

Table of Contents

FCR Manual Installation.....	1
Dependencies.....	1
Database.....	1
httpd.....	2
FCR Portal configuration.....	4
Cron jobs.....	4

FCR Manual Installation

During the installation there are a few areas that must be installed/configured.

- database
- httpd
- FCR Portal software
- cron jobs

Dependencies

FCR depends on several `perl` modules. These are available in the CERN SAM apt/yum repository.

```
perl-DBI
perl-DBD-Oracle
perl-AppConfig
mod_perl
perl-Apache-Session
perl-Apache-DBI
perl-IPC-Shareable
perl-Log-Dispatch
perl-Log-Log4perl

oracle-instantclient-basic
oracle-instantclient-tnsnames.ora
oracle-instantclient-sqlplus
oracle-instantclient-devel
```

The configuration we're using on the production servers at CERN right now is the following:

```
perl-DBI-1.40-8
perl-DBD-Oracle-1.16-4.el4
perl-AppConfig-caf-1.4.12-1
mod_perl-1.99_16-4.5
perl-Apache-Session-1.80-1.2.el4.rf
perl-Apache-DBI-0.94-4
perl-IPC-Shareable-0.6-1
perl-Log-Dispatch-2.11-1
perl-Log-Log4perl-1.06-1

oracle-instantclient-basic-10.2.0.3-2.sl4
oracle-instantclient-tnsnames.ora-10.2.0.1-1.cern
oracle-instantclient-sqlplus-10.2.0.3-2.sl4
oracle-instantclient-devel-10.2.0.3-2.sl4
```

- **Warning:**

- ◆ Not all `perl` oracle-instantclient versions are compatible with the `perl` DB modules :-/ The versions we are running right now seem to be compatible, but this is not always the case...
- ◆ Also, we've observed that oracle-instantclient doesn't take in account other `tnsnames.ora` files, than the one at `/etc`, even if theoretically such a configuration would be supported.

Database

FCR sharing the database with SAM. FCR preferably should have a separate database service than SAM. Both applications are using Oracle databases. Some of the FCR tables have currently reside on the SAM DB so the BDII2Oracle tool could easily update them.

Included in the FCR software there are scripts to create the necessary tables and synonyms in case of having the SAM database separated from FCR. There are scripts to take care of granting sufficient privileges to the tables/synonyms.

In the CERN production instance we have 3 DB accounts to use:

- admin account (LCG_FCR): to perform admin operations on the DB (creating, dropping tables, etc.)
- read-write account (LCG_FCR_PORTAL_VO): for the FCR Admin portal, to enable operations performed by VO responsables (modifications on Critical Test list, black-whitelisting, etc)
- read account (LCG_FCR_PORTAL_USER): for the FCR User portal to query current status of services

The FCR DB scripts suppose that you have the same usernames set up in your DB where you want to install FCR.

The DB scripts are available in the /db directory under \$FCR_HOME. See FCR Production Services in order to understand more about the purpose of these tables, triggers, synonyms. (In the lists below, ordering numbers indicate the sequence how the scripts should be executed one after the other.)

Scripts to be executed by the SAM DB user

- 1. create-sam-priv.sql: privileges for FCR users for FCR-related tables in the SAM namespace
- 6. create-sam-syn.sql: synonyms for FCR tables that should be modified by trigger in SAM DB
- 7. create-sam-triggers.sql: triggers for the SAM tables

Scripts to be executed by the FCR DB user

- 2. create-fcr-root-syn.sql: synonyms for the FCR DB root user
- 3. create-fcr-tables.sql: FCR-specific tables (see FCR Production Service in order to understand more about the purpose of these tables)
- 4. create-fcr--priv.sql: privileges on FCR-portal-specific tables (storing user settings, etc.) for the FCR portal DB users
- 5. create-fcr-triggers.sql: triggers for the SAM tables

Scripts to be executed by the FCR portal users

- 8. create-fcr-user-syn.sql: synonyms for the FCR portal users

httpd

At FCR Production Services you find information about the layout required to run an FCR server in terms of httpd web server configuration.

