TECHNICAL NOTES

ECaccess Administrator's Guide

(Version 3.1.0)

Laurent Gougeon

October 2006

European Centre for Medium-Range Weather Forecasts
Europäisches Zentrum für mittelfristige Wettervorhersage
Centre européen pour les prévisions météorologiques à moyen terme

European Centre for Medium-Range Weather Forecasts
Shinfield Park,
Reading, RG2 9AX, United Kingdom

TABLE OF CONTENTS

1 INTRODUCTION

This guide is intended for the person who is to perform the administrative task of installing and/or maintaining the ECaccess software.

ECaccess is a framework for batch and interactive access to ECMWF computing and archiving facilities for Member State and other ECMWF users. The ECaccess software includes the ECaccess gateway (for UNIX and Windows platforms) and the ECaccess tools (for UNIX platforms). ECaccess tools are a set of command to run from within scripts.

Access is currently available via the Internet, the RMDCN and Leased Lines.

This guide contains five chapters (including this introduction) describing concepts and procedures for installing and/or maintaining both the ECaccess gateway and the ECaccess tools.

Throughout the guide, the terms "gateway" and "ECaccess gateway" and "tools" and "ECaccess tools" are used interchangeably.

1.1 Getting started

Refer to the following chapters to help you start using the gateway:

- Chapter 2: describes the ECaccess global architecture, focusing on the gateway (the batch authentication mechanism, the file transfer mechanism and so on).
- Chapter 3: describes how to get, install and set-up either the gateway or the tools distribution.

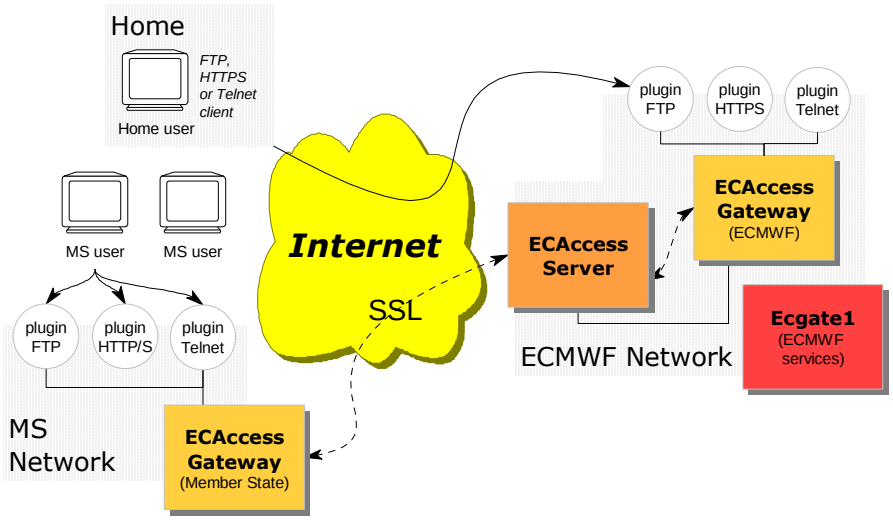
1.2 Running the gateway

Refer to the following chapters for how to run the gateway in a production environment:

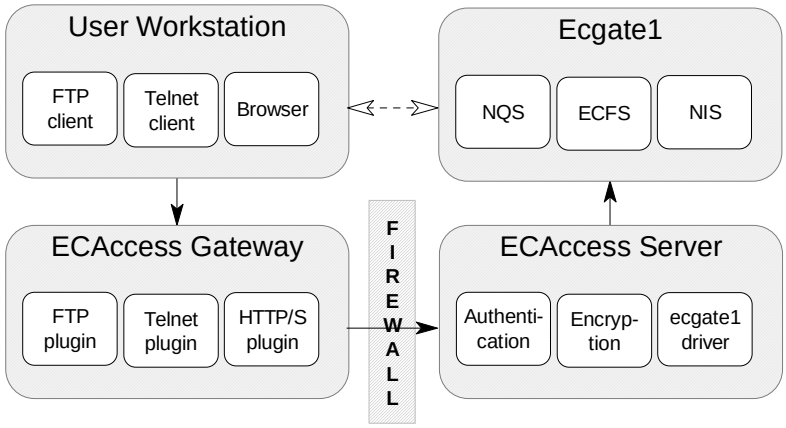
- Chapter 4: describes how to start and test the gateway.
- Chapter 5: describes the security network requirements.

