Logging Collector 3.0
Installation Guide

Version 1.1
10/12/2005

Version: 1.1
Date: 10/12/2005
Authors: S. Traldi
CI identifier

Revision History

Date	Version	Description	Author
March 22, 2005	1.0	Document creation	S. Traldi
October 10, 2005	1.1	Altering post-installation: adding parameter to web.xml	S. Traldi

CI Record

Field	Description
CI Identifier	
Description	
Submission Date	
Submitted By	
Components	
Dependencies/Related	
External Identifier	
Point of Contact	
Comments	
Physical Location	

Table of Contents

This is a note, e.g.: Always put originator and targetAddr in your SOAP messages when sending SOAP messages to other XDAQ applications

This is a help item, e.g: How do I restart an executive, if I lost connection to the host

This is an orientation item,e g.: You have the choice between using a Semaphore or polling the channel. The advantage of the first is bla bla, the advantage of the second is bla bla.

This is a tip, e.g.: To query a parameter several times, use the timer in the window “Properties” of the slected application.

The Logging Collector

1 Introduction

The Logging Collector is a web application designed and developed to collect logging information from log4j compliant applications.
The logging collector allows also to store logging information in a persistent way in database and/or to distribute/publish it through a real time message system (JMS).
This guide explains how to install the Logging Collector and some additional software needed by the collector to work.

Here you can find three different installation procedures:

1. BASIC: Only the core Web Collector Application is installed under a Tomcat Servlet Container. In this case you can distribute logging information just using the output Chainsaw.
   This type of installation is done when you just test the Collector Application, and you needn't many clients who receive the logging information.
2. ADVANCED: Further to BASIC setup the following components are installed, if they are not already available:
   - a Sun Message Queue JMS (with a shared nfs file as JNDI lookup repository)
   - a MySql database

   This type of installation is done when there are many clients that want to receive the logging information or you need to the logging information in a persistent way.
3. COMPLETE: Further to BASIC setup the following components are installed, if they are not already available:
   - a Sun Message Queue JMS (with an OpenLDAP service as JNDI lookup repository)
   - a MySql or Oracle database.

   This type of installation requires a system administrator to install ORACLE.

2 Installation Overview

This chapter describes the different types of Collector Web Application installations that you can perform, and issues that you should consider before installing the software. It includes information about the following topics:

- Installation Overview
- Collector Web Application Installation Methods

2.1 Installation Overview

The Collector Application installation process consists of four parts:

1. Completing pre-installation task: Chapter 3 describes pre-installation tasks that you must complete before installing the product.
2. Installing Collector Web Application: Chapter 4 describes the commands to do for install the collector product. The script to launch, the command to execute.
3. Installing external software: Chapter 5 describes the external software installation procedures. Some of those software are necessary for the collector, some other are optional for the collector, but provide many functionalities of the entire system.
4. Completing post-installation tasks: Chapter 6 describes all the recommended and required post-installation tasks.

2.2 Logging Collector Installation Method

You can choose three different installation methods to install the Collector Web Application, as follows:

- Basic Installation method
  Choose this type of installation if you want to quickly install the Collector Application. The application will be installed without all its functionalities. You will see the collector work in a easy way.
  - See the Chapter 3 for the pre-installation requirement. If you have all the platform and programs installed,
  - pass to installing the Software: chapter 4 (Installing Collector Web Application) paragraph 4.1 (Download from CVS Repository) and paragraph 4.2 (deploy the Collector Web Application).
- Advanced Installation method
  Select this installation method if you want see the collector work using all functionalities, but with a not large distributed system.
  - Do the Basic Installation and
  - pass to installing the external Software: chapter 5 (Installing external software), paragraph 5.3 (Mysql) and paragraph 5.4 (JMS – IMQ without OpenLDAP, using only the filesystem shared).
- Complete Installation method
  Choose this installation method if you want install a complete and distribute system, to allow many users to work with the collector application.
  - Do the Basic Installation and
  - pass to Installing Software: chapter 5 (Installing external software), paragraph 5.4 (JMS – IMQ with OpenLDAP), paragraph 5.3 (Mysql). If you prefer you

can install Oracle instead of mysql, see the Oracle documentation or the Appendix in this document.

- After this you could see also the Chapter 6 (Completing post-installation tasks) to configure the Collector.

3 Completing pre-installation task

This chapter describes the task that you must complete before you start the Collector Web Application. It includes information about the following tasks:

- Log In to the System as root (only for linux users)
- Check the Hardware Requirements
- Check the Software Requirements
- Check the existence of System Variables

3.1 Log In to the System as root (only for Linux users)

Before you install the Collector software, you must complete several tasks as the `root` user. To log in as the `root` user, complete one of the following procedures.

Note: Unless you intend to complete a silent-mode installation, you must install the software from an X Window System workstation, an X terminal, or a PC or other system with X server software installed

- If you are installing the software from an X Window System workstation on X terminal:
  1) Start a local terminal session, for example an X terminal (`xterm`).
  2) If you are not logged in as the `root` user, enter the following command to switch user to `root:`

```
$  su – root
password:
```

  just type the correct password.

3.2 Check the Hardware Requirements

The system must meet the following minimum hardware requirements for basic installation:

- PII 200 MHz
- 64 MB of physical RAM
- 150 MB of disk space in the /usr to install java, tomcat and the collector application

The system must meet the following minimum hardware requirements for Easy Advanced Installation:

- PII 200 MHz
- 128 MB of physical RAM
- 300 MB of disk space in the /usr to install java, tomcat, the collector application, the mysql database and JMS
- is required a partition shared like nfs

The system must meet the following minimum hardware requirements for Advanced Installation:

- PIII 400 MHz
- 512 MB of physical RAM
- (Optional) 1 GB of swap space (or twice the size of RAM – for Oracle installation)
- (Optional) 400 MB of disk space in the /tmp directory for installation swap
- (Optional) Between 1.5 GB and 3 GB of disk space for Oracle software, depending on the installation type and platform.
- 400 MB of disk space in the /usr to install java, tomcat, the collector application, the mysql database and JMS with OpenLDAP and Berkley DB.

To ensure that the system meet these requirements, follow these steps:

1) To determinate the speed processor, enter the following commands:
   Linux : shell> more /proc/cpuinfo
   Windows: Start->Run->Msinfo32

2) To determinate the physical RAM size, enter one of the following commands:
   Linux : shell> more /proc/meminfo
   Windows: Start->Run->Msinfo32

3) To determinate the amount of disk space in the system, enter one of the following commands:
   Linux : shell> df -h
   Windows: Start->Run->cmd->chkdsk

4) To determinate the size of the configured swap space, enter the following command:
   Linux : shell> grep SwapTotal /proc/meminfo

3.3 Check the Software Requirements

To see the collector working you must have installed in you Pc the Java Platform SDK 1.4.2_0X and a Servlet Container Jakarta Tomcat 5.0.X.
To determinate if the software are installed or less, try to type from a shell the command `java –version`, if you have in output a string, see if this match with thee correct version Java SDK 1.4.2_0X, otherwise install Java. For Tomcat try to find if in your file system there is a directory like jakarta-tomcat-5.0.X: type the command as root user: `cd /` (to change directory to the root of the file system) and `find . –name jakarta-tomcat*` (to find if there is a tomcat in your machine).
If you don’t have these two components installed see next chapter Installing Software, paragraph 1 (Java Installation) and paragraph 2 (Tomcat Installation).

3.4 Check the existence of System Variables

To determinate if the System Variables are set correctly, type the command `set` from a bash

or korn shell or `printenv` from a tcsh or a c shell, and control if there are these two variables JAVA_HOME, and CATALINA_HOME, if you haven't these variables see next chapter Installing Software, paragraph 1 (Java Installation) and paragraph 2 (Tomcat Installation) .

4 Installing Collector Web Application

4.1 Download from CVS Repository

- Login to cvs use the command:
  cvs -d :pserver:cvsanonymous@cmscvs.cern.ch:/cvs_server/repositories/TriDAS login
- password: 98passwd
- download the project in some place:
  cvs -d :pserver:cvsanonymous@cmscvs.cern.ch:/cvs_server/repositories/TriDAS co -d <directory_selected> RunControl/tools/Collector
  where <directory_selected> is the place where you want to put the project Collector version 2.0. (For example)
  cvs -d :pserver: cvsanonymous@cmscvs.cern.ch:/cvs_server/repositories/TriDAS co -d collector RunControl/tools/Collector
- if you don't matter the directory structure, just do:
  cvs -d :pserver: cvsanonymous@cmscvs.cern.ch:/cvs_server/repositories/TriDAS co RunControl/tools/Collector
- a directory path RunControl/tools/Collector will be created, with the collector project inside.

