DataGRID

INSTALLATION GUIDE FOR EDG REPLICA MANAGER 1.8.1

WP2 EDG REPLICA MANAGER

Document identifier:	**DataGrid-02-ERM-INSTALL-GUIDE**
EDMS id:	
Date:	February 4, 2005
Work package:	**WP2: Data Management**
Partner(s):	**CERN, PPARC, HIP, INFN, ITC, KDC**
Lead Partner:	**CERN**
Document status:	**FINAL**
Author(s):	WP2
File:	**edg-replica-manager-installguide**

Abstract: This document gives an overview of how to install the EDG Replica Manager (Reptor).

CONTENTS

Document Change Record

Issue	Date	Comment	Author
0_1_0	28 February 2003	First Draft	Kurt Stockinger
0_2_0	7 March 2003	small modifications for document IDs, software dependencies updated and configuration section added	Heinz Stockinger
0_3_0	25 March 2003	removed R-GMA related appendix, updated some package dependencies and reflect changes in configuration file	Heinz Stockinger
1_0_0	28 March 2003	clean the guide, mention the proper installation steps for LCFG.	Heinz Stockinger
2_0_0	10 April 2003	Add variables for replacement by the build system.	Levi Lucio
2_0_1	4 June 2003	software dependencies updated	Heinz Stockinger
2_0_2	13 June 2003	some minor editorial changes; corrected link to EDG software repository	Heinz Stockinger
2_0_3	16 June 2003	some additions for MDS in configuration file. Appendix updated with example configuration files	Heinz Stockinger
2_0_4	20 Sep 2003	Update for new config locations etc.	Peter Kunszt

1. INTRODUCTION

1.1. OBJECTIVES OF THIS DOCUMENT

This document describes the installation and configuration of the EDG Replica Manager (Reptor). The usage of the EDG Replica Manager is described in the user guide [A6].

1.2. APPLICATION AREA

WP2 Data Management.

1.3. APPLICABLE DOCUMENTS AND REFERENCE DOCUMENTS

Applicable documents

[A1]   D2.2 WP2 Architecture and Design Document.
[A2]   D12.2 DataGrid Naming Conventions document, WP12 Project Management.
       http://edms.cern.ch/document/328838
[A3]   D12.2 DataGrid Developers Guide, Quality Assurance Group.
       DataGrid-12-TED-358824-1-1, https://edms.cern.ch/document/358824/1.1
[A4]   D12.4 DataGrid Architecture
[A5]   D2.5 EDG Replica Manager User Guide
[A6]   D2.5 EDG Replica Optimization Service Install Guide
[A7]   D2.5 EDG Replica Location Service Install Guide
[A8]   D2.5 EDG Replica Metadata Catalog Install Guide

Reference documents

[R1]   L. Guy, E. Laure, P. Kunszt, E. Laure, H. Stockinger, K. Stockinger, Replica Management in DataGrids. Presented at GGF5.
       http://edms.cern.ch/document/350430.
[R2]   W. H. Bell, D. G. Cameron, L. Capozza, P. Millar, K. Stockinger, F. Zini. Design of a Replica Optimization Framework. Technical Report DataGrid-02-TED-021215, CERN, Geneva, Switzerland, December 2002.
[R3]   Ann Chervenak, Ewa Deelman, Ian Foster, Wolfgang Hoschek, Adriana Iamnitchi, Carl Kesselman, Peter Kunszt, Matei Ripeanu, Heinz Stockinger, Kurt Stockinger, and Brian Tierney. Giggle: A Framework for ConstructingScalable Replica Location Services. In Proc. of the Int'l. IEEE Supercomputing Conference (SC 2002), Baltimore, USA, November 2002.
[R4]   R-GMA Documentation: http://hepunx.rl.ac.uk/edg/wp3/documentation/index.html

1.4. TERMINOLOGY

Definitions

API	Application Programming Interface
CE	Computing Element
CLI	Command Line Interface
EDG	European DataGrid
RLS	Replica Location Service
RMC	Replica Metadata Catalog
ROS	Replica Optimization Service

INSTALLATION GUIDE FOR EDG REPLICA
MANAGER 1.8.1

WP2 EDG Replica Manager

Doc. Identifier:
DataGrid-02-ERM-INSTALL-GUIDE

Date: **February 4, 2005**

2. THE EDG REPLICA MANAGER

The EDG Replica Manager (Reptor) is a replica manager that is based on web service technology such as the Simple Object Access Protocol (SOAP) as well as the Java version of the Globus toolkit. In addition, Reptor uses the Replica Location Service (RLS), the Replica Metadata Catalog (RMC) and the Replica Optimization Service (ROS or Optor) to manage replication across a distributed Grid environment. In this document we concentrate on the installation and refer to the EDG Replica Manager User Guide for usage [A5]. For design details refer to [A1] and [R1].

3. EDG REPLICA MANAGER REQUIREMENTS AND INSTALLATION

In this section we give details about software packages and services that are required for running Reptor and then go into details of how to install Reptor.

3.1. REQUIRED SOFTWARE PACKAGES

The following packages (RPMs) are required to run Reptor + Optor (i.e. the EDG Replica Manager with the Replica Optimisation Service (ROS)). For version numbers see the RPMs listed in the next subsection. Note that EDG Replica Manager requires that the services **RLS**, **RMC** and **ROS** are correctly installed and configured and we refer to the respective installation guides below:

- **Java j2sdk**

- **Jakarta Tomcat4** (for RLS, RMC and ROS only)

- **edg-java-data-util**: data management utilities (CVS package name edg-java-data-util).

- **edg-java-data-util-config**: data management utilities (CVS package name edg-java-data-util).

- **edg-java-security-client**: security components (CVS package name edg-java-security).

- **edg-local-replica-catalog-client**: Replica Location Service. Please refer to the RLS Installation Guide [A7]. (CVS package name edg-rls-server)

- **edg-replica-metadata-catalog-client**: Stores additional metadata for replicas. Please refer to the RMC Installation Guide [A8]. (CVS package name edg-metadata-catalog).

- **edg-replica-optimization-client**: Replica Optimization Service (ROS or Optor). Please refer to the ROS Installation Guide [A6] (CVS package name edg-ros).

- **edg-se-webservice-client**: *optional* interface to the SE (SRM) provided by WP5.

- **edg-rgma-api-java**: *optional* interface to information service provided by WP3

- **Globus 2** Through VDT

- **CoG**: Java implementation of Globus

3.2. OBTAINING THE EDG REPLICA MANAGER SOURCES

We distribute standard EDG RPMs. The following RPMs[1] are necessary to install the EDG Replica Manager:

- edg-replica-manager-1.8.1-1.noarch.rpm

- edg-replica-manager-test-1.8.1-1.rpm

- edg-replica-manager-config-1.8.1-1.rpm

- edg-replica-manager-interface-1.8.0-1.rpm

- edg-replica-manager-gridftp-client-1.8.1-1.i386.rpm

which can be obtained from the standard EDG software repository:

```
http://datagrid.in2p3.fr/distribution/autobuild/i386-rh7.3/wp2/RPMS/
```

where you can also find all other RPMs that EDG Replica Manager depends on.

3.3. EDG REPLICA MANAGER INSTALLATION

Once all the packages and services stated above are correctly installed and configured, the EDG Replica Manager can be installed.

The EDG Replica Manager package (edg-replica-manager) is a client side package. All the RPMs may be installed as root as follows:

```
rpm -ivh edg-replica-manager-1.8.1-1.noarch.rpm\
 edg-replica-manager-config-1.8.1-1.noarch.rpm\
 edg-replica-manager-test-1.8.1-1.noarch.rpm\
 edg-replica-manager-gridftp-client-1.8.1-1.i386.rpm\
 edg-replica-manager-interface-1.8.0-1.noarch.rpm
```

This installs the package by default into /opt/edg. Non-root installation is possible if the RPM database path is specified with the option --dbpath= and the location where the package should be installed with --prefix=.

4. EDG REPLICA MANAGER DEPLOYMENT

In the section we describe how to correctly configure the package. The package can be configured (and also installed) via LCFGng as well as manually. Here $EDG_LOCATION is the installation location of the EDG Replica Manager, which is /opt/edg unless another prefix was specified with the --prefix= option during installation.

---

[1]The test RPM edg-replica-manager-test-1.8.1-1.rpm is optional

4.1. MANUAL CONFIGURATION

The config RPM installs the edg-replica-manager-configure script into the $EDG_LOCATION/sbin directory. This script can be used to automatically generate the default configuration file. Before running the script, edit the file

```
$EDG_LOCATION/etc/edg-replica-manager/edg-replica-manager.conf.values
```

It will look like

```
@EDG.LOCATION@|/opt/edg|location of edg middleware
@LOCALDOMAIN@|foo.com|the local domain
@DEFAULT.SE@|mySE.foo.com|the host of the close SE
@DEFAULT.CE@|myCE.bar.com|the host of the close CE
@INFOSERVICE@|RGMA|The info provider to use. It can be Stub, MDS or RGMA
@RLS.MODE@|LrcOnly|The mode the RLS should be run in. LrcOnly or WithRli
@STUBFILE@|info-service-stub.properties|The properties file for the \
 static file - only needed in Stub mode
@MDS.HOST@|mdshost.bar.com|The host of the MDS info provider
@MDS.PORT@|2135|The port of the MDS info provider
@ROS.FAILURE@|true|Fail if no ROS is available
@CONF.GCC@|_gcc3_2_2|The gcc suffix as used on the build box \
   (empty for 2.95, _gcc3_2_2 for 3.2.)
```

Fill in the values corresponding to your settings. An explanation of these values is found in section 4.3. Now you can run (the --verbose flag is optional)

```
$EDG_LOCATION/sbin/edg-replica-manager-configure \
   $EDG_LOCATION/etc/edg-replica-manager/edg-replica-manager.conf.values [--verbose]
```

This will generate the default config file to

```
$EDG_LOCATION/var/etc/edg-replica-manager/edg-replica-manager.conf
```

4.2. LCFGNG CONFIGURATION

The edg-replica-manager LCFGng object does the same as the manual install, but it generates the values file from the LCFG object and stores it in

```
$EDG_LOCATION/var/etc/edg-replica-manager/edg-replica-manager.conf.lcfg
```

Then it runs the edg-replica-manager-configure script.

4.3. CONFIGURATION FILE EDG-REPLICA-MANAGER.CONF

As mentioned before, the default configuration file will be in the following location:

```
$EDG_LOCATION/var/etc/edg-replica-manager/edg-replica-manager.conf
```

The parameters are:

- **location**: You can either give your default CE or default SE. This parameter is used to bootstrap the info services and to know where the client tool is located. These options have to be set in all cases and need to be configured in the post-install step described above (manual or lcfg).

- **file copy method**: The URL copy functionality is implemented in pure Java, using CoG as well as through JNI using the (native) GASS copy C library. If you don't have the globus native libraries installed, you still can use the package by switching to the CoG-based implementation. However, it comes with a performance hit: the CoG implementation does not allow for multiple parallel streams for file transfers and is generally slower. The two possible values are as follows:

  ```
  org.edg.data.reptor.storage.GlobusURLCopyCoGImpl
  org.edg.data.reptor.storage.GlobusURLCopyNativeImpl
  ```

- **GridFTP service utilities** like `exists`, `file size`, `mkdir` etc. are implemented either through CoG in pure Java or through JNI using the GridFTP C client library. Similarly as for the file copy method, the two implementations that you may use are:

  ```
  org.edg.data.reptor.storage.GridFTPCoGImpl
  org.edg.data.reptor.storage.GridFTPNativeImpl
  ```

- **Information service**: This is the main parameter to configure an information service that is then used for all for information service requests. In addition, information about services like RLS, RMC and ROS is obtained from the information service.

  EDG Replica Manager can be used with R-GMA, MDS or it can read all information from a configuration file. The following implementations are available:

  ```
  org.edg.data.reptor.info.InfoServiceStub
  org.edg.data.reptor.info.InfoServiceRGMA
  org.edg.data.reptor.info.InfoServiceMDS
  ```

  1. **Stub**: The `Stub` does not read any information from an information service but from a configuration file specified by another property, `config.infoservice`
  2. **RGMA**: If this class is used, R-GMA is contacted for all information service requests.
  3. **MDS**: If this class is used, MDS is contacted for all information service requests.

- **SRM specific properties** can be set: retry interval, timeouts etc. Please refer to Appendix A for details.

- **failIfNoROS** if this is set to false, then the replica manager will not fail if no ROS is available. By default it is set to true. Of course the optimization commands will become meaningless in this case and will exit with an error.

5. TESTING THE INSTALLATION

If you have $EDG_LOCATION/bin in your path, you should be able to invoke the edg-replica-manager command. It should display some help information.

In the test RPM (edg-replica-manager-test-1.8.1-1.rpm) you find API and CLI tests that are installed in the directory $EDG_LOCATION/sbin/test. For further information on these test programs refer to the User Guide (Section Test Suite).

6. APPENDIX A: EXAMPLE FOR EDG-REPLICA-MANAGER.CONF

A possible example for a the file `edg-replica-manager.conf` is as follows:

```
#
# Configuration for the EDG Replica Manager
#


# ================================================================
#
# Locality. You can either give your local CE or local SE.
# this is used to bootstrap the info services and to know where
# we are located.
localDomain=cern.ch
defaultCE=lxshare0313.cern.ch
defaultSE=pcrd24.cern.ch


# ================================================================
#
# The URL Copy is implemented in pure Java, using CoG as well as
# through JNI using the native GassCopy client (as globus-url-copy).
# The two implementations at your disposal are:
#
# CoG
# Native
#
# you can also set the default number of streams for small and large files,
# and the threshold in MB when a file is considered to be 'large'.

#impl.globusURLCopy=CoG
impl.globusURLCopy=Native
default.nstreams.smallfiles=1
default.nstreams.bigfiles=3
filesize.threshold=1


# ================================================================
#
# The GridFTP utilities like exists, ls, fileSize, mkdir are implemented
# either through CoG in pure Java or through JNI using the GridFTP client.
# Similarly as for the copy, the two implementations that you may use are
#
# CoG
# Native
#
# For the native implementation, the name of the JNI library interfacing
# to gridFTP needs to be set in the library.gridFTP property. This
# dynamic library will be loaded from the classpath.

#impl.gridFTP=CoG
impl.gridFTP=Native
library.gridFTP=edg_rm_gridftp


# ================================================================
#
# Information service: The implementation of the Info
# service interface.
#
```

```
# Available are:
# Stub  - local config file
# RGMA
# MDS
#

impl.infoService=Stub
#impl.infoService=RGMA
#impl.infoService=MDS


# ===============================================================
#
# Replica Catalog type: The implementation of the
# replica catalog interface.
#
# Available are:
# LrcOnly -- a single LRC per VO
# WithRli -- a full RLS deployed
#

impl.ReplicaCatalog=LrcOnly
#impl.ReplicaCatalog=WithRli


#
# The sections below are configuration options specific to the
# InfoService implementations.
#
# ---------
# InfoServiceStub : Local config file 'info provider'
# ---------
# The InfoServiceStub implementation of the info provider interface
# reads all information from the configuration file given in the property
# config.infoService.
#
config.infoService=info-service-stub.properties

# ---------
# InfoServiceRGMA : Use R-GMA
# ---------
# R-GMA initializes itself from its properties, given by an environment
# variable RGMA_PROPS, where the file Consumer.props needs to exist.
# Additional configuration options are the timeout (in seconds) to
# wait for a response.
#
rgma.timeout=6000

# ---------
# InfoServiceMDS : Use MDS
# ---------
# The mds.url property will set the endpoint of the service
# to connect to, the mds.root property sets the root element
# of the initial context.
# If another ldap context is used than the one provided by
# sun, its name can be given in the mds.ldap.contextfactory property.
#
mds.url=ldap://lxshare0242.cern.ch:2135
mds.root=mds-vo-name=local,o=grid
```

```
mds.ldap.contextfactory=com.sun.jndi.ldap.LdapCtxFactory


# ================================================================
#
# SRM specific properties.
# set the retry and timeout intervals for MSS requests (in seconds)
#
srm.prepareToGet.retryinterval=10
srm.prepareToGet.timeout=3600
srm.prepareToPut.retryinterval=10
srm.prepareToPut.timeout=300


# ================================================================
#
# Miscellaneous
#
# Failure if not ROS is available
# This may be set to 'false' if you want the replica manager
# not to complain if ROS is unavailable. The replica selection
# for getBestFile and listBestFile will be a simple random function
# and corresponds rather to getRandomFile and listRandomFile, respectively.
failIfNoROS=true
```

7. APPENDIX B: EXAMPLE FOR INFO-SERVICE-STUB.PROPERTIES

Since this is the main configuration that can be used if the information service is out of service, we give a possible example here. The file below is configured for the VO "wpsix".

Note that this file is not required if R-GMA is used.

```
#
# Not having a proper info service, this stub can be used to have an
# operational replica manager client.

#
# ROS, RMC, RLS configuration for specific VOs
#

# VO = wpsix
info.stub.wpsix.RMC = http://lxshare0342.cern.ch:8080/edg-replica-metadata-catalog \
   /services/edg-replica-metadata-catalog
info.stub.wpsix.ROS = http://lxshare0343.cern.ch:8080/edg-replica-optimization \
   /services/edg-replica-optimization
info.stub.wpsix.LRC = http://lxshare0344.cern.ch:8080/edg-local-replica-catalog \
   /services/edg-local-replica-catalog


#================================================================================
#
# CE Configuration
#
# the CE can be a comma-separated list of CE hosts.
# for each CE the closeSE's and accepted VOs can be given separately (comma-separated list).
#
info.stub.CE = lxshare0313.cern.ch,ccgridli01.in2p3.fr,\
    gppce06.gridpp.rl.ac.uk
```

```
# CE at CERN
info.stub.CE.lxshare0313.cern.ch.name = CERN
info.stub.CE.lxshare0313.cern.ch.closeSEs = pcrd24.cern.ch,testbed008.cern.ch
info.stub.CE.lxshare0313.cern.ch.VOs = test1,test2,wpsix

# CE at LYON
info.stub.CE.ccgridli01.in2p3.fr.name = LYON
info.stub.CE.ccgridli01.in2p3.fr.closeSEs = ccgridli02.in2p3.fr
info.stub.CE.ccgridli01.in2p3.fr.VOs = wpsix

# CE at RAL
info.stub.CE.gppce06.gridpp.rl.ac.uk.name = RAL
info.stub.CE.gppce06.gridpp.rl.ac.uk.closeSEs = gppse06.gridpp.rl.ac.uk
info.stub.CE.gppce06.gridpp.rl.ac.uk.VOs = wpsix


#=============================================================================
#
# SE Configuration
#
# the SE can be a comma-separated list of hosts.
# for each SE the associated LRC, comma-separated list of VOs, each storage
# directory for each VO and the comma separated list of protocols can be given.
#

info.stub.SE = pcrd24.cern.ch,testbed008.cern.ch,ccgridli02.in2p3.fr,\
   gppse06.gridpp.rl.ac.uk

# SE at CERN
#

info.stub.SE.pcrd24.cern.ch.name = CERN-SE1
info.stub.SE.pcrd24.cern.ch.type = disk
info.stub.SE.pcrd24.cern.ch.VOs = test1,test2,wpsix
info.stub.SE.pcrd24.cern.ch.test1.directory = /test1
info.stub.SE.pcrd24.cern.ch.test2.directory = /test2
info.stub.SE.pcrd24.cern.ch.wpsix.directory = /wpsix
info.stub.SE.pcrd24.cern.ch.accesspoint = /data
info.stub.SE.pcrd24.cern.ch.protocols = gsiftp
info.stub.SE.pcrd24.cern.ch.lrc = http://lxshare0344.cern.ch:8080 \
   /edg-local-replica-catalog/services/edg-local-replica-catalog
info.stub.SE.pcrd24.cern.ch.rli = http://lxshare0344.cern.ch:8080 \
   /edg-replica-location-index/services/edg-replica-location-index

# SE at CERN
#
info.stub.SE.testbed008.cern.ch.name = CERN-SE2
info.stub.SE.testbed008.cern.ch.type = disk
info.stub.SE.testbed008.cern.ch.VOs = test1,test2,wpsix
info.stub.SE.testbed008.cern.ch.test1.directory = /
info.stub.SE.testbed008.cern.ch.test2.directory = /
info.stub.SE.testbed008.cern.ch.wpsix.directory = /
info.stub.SE.testbed008.cern.ch.protocols = gsiftp
info.stub.SE.testbed008.cern.ch.accesspoint = /tmp
info.stub.SE.testbed008.cern.ch.lrc = http://lxshare0344.cern.ch:8080 \
   /edg-local-replica-catalog/services/edg-local-replica-catalog
info.stub.SE.testbed008.cern.ch.rli = http://lxshare0344.cern.ch:8080 \
```

```
    /edg-replica-location-index/services/edg-replica-location-index

# SE at LYON
#
info.stub.SE.ccgridli02.in2p3.fr.name = LYON
info.stub.SE.ccgridli02.in2p3.fr.type = disk
info.stub.SE.ccgridli02.in2p3.fr.VOs = wpsix
info.stub.SE.ccgridli02.in2p3.fr.accesspoint = /edg/StorageElement/dev2
info.stub.SE.ccgridli02.in2p3.fr.wpsix.directory = /wpsix
info.stub.SE.ccgridli02.in2p3.fr.protocols = gsiftp
info.stub.SE.ccgridli02.in2p3.fr.lrc = http://lxshare0344.cern.ch:8080 \
    /edg-local-replica-catalog/services/edg-local-replica-catalog
info.stub.SE.ccgridli02.in2p3.fr.rli = http://lxshare0344.cern.ch:8080 \
    /edg-replica-location-index/services/edg-replica-location-index

# SE at RAL
#
info.stub.SE.gppse06.gridpp.rl.ac.uk.name = RAL
info.stub.SE.gppse06.gridpp.rl.ac.uk.type = disk
info.stub.SE.gppse06.gridpp.rl.ac.uk.VOs = wpsix
info.stub.SE.gppse06.gridpp.rl.ac.uk.accesspoint = /flatfiles/06
info.stub.SE.gppse06.gridpp.rl.ac.uk.wpsix.directory = /wpsix
info.stub.SE.gppse06.gridpp.rl.ac.uk.protocols = gsiftp
info.stub.SE.gppse06.gridpp.rl.ac.uk.lrc = http://lxshare0344.cern.ch:8080 \
    /edg-local-replica-catalog/services/edg-local-replica-catalog
info.stub.SE.gppse06.gridpp.rl.ac.uk.rli = http://lxshare0344.cern.ch:8080 \
    /edg-replica-location-index/services/edg-replica-location-index
```