This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Framework Deployment Guide

Permission Prerequisites

4/26/2025

Permission Prerequisites

Contents

This section describes the minimum permissions required to install and run Management Framework components. For information about minimum permissions required for other Genesys components, refer to product- or component-specific documentation.

System Permissions

The following table provides the minimum permissions required to install and run Framework components.

Component	Minimum Permissions (UNIX)	Minimum Permissions (Windows)
Configuration Server	Users group	Administrators group [a]
Solution Control Server	Users group	Administrators group
Message Server	Users group	Administrators group
SNMP Master Agent	Users group	Administrators group
Local Control Agent [b]	`root`	Administrators group

a. The user account for the running process is usually determined by the user or object that started the process. For example, if a process is started by LCA, then the process inherits its permissions from LCA.

b. `root` or Administrators permission is required to install the component because, during installation, it updates the startup file and registry.

After a component is installed, you can update the component to start under a different user account with lower privileges. However, before doing so, make sure that you updated the working directories with the correct read and write permissions.

Example

> **Important**
>
> Support is discontinued for Genesys Deployment Agent (GDA) in LCA release 8.5.100.31 and later.

To run LCA and Genesys Deployment Agent (GDA) as a non-root user, do one of the following, depending on your operating system:

On UNIX

Create startup scripts for LCA and GDA that set up LCA and GDA to run under the non-root user. For these scripts, it is assumed that LCA is installed in **/home/genesys/GCTI**, and the name of the non-

root user is genesys. See LCA Startup Script-gctilca and GDA Startup Script-gctigda for examples of these scripts. To install the startup scripts, put them in the directory **/etc/rc.d/init.d/** and run one or both of the following commands, as required:

```
chkconfig -add gctilca
chkconfig -add gctigda
```

On Windows

Change the account associated with the LCA service. One way to do this is through Windows Administrative Services, as follows:

1. Go to **Start > Settings > Control Panel > Administrative Services > Services**, right-click **LCA**, and select **Properties**.
2. Open the **Log On** tab and in the **Log on as** section, select **This account**, and change the account associated with the LCA service.

Database User Privileges

A database user that accesses the Configuration Database on behalf of Configuration Server, that is, the user identified in the Configuration Server configuration file, requires basic database privileges, as defined in this section.

When the database is created, it is assumed that it is created under the new user and the initialization scripts are under that user account, unless otherwise stated.

Oracle

After the new database user is created, grant the necessary privileges as follows:

```
GRANT CONNECT TO <DB user>
GRANT CREATE TABLE TO <DB user>
GRANT UNLIMITED TABLESPACE TO <DB user>
GRANT CREATE PROCEDURE TO <DB user>
```

MS SQL

For MS SQL 2000, grant the `public` role to the new database user on the **Database Access** tab of the **SQL Server Login Properties** dialog box for the new user. In addition, grant the following privileges:

```
GRANT CREATE TABLE TO <DB user>
GRANT CREATE PROCEDURE TO <DB user>
```

For MS SQL 2005 and later, grant the `public` and `db_owner` roles to the new database user.

DB2

Grant the necessary privileges as follows:

```
CONNECT TO <database>;
GRANT CREATE TAB,CONNECT ON DATABASE TO USER <DB user>;
CONNECT RESET;
```

PostgreSQL

From `pgAdmin`, grant the following privileges:

- Can create database object
- Can create roles

Or, you can execute the following query:

```
CREATE ROLE <DB user> LOGIN ENCRYPTED PASSWORD <encrypted password> NOINHERIT CREATEDB
CREATEROLE VALID UNTIL 'infinity';
```

To configure client authentication, update the **pg_hba.conf** file, located in the data directory under the PostgreSQL installation folder. For example:

```
host GCTI_Test gctitest <IP address1>/32 trust
host GCTI_Test gctitest <IP address2>/32 trust
```

This enables the DB user `gctitest` to connect to the `GCTI_Test` database from the hosts `IPaddress1` and `IPaddress2`.

Sample Scripts

This section contains sample scripts required to run LCA and GDA on UNIX under a non-root user.

> **Important**
>
> Support is discontinued for Genesys Deployment Agent (GDA) in LCA release 8.5.100.31 and later.

LCA Startup Script-`gctilca`

The following is an example of a script to allow LCA to run under a non-root user.