If you use an IDE (Integrated Development Environment) like Eclipse, Jbuilder or Jdev, you could download the project setting the following variables:

Host:	cmscvs.cern.ch
Repository Path:	/cvs_server/repositories/TriDAS
User:	cvsanonymous
Password:	98passwd
Connection type:	pserver
Port:	defult Port (2401)
Module:	RunControl/tools/Collector

If you want to know what are inside the Collector_3_0 project just downloaded, see the chapter 7: Directory Structure

If you want just to see the project but not download it:

http://cmsdoc.cern.ch/swdev/viewcvs/viewcvs.cgi/TriDAS/RunControl/?cvsroot=TriDAS

4.2 Deploy Application Collector into Tomcat and start the Application

- Enter in directory you are created before
  cd /<directory_selected>
- Change directory into /release/war
  cd release/war

- Copy the file inside the directory into $CATALINA_HOME/webapps.
- Now you can just launch:
  cp Collector.war $CATALINA_HOME/webapps
- Change directory entering into $CATALINA_HOME/webapps
  cd $CATALINA_HOME/webapps
- Start up the Tomcat with the command:
  ../bin/startup.sh
- Open a browser and type the url:
  http://<PC where Tomcat is running>:<port>/Collector
  usually the port is 8080 an example could be:
  http://lxcmddemo.cern.ch:8080/Collector

Now see the example in the User Guide Manual at chapter 7 paragraph 7.1 Logging Collector – Chainsaw to test the Collector, or if you are installed JMS or the DB see if the logging information are arrived to the client or inside the tables in the database.

5 Installing external Software

Logging Collector uses the external platform and program to work:

- Sun Java 1.4.2_0X
- Apache Jakarta Tomcat 5.0.28
- A Database (Oracle or Mysql)
- JMS (Sum Message Queue)

5.1 Java Installation

The logging collector requires a Java Platform Installed in the PC.

- Download the Java Standard Edition sdk from java.sun.com : http://java.sun.com/j2se/1.4.2/download.html chose Download J2SE SDK, accept the license agreement, chose a platform.
- Linux: j2sdk-1_4_2_07-linux-i586.bin
  - Copy the file in some place, usually in /usr/local or /usr/java or /usr/local/java.
  - Give the execution privileges to that file j2sdk-1_4_2_07-linux-i586.bin chmod 755 j2sdk-1_4_2_07-linux-i586.bin
  - Execute the file. This will create a directory, usually like j2sdk1.4.2_07.
  - Set the java home with the proper path: (for example)
    export JAVA_HOME=/usr/java/j2sdk1.4.2_07 for bash or korn shell (bash, ksh)
    setenv JAVA_HOME /usr/java/j2sdk1.4.2_07 for c shell (tcsh, csh)
  - Add to the PATH variable the java bin command:
    export PATH= /usr/java/j2sdk1.4.2_07/bin:$PATH for bash or korn shell
    setenv PATH /usr/java/j2sdk1.4.2_07/bin:$PATH for c shell (tcsh, csh)
- Windows: j2sdk-1_4_2_07-windows-i586-p.exe
  - Execute the file, and chose the proper directory to install the Java SDK.
  - Set the System Variables in the Control Panel. JAVA_HOME and PATH.

If you have problems to install java, see the documentation in
http://java.sun.com/j2se/1.4.2/docs/index.html

5.2 Tomcat Installation

The Jakarta Tomcat is required for the Collector Application. As the Collector is a Web Application needs the Tomcat to run.

- Download the Tomcat version 5.0.28 from the jakarta site. http://jakarta.apache.org/site/downloads chose Tomcat, Tomcat 5, 5.0.28 Binary for your operating system.
- Linux: jakarta-tomcat-5.0.28.tar.gz
  - Gunzip the file in some place, usually in /usr/local using
    tar -xzvf jakarta-tomcat-5.0.28.tar.gz
  - This create a directory containing the Application Tomcat, usually jakarta-tomcat-5.0.28

- Set the system variable: (for example)
  export CATALINA_HOME=/usr/local/jakarta-tomcat-5.0.28
  for bash or korn shell (bash, ksh)
  setenv CATALINA_HOME /usr/local/jakarta-tomcat-5.0.28
  for c shell (tcsh, csh)

- Windows: jakarta-tomcat-5.0.28.zip
  - Gunzip the file in some place.
  - This create a directory containing the Application Tomcat, usually jakarta-tomcat-5.0.28
  - Set the System Variable in the Control Panel. (CATALINA_HOME)

- To start and stop tomcat use the startup.sh and shutdown.sh for linux, and startup.bat and shutdown.bat for windows in $CATALINA_HOME/bin.

5.3 Mysql Installation

- Download the mysql database server 4.1.10 to http://dev.mysql.com/downloads/index.html chose MySQL 4.1, select the version you need. For example mysql-standard-4.1.10a-pc-linux-gnu-i686.tar.gz
- copy the file mysql-standard-4.1.10a-pc-linux-gnu-i686.tar.gz in some place. For example in usr/local/
- execute this group of commands:

```
shell> groupadd mysql
shell> useradd -g mysql mysql
shell> cd /usr/local
shell> gunzip < /PATH/TO/MYSQL-VERSION-OS.tar.gz | tar xvf -
shell> ln -s FULL-PATH-TO-MYSQL-VERSION-OS mysql
shell> cd mysql
shell> scripts/mysql_install_db --user=mysql
shell> chown -R root .
shell> chown -R mysql data
shell> chgrp -R mysql .
shell> bin/mysqld_safe --user=mysql &
```

For any problems you find refer to the documentation in
http://dev.mysql.com/doc/mysql/en/index.html
See also the README and INSTALL-BINARY file in the directory FULL-PATH-TO-MYSQL-VERSION-OS (for example: mysql-standard-4.1.10a-pc-linux-gnu-i686).

5.4 JMS Installation

The JMS is like Oracle and Mysql, is not necessary to the collector working. But if you would like some subscribers retrieve the log, you need to install JMS.
There are some way to install JMS. You could install JMS and use a nfs file system to map the Topic and the TopicFactory. Or in other way (better way), you could install an OnpeLDAP service, where the Topic and the TopicFactory are installed.

5.4.1 *IMQ Installation*

- download mq3_6-plt-linx86.zip from the site http://www.sun.com/software/products/message_queue/index.xml
- chose Download last release, Sun Java System Message Queue 3.5 SP2 Platform Edition, Download, Continue and accept the License Agreement.
- Download the platform file you want. For example Red Hat Linux imq3_5-plt-linx86.zip
- unzip the file imq3_5-plt-linx86.zip:
  unzip imq3_5-plt-linx86.zip
- cd mq3_5-plt/rpms
- install the rpm file. (use --test to know if is possible to install the rpm or there are some dependences)
  rpm -ivh --test imq-3_5-03.i386.rpm
- if the result test is ok. Install the rpm. If there're some dependences. Install the other proper rpm files.
  rpm -ivh imq-3_5-03.i386.rpm

To create the connection Topic Factory and the Topics in JMS you could use the file in the directory tool/jms/configurationFiles
add_remote_fs_tcf.properties
add_remote_fs_t.properties
OR
add_ldap_fs_tcf.properties (only if you have installed an OpenLDAP system)
add_ldap_fs_t.properties (only if you have installed an OpenLDAP system)

Note: Remember to change the parameters of the provider URL, ConnectionFactory name and Topic name.
The provider URL is a pointer in the file system nfs or in an OpenLDAP Service. In the first case you just put a directory name created in a nfs filesystem (visible from the tomcat, and all the client which use JMS and the collector). In the second case you must set an URL where the OpenLDAP Server is installed.

The commands to use the above file are:
/opt/imq/bin/imqobjmgr -javahome $JAVA_HOME -i add_remote_tcf.properties
/opt/imq/bin/imqobjmgr -javahome $JAVA_HOME -i add_remote_t.properties

On the other way you could create manually the Connection Topic Factory, and the Topics with the following commands.

Create the Connection Topic Factory:
This imqobjmgr command can be used to add a QueueConnectionFactory object where JNDI lookup name = MyQueueConnectionFactory, the Provider URL = C:/Temp on windows or /tmp on Unix.
This is the command used to create the TopicFactory:
imqobjmgr add
-t qf
-l "MyQueueConnectionFactory"

-j java.naming.provider.url=file:///tmp
-j java.naming.factory.initial=com.sun.jndi.fscontext.RefFSContextFactory

Create the Topic:
This imqobjmgr command can be used to add a Queue object where JNDI lookup name = MyQueue, Destination Name = MyQueueDest, Provider URL = C:/Temp on windows or /tmp on Unix
imqobjmgr add
-t q
-l "MyQueue"
-o "imqDestinationName=MyQueueDest"
-j java.naming.provider.url=file:///tmp
-j java.naming.factory.initial=com.sun.jndi.fscontext.RefFSContextFactory

To startup JMS use the command: /opt/imq/bin/imqbrokerd -javahome $JAVA_HOME

Other Information
For a better understanding see the specific documentation of Sun Message Queue in the files: MessageQueue_InstallGuide.pdf and MessageQueue_AdministrationGuide.pdf
You can find these files in http://www.sun.com/software/products/message_queue/index.xml

5.4.2 OpenLDAP Installation

- Download automatic from shell the file openldap-2.2.24.tgz
  wget ftp://ftp.openldap.org/pub/OpenLDAP/openldap-release/openldap-2.2.24.tgz
- or if you prefer download manually the file openldap-2.2.24.tgz from
  http://www.openldap.org/software/download/
- copy the file in /usr/local or in same place that you would like install OpenLDAP.
  cp openldap-2.2.24.tgz /usr/local
- unzip the file openldap-2.2.24.tgz
  tar -zxvf openldap-2.2.24.tgz
- change directory, and enter in the just unziped directory for example
  /usr/local/openldap-2.2.24
  cd /usr/local/openldap-2.2.24
- execute the configuration file and make the bin of OpenLDAP
  shell> ./configure
  shell> make depend
  shell> make
  shell> make test
  shell> su root -c 'make install'
- if the result of every commands is positive, you can continue the installation. If after the command ./configure commands have an output like this or similar:
  checking Berkeley DB version for BDB backend... no
  configure: error: BDB/HDB: BerkeleyDB version incompatible
- you need version of BerkleyDB 4.2.*, so you must install a BerkleyDB. See the following chapter for the installation of a BerkleyDB.
- If all are ok, modify the file slapd.conf in /usr/local/etc/openldap, using for example the file in directory tool/jms/configurationFiles Remember to put the correct field in

dc.

5.4.3 BerkleyDB Installation

- download automatic from shell the file db-4.2.52.NC.tar.gz
  wget http://downloads.sleepycat.com/db-4.2.52.NC.tar.gz
- or if you prefer download manually the file db-4.2.52.NC.tar.gz (if the automatic download didn't run) from http://www.sleepycat.com/download/index.shtml
- copy the file in some place in your file system for example in /usr/local
  cp db-4.2.52.NC.tar.gz /usr/local
- unzip the file db-4.2.52.NC.tar.gz
  tar -zxvf db-4.2.52.NC.tar.gz
- change directory entering in the directory just create unzipping the file. For example /usr/local/db-4.2.52.NC
  cd /usr/local/db-4.2.52.NC
- enter in the subdirectory build_unix
  cd build_unix
- execute the following commands. That are the configuration files and the bin to make.
  shell> ../dist/confiure
  shell> make
  shell> make install
  shell> make clean
  shell> make
- set some system variables. if use a bash/korn:

export LD_LIBRARY_PATH=/usr/local/BerkeleyDB.4.2/lib:$LD_LIBRARY_PATH
export CPPFLAGS="-I/usr/local/BerkeleyDB.4.2/include":$CPPFLAGS
export LDFLAGS="-L/usr/local/BerkeleyDB.4.2/lib":$LDFLAGS

- if use a C shell:

setenv LD_LIBRARY_PATH /usr/local/BerkeleyDB.4.2/lib
setenv CPPFLAGS "-I/usr/local/BerkeleyDB.4.2/include"
setenv LDFLAGS "-L/usr/local/BerkeleyDB.4.2/lib"

5.4.4 Error and References

If some errors occurred see the documentation in

- http://www.sleepycat.com/ for BerkeleyDB
- http://www.openldap.org for OpenLDAP
- http://www.sun.com/software/products/message_queue/index.xml for Sun Message Queue (IMQ)

In the directory /tool/jms/sun/installScrpits you could find a file named install_jms.sh that is a scripts linux which install the entire system JMS (BerkleyDB, OpenLDAP, IMQ (Sun Message Queue). See that file also.

6 Completing post-installation tasks

6.1 Configure the Collector Application properties

Some indications how to configure the collector, before start it.
In directory /collector/WEB-INF there is a file web.xml
Open that file and set some parameters:
where you find the following lines :

```
<init-param>
    <param-name>CollectorInternalLogLevel</param-name>
    <param-value>DEBUG</param-value>
</init-param>
```

set the param-value to the Level you want the Collector Log write Log, if you set DEBUG, all internal Logging event are stored in the file CollectorLog.<date>.txt inside the /collector/Log directory. If you set WARN or ERROR, only the warnings or error in output to the collector code are stored in that file.

where you find the following lines :

```
<init-param>
    <param-name>StateStartMode</param-name>
    <param-value>CONFIGURE</param-value>
</init-param>
```

set this parameter to deterinate how you want the Log Collector start. If you set to START when the tomcat will start also the Log Collector are in start mode, keeping the configuration of the last time started.

where you find the following lines :

```
<init-param>
    <param-name>PingCommandEnabled</param-name>
    <param-value>ON</param-value>
</init-param>
```

before adding a Socket Appender the Log Collector makes a ping command to see if the host where the log messages will send is reachable. Set to OFF if you can't use the ping command in the host where the Log Collector is installed or in the case the receiving host can't accept ping command.

where you find the following lines :

```
<init-param>
    <param-name>CollectorLogFileName</param-name>
    <param-value>CollectorLog.%d{yyyy-MM-dd}.txt.zip</param-value>
</init-param>
```

set properly the param-value. CollectorLog is standard, %d means that you want a daily Log file, every day the application archives the old collector Log file in a zip file and create a new file named CollectorLog.<date>.txt where date is today date in format {yyyy-MM-dd}. If you put %w instead of %d you have a weekly collector Log file.

These two parameters are useful to make the collector quickly, just think write all log is convenient to debug the collector but on the other side decrease the collector performance. So write a log daily it could be helpful to debug the application, but if I would like to see the yesterday log file I must open a zip file, and see.

6.2 Configure the collector Application

Remember to copy the the database library in /WEB-INF/lib if you want to use the appender to Database. After the copy remember to restart tomcat, try to shutdown tomcat and startup tomcat (see the previous command in chapter 2 to do it)
See the other documentation manual the User Guide to well configure the Log Collector using the Configuration web page.

7 Directory and files

7.1 Directory Structure

The project collector, one time it's downloaded from CVS Repository, have inside this structure directory:

- bin
- collector
  -> images jsp WEB-INF FileConfiguration Log
- collectorClient
  -> lib outputFiles scripts
- collectorClientWebServiceCommand
  -> lib scripts
- j2src
  -> the collector source code
- release
  -> client doc war
- tools
  -> db jms

In the *bin* directory will be inserted the compiled collector source programs.

In the *collector* directory there are some directories and some file. The must important directories are *WEB-INF, FileConfiguration, Log*.
The *WEB-INF* directory contains inside the scripts ant to build the collector application, the collector clients, the webservices clients and a directory *lib* where there're the libraries used to the collector to work.
The *FileConfiguration* directory contains a XML file that is the real configuration of the collector. See the related documentation to configure properly the collector.
The *Log* directory contains the collector (internal) log. The last file created is a txt simple file. The other file are in zip format (older files).

In the *collectorClient* directory there are 3 directories *lib, outputFiles, scripts*. All the compiled code (the real clients code) are in a jar file inside the directory *lib*.
The *lib* directory also contains all the libraries for using JMS and Log4J.
The *scripts* directory contains the scripts to launch the JMS or socket clients and a *configFiles* directory. Remember to give the proper linux privileges to permit the script work. Then configure the client using the file inside the *configFiles* directory. For example edit the ClientJMSCollector.config and set the initial_context_factory, the provider_url, the topicConnectionFactory and the topicName with the correct parameters.
In the *outputFiles* directory will be created 3 files in 3 different format: one txt, one html and one xml containing the log arrived from collector. (result of the subscription at JMS or the reading socket).

In the *collectorClientWebServiceCommand* directory there are 2 directories *lib* and *scripts*.

All the compiled code (the real clients WebServices code) are in a jar file inside the directory *lib*.
The *lib* directory also contains all the libraries for using WebServices invocations.
The *scripts* directory contains the scripts to launch the clients which send command to collector. There is a *configFiles* directory, where you can find a file: CollectorWebServiceConfiguration.config , edit it and put the correct CollectorServiceAddress.
(For example http://asspes.lnl.infn.it:8080/Collector_3_0/Collector)

In the *j2src* directory there are the collector source code.

In the *release* directory there are 4 directories: *client, chainsaw, doc* and *war*. Inside this directory there is a zip file containing the all packages of the collector.
In *client* directory there are 2 zip files, one contains the collectorClient previously described, the other zip file contains the collectorClientWebServiceCommand before shown.
In *chainsaw* directory, there is a file chainsaw.zip, you have to unzip the file and use chainsaw to receive logging information form JMS. To know how to configure the script see the User Guide Manual.
In *doc* directory you can find this documentation, the user manual and all the javadoc of the project.
In *war* directory, you can find the war file, just ready to put in tomcat webapps directory.

In *tools* directory there are 2 directories *db* and *jms*.

In the *db* directory there are 2 directories: *mysql* and *oracle*. Inside at these 2 directories there are other 2 directories for each of those: *script* and *lib*.
Inside *lib* you can find the libraries of oracle or mysql to copy in WEB-INF/lib to permit the collector to work. Or you could use it to create new client for oracle or mysql.
Inside *script* you can find the script to create the table in the database, the indexes, the trigger, ...

In *jms* directory there is 2 directories *sun* and *configurationFiles.*
Into the *sun* directory thre are two more directories *lib* and *install_script*.
The *lib* directory is useful to contain all libraries needed to collector to work with sun JMS.
In *install_script* you can find a linux script which install all the JMS structure, The BerkeleyDB 4.2, the OpenLDAP and the IMQ Sun Message Queue.
In *configurationFiles* directory you could find some file used in the JMS or openLDAP installations.

7.2 Problem with Scripts

For linux script: if someone doesn't work. Control the file privileges, if you have the execution privileges ok otherwise use chmod:
chmod 555 <namescrtipt.sh>

7.3 Script Ant

- Enter in directory /collector/WEB-INF and type the command:

ant -projecthelp

in output you should have all the compile possibilities:

buildJavaDoc	Make the java docs.
compile	Compiles the source code.
deploy	Deploy War in the WebServer.
releaseAll	Build All into zip file.
ReleaseCollectorClient	Build Release Collector Client into zip file.
releaseCollectorClientWebServiceCommand	Build Release Collector Client Web Service Command into zip file.
releaseWar	Build Release War.

7.3.1 Using Script Ant

Use the Ant scripts is easy, just chose one of the possible commands to build the project, to make the java docs, to create all zip files.
Each commands is important, but just to view the collector work, configure the file build.properities in /colelctor/WEB-INF. Open that file with a simple editor, find the deploy section. Set the directory path to $CATALINA_HOME/webapps and use the releaseWar, and deploy c ommands:
ant –releaseWar
ant –deploy
These two commands compile the project Colelctor and create a file .war and copy the Collector.war file in $CATALINA_HOME/webapps. Stop the Tomcat if it is running, and start the Tomcat. And the Web Application is ready to run.

Appendix A: Oracle Installation

This Component is not necessary to the collector working. But if you want to archive the logs in a Database this one of the two possibility.
To install Oracle Database is not so easy. Here there is a simple way to install it, but I think is better that the oracle installation is done by person who know how to do it. Anyway you must register to ww.otn.oracle.com, you must set some system variables, change some kernel variables and you must have some prerequirements system hardware and software, you can find it in Oracle Manual in
http://www.oracle.com/technology/documentation/database10g.html

Quick installation only to Linux RedHat AS or ES (also RedHat 9.0):

Adding Users and Groups

First, you will need to create the Oracle installation and users and groups. Oracle installation needs two Unix user groups and one runtime Oracle user.

Log in as root and issue the following commands in a terminal:

```
[root@dbasspes]# groupadd dba
[root@dbasspes]# groupadd oinstall
[root@dbasspes]# useradd -g oinstall -G dba oracle
[root@dbasspes]# passwd oracle
```

The last command will prompt you to enter the password for your `oracle` user. Make sure you remember it, because you will probably need it as we go along.

Creating Directories

Oracle recommends OFA, the Optimal Flexible Architecture directory structure for a deployment server, but on a development machine and for the sake of simplicity, we will install everything under */opt/ora9*.
Just make sure you have at least 3.5GB available for a full installation including one database, and issue the following commands as root:

```
[root@dbasspes]# mkdir -p /opt/ora10/product/10.1.0.3
[root@dbasspes]# mkdir /var/opt/oracle
[root@dbasspes]# chown oracle.dba /var/opt/oracle
[root@dbasspes]# chown -R oracle.dba /opt/ora10
[root@dbasspes]# chmod 755 /var/opt/oracle
```

You have now created Oracle runtime directories and granted write privileges to user `oracle` and execute privileges to group `dba`.

Installing Required Tools and Libraries

You will need to install the following Red Hat backward-compatibility and software-development packages before we get further underway. All of these packages can be found on Red Hat installation CDs 1-3.

```
gcc-3.2.2-5
cpp-3.2.2-5
glibc-devel-2.3.2-11.9
binutils-2.13.90.0.18-9
compat-gcc-7.3-2.96.118.i386.rpm
compat-libgcj-7.3-2.96.118.i386.rpm
compat-libgcj-devel-7.3-2.96.118.i386.rpm
```

```
nss_db-compat-2.2-20.i386.rpm
```

You can install these packages using Redhat's graphical package manager available in Start menu->System Settings->Add/Remove Applications, or from the command line, using:

```
rpm -Uvh <package_name>
```

Setting Kernel Parameters

Red Hat religiously sets some kernel parameters too conservatively. Check your hardware configuration and assign enough shared memory, open files, and ports, or you may run into trouble installing and running Oracle. Append these lines to */etc/sysctl.conf* to set kernel parameters:

```
kernel.shmmax = 536870912
kernel.shmmni = 4096
kernel.shmall = 2097152
kernel.sem    = 250 32000 100 128
fs.file-max   = 65536

net.ipv4.ip_local_port_range = 1024 65000
```

Append these lines to */etc/security/limits.conf* to modify your resource limits:

```
oracle soft nofile 65536
oracle hard nofile 65536
oracle soft nproc 16384
oracle hard nproc 16384
```

Reboot the system so the kernel changes can take effect. If rebooting is not an option, you can change the kernel params at runtime by issuing:

```
[root@dbasspes]# echo 250 32000 100 128 > /proc/sys/kernel/sem
[root@dbasspes]# echo 536870912 > /proc/sys/kernel/shmmax
[root@dbasspes]# echo 4096 > /proc/sys/kernel/shmmni
[root@dbasspes]# echo 2097152 > /proc/sys/kernel/shmall
[root@dbasspes]# echo 65536 > /proc/sys/fs/file-max
[root@dbasspes]# echo 1024 65000 > /proc/sys/net/ipv4/ip_local_port_range
```

For a full explanation of the `/proc` filesystem and available parameters, you might want to read Red Hat's Online Linux Manual.

Setting up the `oracle` User Environment

Log in as the `oracle` user:

```
% su - oracle
```

I will assume that you are using the default `bash` shell for this user. Setting environment variables in other shells may differ from this example, so check your shell's manual page or set `bash` as the `oracle` user's shell. We will set up Oracle basic environment (users, paths, locale) and some extra values needed for Oracle to run correctly on Red Hat. Put the following lines at the end of *~/.bashrc*:

```
# oracle 10g
export ORACLE_BASE=/opt/ora10
export ORACLE_HOME=/opt/ora10/product/10.1.0.3
export PATH=$ORACLE_HOME/bin:$ORACLE_HOME/Apache/Apache/bin:$PATH
export ORACLE_OWNER=oracle
export ORACLE_SID=ora10g
export ORACLE_TERM=xterm

# Use old Linuxthreads with floating stacks instead of
# the new Native POSIX Thread Library (NPTL)
export LD_ASSUME_KERNEL=2.4.1
export THREADS_FLAG=native
```

```
# Edit paths
export LD_LIBRARY_PATH=/opt/ora10/product/10.1.0.3/lib:$LD_LIBRARY_PATH
export PATH=/opt/ora10/product/10.1.0.3/bin:$PATH

#
# change this NLS settings to suit your country:
# example:
# german_germany.we8iso8859p15, american_america.we8iso8859p2 etc.
#
export NLS_LANG='AMERICAN_AMERICA.US7ASCII'
```

If you are using other national settings for Oracle (these are American), consult the supported settings and change the `NLS_LANG` variable accordingly.
The Red Hat 10 Linux kernel comes with the new Native POSIX Thread Library, which causes Oracle installation to hang. By setting the `LD_ASSUME_KERNEL` variable to an older kernel version, we are making Linux use the old Linuxthreads library. For more information about the difference between these threading methods, please consult the Red Hat site.

Running installer

To install 10g on Linux, Oracle recommends at least 512MB of RAM and at least 600MB of swap space. If you have less then 512MB of RAM and upgrading is not an option, you can resize your swap partition or create temporary swapping space. The later is a much more convenient option, as you will be needing this space only during the installation.
To set up a temporary Linux swap area, execute these lines as root:

```
% dd if=/dev/zero of=tmp_swap bs=1k count=900000
% chmod 600 tmp_swap
% mkswap tmp_swap
% swapon tmp_swap
```

After you finish installing, you can free this space:

```
% swapoff tmp_swap
% rm tmp_swap
```

Now that you have all the major obstacles out of the way, you can run the installer. Please remember that the Oracle installer **must** be run from X. You will need to allow the local `oracle` user to write to your X display:

```
$ xhost +127.0.0.1
```

ok now you are ready to install Oracle 10g.

Download Oracle and start intsallation

- Download the file to http://www.oracle.com/technology/software/products/database/oracle10g/index.html
- Tag all check boxes and accept the License Agreement
- Download the file: ship.db.lnx32.cpio.gz
- gunzip the file:
  gunzip ship.db.lnx32.cpio.gz
- Extract the cpio archives with the command "cpio -idmv < <filename>"
  cpio -idmv < ship.db.lnx32.cpio
- now you could go to install/linux directory and execute runInstaller.sh

chose your preferences: the directory where Oracle will be installed, the position of the database data, the database name, the database system and dba password, ...

PostIstallation:

If the database do not work or you must reboot the PC, you must startup the listen and the database system.

To startup the database system, enter in the machine with oracle user.

- change directory $ORACLE_HOME/bin,
  cd $ORACLE_HOME/bin
- execute the start listener command
  lsnrctl start the listener will startup.
- Enter in sqlplus with
  sqlplus /nolog
- and connect to the / of the database,
  connect / as sysdba
- execute the statup command
  startup
- Now the database is up. Execute exit or quit.
  exit

Oracle ready to use.