(At least) 2 virtual hosts have to be configured for the HTTPS and HTTP accesses. So a file like /etc/httpd/conf.d/fcr.conf should contain at least the following:

```
# Application fcr
NameVirtualHost *:8083
NameVirtualHost *:8443
NameVirtualHost <hostname>:80
Listen 0.0.0.0:8083

LoadModule perl_module modules/mod_perl.so

<VirtualHost *:8443>

    ServerName lcg-fcr.cern.ch
```

FcrInstNoQuat < LCG < TWiki

```
Alias /fcr /opt/lcg/FCR/cgi
Alias /fcr-styles /opt/lcg/FCR/styles
Alias /fcr-data /opt/lcg/FCR/bdii-data

DocumentRoot /var/www/html

ErrorLog logs/ssl_error_log
CustomLog logs/ssl_request_log "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
TransferLog logs/ssl_access_log

SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown downgrade-1.0 force-response-

# Module perl

PerlModule ModPerl::Registry
PerlPassEnv TLS_ADMIN
PerlPassEnv ORACLE_HOME
PerlOptions +Parent
PerlSwitches -Mlib=/opt/lcg/FCR/lib/perl
PerlWarn On

# Module ssl

SSLCertificateKeyFile /etc/grid-security/hostkey.pem
SSLCACertificatePath /etc/grid-security/certificates
SSLCertificateFile /etc/grid-security/hostcert.pem
SSLCACRevocationPath /etc/grid-security/certificates

SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP
SSLEngine on
SSLVerifyClient require
SSLVerifyDepth 10

<Directory "/var/www/cgi-bin">
    # Module ssl
    SSLOptions +StdEnvVars
</Directory>

<Location /fcr>
    SetHandler perl-script

    Options +ExecCGI

    # Module perl
    PerlHandler ModPerl::Registry

    # Module ssl
    SSLRequireSSL
</Location>

<Files ~ "\.(cgi|shtml|phtml|php3?)$" >
    # Module ssl
    SSLOptions +StdEnvVars
</Files>

</VirtualHost>

<VirtualHost *:8083 <hostname>:80>

    ServerName lcg-fcr.cern.ch
    Alias /fcr-data /opt/lcg/FCR/bdii-data
    DocumentRoot /var/www/html

</VirtualHost>
```

In order to help `httpd` configuration, a template file is available at `/opt/lcg/FCR/conf/fcr-httpd.conf.template`.

FCR Portal configuration

The portals have a configuration file in `/opt/lcg/FCR/conf/fcr.conf`. This is the place to define the DB connection parameters using variables:

- `dbname`: FCR DB (service) name
- `dbuser`: FCR Admin Portal database user (`LCG_FCR_PORAL_VO`)
- `dbpw`: FCR Admin Portal database user password
- `dbuseruser`: FCR User Portal database user (`LCG_FCR_PORTAL_USER`)
- `dbuserpw`: FCR User Portal database user password
- `oracle_home`: `$ORACLE_HOME`
- `tns_admin`: location of `tnsnames.ora`

`oracle_home` and `tns_admin` does not necessarily need to be configured in case of using Oracle instantclient.

The following variables should be used in order to define the nature of the portal:

- `sitestatus`=["Certified"]
- `sitetype`=["Production"]

Values should be specified in the following format (python list): `["value1", "value2", "value3"]`

The rest of the variables normally don't need to be modified.

Cron jobs

In order to have the "blacklisting" LDIF file generated regularly, `cron` jobs need to be set up. The script that contacts the DB in order to collect information about what should go into the "blacklist" is `/opt/lcg/FCR/bin/gen-exclude-ldif.pl`. In order to ease the usage of this script, a wrapper script is provided, which should be used in a `cron` jobs something like this:

```
0,10,20,30,40,50 * * * * root /opt/lcg/FCR/cron/cron-gen-exclude-ldif.sh 2>> /opt/lcg/FCR/log/ge
```

For the file to be copied to `/etc/cron.d` a template file is available at `/opt/lcg/FCR/cron/fcr-exclude-ldif_template`.

-- JuditNovak - 25 Sep 2007

This topic: LCG > FcrInstNoQuat

Topic revision: r8 - 2007-10-03 - JuditNovak



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback