cp /etc/jarvis/apache/jarvis.conf /etc/apache2/conf-available/jarvis.conf  
# cd /etc/apache2/conf-enabled/  
# ln -s ../conf-available/jarvis.conf .
```

The Apache instructions are as follows:

```
ScriptAlias /jarvis-agent "/usr/share/jarvis/cgi-bin/agent.pl"  
Alias /jarvis/ "/usr/share/jarvis/htdocs/"  
Alias /jarvis-demo/ "/usr/share/jarvis/demo/htdocs/"
```

If using Apache 2.4, you are likely to need to remove some restrictions. Firstly, grant access to the appropriate directory for Jarvis. In /etc/apache2/conf-available/jarvis.conf, add:

```
<Directory /usr/share/jarvis/>  
Options FollowSymLinks  
AllowOverride None  
Require all granted  
</Directory>
```

Secondly, CGI must be explicitly enabled:

```
# a2enmod cgi
```

You may need to restart the apache webserver:

```
# apache2ctl restart
```

2.4 Apache Configuration – With mod_perl

Jarvis supports the Apache mod_perl mechanism. Without mod_perl, the webserver must compile and execute the Perl code for each request. Using Apache mod_perl, the script is compiled once.

Database connections can also be re-used, and performance is generally superior. See the mod_perl page at <http://perl.apache.org/> for further details. Configuration is as follows:

```
# cd /etc/apache2/conf.d
# rm jarvis.conf
# ln -s /etc/jarvis/apache/jarvis-mod-perl.conf ./jarvis.conf
# apache2ctl restart
```

2.5 Apache Configuration – With mod_perl & Drupal

Note that if you are co-hosting Jarvis on a server running drupal from the root path, you will probably need to exclude jarvis-agent from the drupal rewrite rule. E.g. something like the following needs to be added to /usr/share/drupal6/.htaccess.

```
RewriteCond %{REQUEST_URI} !^/jarvis-agent/
```

Otherwise you will find that Drupal attempts to handle your jarvis-agent requests. Note that this is only required when running with mod_perl. When running via a simple ScriptAlias (see below), Apache will give priority to the Alias match over the Drupal match, and the problem does not arise.

3 Installation – Solaris

3.1 Package

Jarvis has support for Solaris 10 or later.

The Jarvis Debian package can be installed with the pkgadd command, e.g:

```
# gunzip jarvis-5.2.0-1.svr4.tar.gz
# tar xvf jarvis-5.2.0-1.svr4.tar
# pkgadd -d jarvis-5.2.0-1.svr4
```

The following directory locations will be populated:

/opt/jarvis/cgi-bin	This contains jarvis-agent, which is the CGI script which accepts requests over http/https when not using mod_perl. The jarvis-agent file is relatively simple. Most of the functionality is in the lib/Jarvis/*.pm files in the lib directory.
/opt/jarvis/demo	Contains demo files which comprise a simple application showing how to use Jarvis with ExtJS and PostgreSQL.
/opt/jarvis/docs	Documentation including this file.
/opt/jarvis/htdocs	This contains a helper JavaScript include file for use with ExtJS.
/opt/jarvis/lib	Contains the majority of the Jarvis functionality as Perl modules.
/opt/jarvis/etc	Sample Apache configuration files.
/var/jarvis	Working directory for SQLite database files for Tracker and Demo applications, also recommended location for CGI session files.
/etc/jarvis	Your application top-level configuration XML files will go here.

3.2 Install Additional Perl Modules

Jarvis is written in Perl, and relies on some additional Perl modules being installed. You can install these via CPAN, or by the following apt-get commands.

REQUIRED

```
sudo apt-get install apache2 \
libcgi-session-perl \
libxml-smart-perl \
libjson-perl \
libtext-csv-perl \
libio-string-perl \
libmime-types-perl \
libdbi-perl \
libdbd-sqlite3-perl \
libsoap-lite-perl \
libjson-pp-perl
```

REQUIRED (if using mod_perl)

```
sudo apt-get install libapache2-mod-perl2 \  
libapache2-dbi-perl \  
libapache2-request-perl \  

```

OPTIONAL

```
sudo apt-get install libdbd-sybase-perl      (DB = SQL Server)  
sudo apt-get install libdbd-pg-perl         (DB = PostgreSQL)  
sudo apt-get install libnet-ldap-perl       (Login = LDAP or ActiveDirectory)  
sudo apt-get install libcrypt-eksblowfish-perl  
                                           (Login = Database, Authentication = 'eksblowfish')
```

Once these packages are installed, proceed to the “Apache Configuration & Testing” step.

3.3 Apache Configuration – Without mod_perl

If performance is not a significant factor, you can simply run Jarvis as a CGI script.

The Debian package performs this configuration automatically, with a soft link:

```
/etc/apache2/conf.d/jarvis.conf -> /etc/jarvis/apache/jarvis.conf
```

The Apache instructions are as follows:

```
ScriptAlias /jarvis-agent "/usr/share/jarvis/cgi-bin/agent.pl"  
Alias /jarvis/ "/usr/share/jarvis/htdocs/"  
Alias /jarvis-demo/ "/usr/share/jarvis/demo/htdocs/"
```

You may need to restart the apache webserver:

```
# apache2ctl restart
```

3.4 Apache Configuration – With mod_perl

Jarvis supports the Apache mod_perl mechanism. Without mod_perl, the webserver must compile and execute the Perl code for each request. Using Apache mod_perl, the script is compiled once.

Database connections can also be re-used, and performance is generally superior. See the mod_perl page at <http://perl.apache.org/> for further details. Configuration is as follows:

```
# cd /etc/apache2/conf.d  
# rm jarvis.conf  
# ln -s /etc/jarvis/apache/jarvis-mod-perl.conf ./jarvis.conf  
# apache2ctl restart
```

3.5 Apache Configuration – With mod_perl & Drupal

Note that if you are co-hosting Jarvis on a server running drupal from the root path, you will probably need to exclude jarvis-agent from the drupal rewrite rule. E.g. something like the following needs to be added to /usr/share/drupal6/.htaccess.

```
RewriteCond %{REQUEST_URI} !^/jarvis-agent/
```


Otherwise you will find that Drupal attempts to handle your jarvis-agent requests. Note that this is only required when running with `mod_perl`. When running via a simple `ScriptAlias` (see below), Apache will give priority to the `Alias` match over the `Drupal` match, and the problem does not arise.

4 Apache Configuration & Testing

4.1 Web-Server

The final task is to configure your web-server to give access to the Jarvis application.

Jarvis is a Perl service, and can operate under any web-server which supports Perl scripts. These include Microsoft IIS, Apache HTTPD and others.

Documentation is provided here for Apache 2.0 or greater. It should be able to be adapted easily to any other web-server's configuration mechanism.

4.2 Apache Configuration – Without mod_perl

If performance is not a significant factor, you can simply run Jarvis as a CGI script without using mod_perl.

For Debian:

Under Debian, the package setup should be performed automatically. The package installation makes a soft link:

```
/etc/apache2/conf.d/jarvis.conf -> /etc/jarvis/apache/jarvis.conf
```

The Apache instructions are as follows:

```
ScriptAlias /jarvis-agent "/usr/share/jarvis/cgi-bin/agent.pl"  
Alias /jarvis/ "/usr/share/jarvis/htdocs/"  
Alias /jarvis-demo/ "/usr/share/jarvis/demo/htdocs/"
```

You may need to restart the apache webserver:

```
# apache2ctl restart
```

For Other Systems:

```
ScriptAlias /jarvis-agent "/opt/jarvis/cgi-bin/agent.pl"  
Alias /jarvis/ "/opt/jarvis/htdocs/"
```

These lines are contained in the supplied "apache/jarvis.conf" file. Under Linux you can generally copy or soft-link this file into the /etc/apache2/conf.d directory.

4.3 Apache Configuration – With mod_perl

Jarvis supports the Apache mod_perl mechanism. Without mod_perl, the webserver must compile and execute the Perl code for each request. Equally, without mod_perl, a new database connection must be made for each request.

Using Apache's mod_perl, the script is compiled once, and re-executed for each request. Database connections can also be re-used. The overhead for each request is greatly reduced. Refer to the mod_perl home page (<http://perl.apache.org/>) for further details.

Once mod_perl is configured and available, add the following lines to your server configuration.

Note: When installing the Debian package, this setup is done automatically.

For Debian:

```
PerlRequire /etc/jarvis/apache/startup.pl
<Location ~ "/jarvis-agent/.*">
    SetHandler perl-script
    PerlResponseHandler Jarvis::Agent
</Location>

Alias /jarvis/ "/usr/share/jarvis/htdocs/"
```

For Other Systems:

```
PerlRequire /opt/jarvis/etc/apache/startup.pl
<Location ~ "/jarvis-agent/.*">
    SetHandler perl-script
    PerlResponseHandler Jarvis::Agent
</Location>

Alias /jarvis/ "/opt/jarvis/htdocs/"
```

These lines are contained in the supplied "apache/jarvis-mod-perl.conf" file. Under Linux you can generally copy or soft-link this file into the /etc/apache2/conf.d directory.

Note that if you are co-hosting Jarvis on a server running drupal from the root path, you will probably need to exclude jarvis-agent from the drupal rewrite rule. E.g. something like the following needs to be added to /usr/share/drupal6/.htaccess.

```
RewriteCond %{REQUEST_URI} !^/jarvis-agent/
```

Otherwise you will find that Drupal attempts to handle your jarvis-agent requests. Note that this is only required when running with mod_perl. When running via a simple ScriptAlias (see below), Apache will give priority to the Alias match over the Drupal match, and the problem does not arise.

4.4 Test

To test that Jarvis is correctly installed, point your browser at the installed machine (e.g. localhost) with the following URL.

http://localhost/jarvis-agent/

You should see the error response.

```
Missing path info.  Send http://.../jarvis-agent/<app-name>/<dataset-name> in
URI!
```

This indicates that Jarvis is installed, and is ready to be configured for an application.

5 Relocating Jarvis

5.1 Relocation

The standard packages install Jarvis into the default locations for your system. Typically, these packages should work “out of the box”. However, there are some cases where you may wish to install Jarvis into non-default locations. Specifically:

- When installing on a platform for which there is no standard Jarvis package.
- When installing a test system, or where multiple versions of Jarvis must co-exist.

If you install Jarvis into another location, then you must change the Jarvis installation paths in the following locations:

- **Modify the Apache configuration (as described in the Apache Configuration chapter).**
- **Modify the Jarvis module path in `etc/apache/startup.pl` (only if using `mod_perl`).**

The `startup.pl` script is a Perl fragment which tells `mod_perl` where to find the module containing the `mod_perl` handler for the Jarvis web service agent. When relocating Jarvis, modify the path to specify the new location of the Jarvis lib directory.

By default it contains the following.

```
use Apache::DBI;

use lib qw(/usr/share/jarvis/lib);      # For Debian.
use lib qw(/opt/jarvis/lib);           # For other systems.

1;
```

6 Tracking Application

6.1 Tracking Requests & Errors

Included in the Jarvis distribution is a mechanism for tracking user requests and associated errors. This is configured in the configuration file e.g. as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<jarvis>
  <app format="json" debug="yes" dump="yes">
    <tracker logins="yes" requests="yes" errors="yes"/>
    <database name="tracker" connect="dbi:Pg:dbname=tracker;host=localhost"
username="tracker_owner" password="tracker_owner"/>
```

The parameters are.

Attribute	Default	Notes
logins	No	Should Jarvis track logins in the tracker database?
requests	No	Should Jarvis track requests in the tracker database?
errors	No	Should Jarvis track errors in the tracker database?

Table 1: Tracker Configuration Parameters

6.2 The Tracker Database

Events are tracked in a postgres database. Assuming you have postgres installed, you can create and initialize it by running the following files (all found in the tracker/db folder):

- create_database.psql
- schema.psql
- all files in the functions-views subfolder

Also update the application configuration files of any Jarvis app wishing to record tracker data.

6.3 The Tracker Display Application

When Jarvis is used, each request and/or error will cause a record to be written into the database tables in tracker.db. You can access these tables for your own purposes. Use your favorite postgres utility to view the schema and content of the tables “request”, “error” and “login”.

Alternatively, use the “tracker” application provided with Jarvis. This is a application written using ExtJS. It is installed by default, but for security purposes it must be configured before use.

6.3.1 Enabling the Display Application

Enabling the tracker display application requires both Apache configuration to provide access to the HTTP documents, and Jarvis configuration for the application to access the tracker database.

The tracker display application requires a number of supporting projects to be installed:

- ExtJS 3.3. Please see below for further details on this.

- SHJS. Syntax Highlighting for Javascript. Please see below for further details on this.
- Protovis. For graphs. Please see below for further details on this.

Once these three pre-requisites are installed, the tracker display application can be configured and will be then available. To configure the tracker application itself, first, enable the display application in Apache: The following Apache configuration is sufficient:

For Debian:

```
Alias /jarvis-tracker/ "/usr/share/jarvis/tracker/htdocs/"
```

For Other Systems:

```
Alias /jarvis-tracker/ "/opt/jarvis/tracker/htdocs/"
```

Configure Jarvis to recognise the tracker. Copy the tracker's example Jarvis application configuration file from “tracker/etc/tracker.xml” to the Jarvis etc directory:

For Debian:

```
cp /usr/share/jarvis/tracker/etc/tracker.xml /etc/jarvis/
```

For Other Systems:

```
cp /opt/jarvis/tracker/etc/tracker.xml /opt/jarvis/etc/
```

Edit the configuration file to ensure all the paths and other configuration are correct. Note that this example file comes with configuration to log Jarvis requests and errors into the tracker database.

Reload Apache's configuration, then point your browser at the following:

```
http://localhost/jarvis-tracker/
```

6.3.2 Installing Tracker Display Application Support Files

The front-end for the demo application is written in JavaScript using the ExtJS and also utilises a number of other libraries for various features of the display application.

ExtJS

Please download the 3.3 release of ExtJS from <http://www.sencha.com/products/js/download.php> and install. The file "ext-all.js" should be accessible by the URL "/ext-3.3/ext-all.js". E.g the Apache configuration may be:

```
Alias /ext-3.3/ "/opt/extjs-3.3/"
```

PROTOVIS

To make the Protovis library available on your server, simply download the SVGWeb enabled Protovis 3.2 or later from <http://gitorious.org/~jlove/protovis/jloves-protovis> and unpack it on your machine. Then configure Apache to serve the protovis files through the alias “/protovis”.

For example, if you have unpacked Protovis into “/opt/”, then the path to protovis.js should be “/opt/protovis/protovis.js”. The following Apache configuration is sufficient.

```
Alias /protovis/ "/opt/protovis-3.2/"
```

Note that this 3.2 release of Protovis does support IE.

SHJS

Similarly for SHJS, download the latest release of SHJS from <http://shjs.sourceforge.net/> and install. For example the Apache configuration may be:

```
Alias /shjs/ "/opt/shjs-0.6/"
```

SIMILE

Similarly for SIMILE, download the 3.3.1 SIMILE “timeline” library release from <http://code.google.com/p/simile-widgets/downloads/list> and install. For example the Apache configuration may be (assuming timeline_js and timeline_ajax are installed under /usr/share/javascript):

```
Alias /javascript/ "/usr/share/javascript/"
```

The home page for the SIMILE widget is <http://www.simile-widgets.org/timeline/>.

6.3.3 Rebuilding Support Tables

Included in the tracker database are a number of tables that are used by the display application for generating data for showing in the browser. These tables are the “calendar” and “day_interval” tables.

Both these tables must have their data generated. The script “populate-generated-tables.pl” that can be found in the “tracker/db” directory can do this for you.

To use, you must first ensure that the Perl module Date::Calc is installed. Under some systems with apt-get, you can perform:

```
sudo apt-get install libdate-calc-perl
```

to do this for you.

Once this is installed, you can run the script directly:

```
perl populate-generated-tables.pl
```

Note that this script uses the default tracker database configuration. If you need to update a different database, you will need to edit the script and alter the database the DBI handler loads.

If you have data that is not within the range Jan 1st 2008 to Jan 1st 2020, then you will need to edit the script further and alter the date range the 'calendar' table is generated for.

7 Demo Application

7.1 Introduction

A simple demonstration application is provided with Jarvis. This application uses an SQLite database. When installing the Debian packages, this demonstration is automatically configured and enabled and available under the address:

```
http://localhost/jarvis-demo/
```

The following instructions are required only when not using the Debian package of Jarvis.

7.2 Complete Basic Jarvis Configuration

First, ensure that Jarvis itself is available via the `/jarvis-agent/` URL. Access the following URL:

```
http://localhost/jarvis-agent/
```

Verify that you receive the response...

```
Missing app name. Send /jarvis-agent/<app>[/<dataset>[/<arg1>...]] in URI!
```

7.3 The Demo Database File

Take a copy of the database to a directory writable by the Apache user. For example:

```
# mkdir -p /var/lib/jarvis/demo/  
# cp /opt/jarvis/demo/db/demo-empty.db /var/lib/jarvis/demo/demo.db  
# chown www-data:www-data /var/lib/jarvis/demo/demo.db
```

If this is not done correctly, you will see the error “Attempt to write to a read-only database” when saving changes via the demo screen.

7.4 Configure Demo Web-Service

Install the demo.xml Jarvis application configuration file:

```
# cp /opt/jarvis/demo/etc/demo.xml /opt/jarvis/etc/demo.xml
```

In Apache configuration create an alias to the Jarvis demo htdocs:

```
Alias /jarvis-demo/ "/opt/jarvis/demo/htdocs/"
```

Change the default values configured in demo.xml to match your installation:

- Location of the demo database file. Default is `/var/lib/jarvis/demo/demo.db`.
- Location of the datasets configuration. Default is `/usr/share/jarvis/demo/dataset`.
- Location of the session files. Default is `/var/lib/jarvis/demo/sessions`.
- Location of the plugin file. Default is `/usr/share/jarvis/demo`.