[+] Show script

```
#!/bin/bash
#
# chkconfig: 345 80 20
# description: run lca
#
# You should put this script to /etc/rc.d/init.d and run command:
# chkconfig --add gctilca
#GCTI home dir
GCTI=/home/genesys/GCTI
DIRNAME=LCA
HOMEDIR=$GCTI/$DIRNAME
USER=genesys
SCRIPTNAME=gctilca
HOME_USER=/home/genesys
PATH=/sbin:/bin:/usr/bin:/usr/sbin
prog=lca
RETVAL=0
if [ ! -x $HOMEDIR/$prog ]; then
exit 1
fi
# Source function library.
. /etc/rc.d/init.d/functions
start () {
echo -n $"Starting $SCRIPTNAME: "
if [ -e /var/lock/subsys/$prog ]; then
echo -n $"$SCRIPTNAME is already running.";
failure $"cannot start $SCRIPTNAME: $SCRIPTNAME already running.";
echo
return 1
fi
daemon --user=$USER ". $HOME_USER/.bash_profile ; cd $HOMEDIR ;
   ./run.sh >/dev/null 2>/dev/null &"
sleep 1
CHECK=`ps -e | grep $prog | grep -v $SCRIPTNAME | awk '{print $4}'`
if [ "$CHECK" = "$prog" ]; then
RETVAL=0
else
RETVAL=1
fi
[ $RETVAL -eq "0" ] && touch /var/lock/subsys/$prog
echo
return $RETVAL
}
stop () {
echo -n $"Stopping $SCRIPTNAME: "
if [ ! -e /var/lock/subsys/$prog ]; then
echo -n $"$SCRIPTNAME is not running."
failure $"cannot stop $SCRIPTNAME: $SCRIPTNAME is not running."
echo
return 1;
fi
killproc $prog
RETVAL=$?
echo
[ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/$prog;
return $RETVAL
```

```
}
usage ()
{
echo "Usage: service $PROG {start|stop|restart}"
}
case $1 in
start)
start
;;
stop)
stop
;;
restart)
stop
start
;;
*)
usage ; RETVAL=2
;;
esac
exit $RETVAL
```

GDA Startup Script-`gctigda`

The following is an example of a script to allow GDA to run under a non-root user.

[+] Show script

```
#!/bin/bash
#
# chkconfig: 345 80 20
# description: run gda
#
# You should put this script in /etc/rc.d/init.d and run command:
# chkconfig --add gctigda
#GCTI home dir
GCTI=/home/genesys/GCTI
DIRNAME=LCA
HOMEDIR=$GCTI/$DIRNAME
USER=genesys
SCRIPTNAME=gctigda
HOME_USER=/home/genesys
PATH=/sbin:/bin:/usr/bin:/usr/sbin
prog=gda
RETVAL=0
if [ ! -x $HOMEDIR/$prog ]; then
exit 1
fi
# Source function library.
. /etc/rc.d/init.d/functions
start () {
echo -n $"Starting $SCRIPTNAME: "
if [ -e /var/lock/subsys/$prog ]; then
echo -n $"$SCRIPTNAME is already running.";
failure $"cannot start $SCRIPTNAME: $SCRIPTNAME already running.";
echo
return 1
fi
daemon --user=$USER ". $HOME_USER/.bash_profile ; cd $HOMEDIR ;
   ./gda >/dev/null 2>/dev/null &"
sleep 1
```

```
CHECK=`ps -e | grep $prog | grep -v $SCRIPTNAME | awk '{print $4}'`
if [ "$CHECK" = "$prog" ]; then
RETVAL=0
else
RETVAL=1
fi
[ $RETVAL -eq "0" ] && touch /var/lock/subsys/$prog
echo
return $RETVAL
}
stop () {
echo -n $"Stopping $SCRIPTNAME: "
if [ ! -e /var/lock/subsys/$prog ]; then
echo -n $"$SCRIPTNAME is not running."
failure $"cannot stop $SCRIPTNAME: $SCRIPTNAME is not running."
echo
return 1;
fi
killproc $prog
RETVAL=$?
echo
[ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/$prog;
return $RETVAL
}
usage ()
{
echo "Usage: service $PROG {start|stop|restart}"
}
case $1 in
start)
start
;;
stop)
stop
;;
restart)
stop
start
;;
*)
usage ; RETVAL=2
;;
esac
exit $RETVAL
```