2 ECACCESS ARCHITECTURE

The ECaccess software allows Member State[1] and other ECMWF users batch and interactive access to the ECMWF computing and archiving facilities. The following diagram corresponds to the global architecture of ECaccess:



Member State users can access Ecgate facilities through ECaccess components:



The components are:

- The ECaccess clients: Member State users can access the gateway interactively using an FTP client, a telnet client or a browser and using the set of ECaccess tools.

- The ECaccess gateways: all Member State users can access ECMWF computing and archiving facilities through a gateway. Full ECaccess functionality requires a gateway to be installed at the Member State. If this is not (yet) the case, limited functionality is available on the ECMWF gateway.

[1] In the following, Member State includes Cooperating States.

- The ECaccess Server: all gateways are connected to this server. It provides technical and high-level services to the gateway, allowing generic access to computing and archiving facilities at ECMWF (through “Ecgate”).

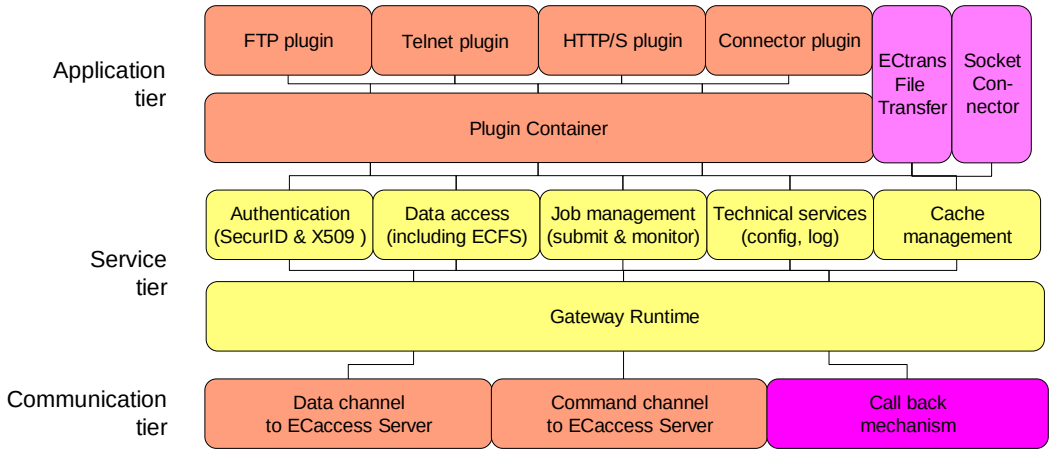
- The Ecgate server: includes services such as LoadLeveler, ECFS, HOME and SCRATCH.

The ECaccess software includes the gateway and a set of commands to access the gateway from within scripts: the ECaccess tools.

The following sections describe the internal architecture of the gateway, the “eccert” mechanism used by the tools to authenticate Member State users, the “ectrans” mechanism used to transfer files from ECMWF to Member State servers and workstations and, finally, the ECaccess Server internal architecture.

2.1 ECaccess gateway

The gateway has been designed using a multi-tier approach. The following picture represents the internal organisation of the gateway, the three tiers and the links between them:



The tiers of the gateway are:

- The applications tier: plugins act on requests/responses passing through the gateway (activated and managed by the plugin container) and the “ectrans” file transfer module waits for incoming transfer requests from the “ectrans” command (started at ECMWF).

- The services tier: the gateway provides a wide range of services (API) to the application tier. They are implemented and executed by the gateway runtime.

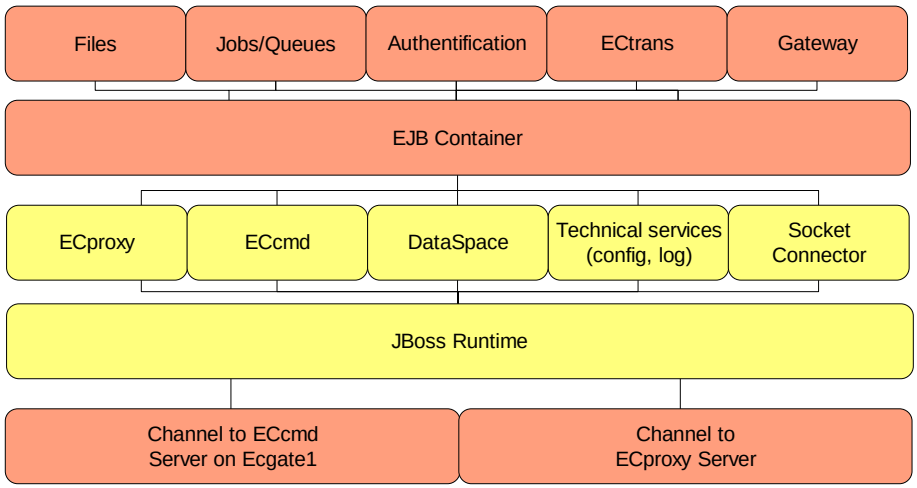
- The communication tier: the gateway runtime communicates with the ECaccess Server using a secure channel for commands and a secure channel for data transfers. A call-back module also waits for notifications from the ECaccess Server (such as “ectrans” notifications).

The gateway runtime module is the “kernel” of the gateway, and is responsible for the applications management: for example, if the communication tier fails to connect to the ECaccess Server or loses a connection, the affected application instance will run in a degraded mode, waiting for the connection to be re-established. Clients trying to connect to applications (such as the FTP plugin) will then be rejected or delayed, depending on whether the connection can be re-established quickly or not.

The gateway runtime module also handles call back management: if the ECaccess Server needs to return data or activate a specific application (such as the “ectrans” module), the gateway runtime module will notify the application and link it to the ECaccess Server.

2.2 ECaccess Server

Like the gateway, the ECaccess Server has been designed using a multi-tier approach. The following picture represents the internal organisation of the ECaccess Server, the three tiers and the links between them:



The ECaccess Server includes built-in mechanisms (based on Java and SSL) to authenticate gateways and to provide a secure framework for remote access to Ecgate services from any plugins (from any gateway).

The tiers of the ECaccess Server are:

- The applications tier: the EJB (Enterprise Java Bean) container manages stateful session beans, which are gateways dedicated modules: the Gateway module is used to manage and to monitor all the connected gateways. The Authentication module allows Member State users to authenticate through the gateway. The Jobs/Queues module provides functionalities to submit and monitor jobs submissions, and the File module provides functionalities to allow data transfers between the gateways and ECMWF.

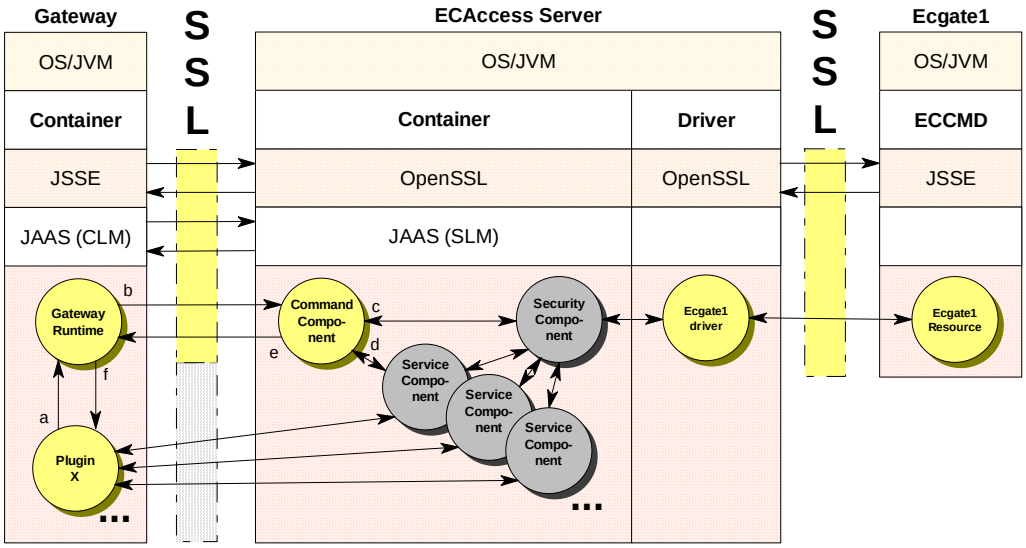
- The services tier: the ECaccess Server provides a wide range of services (API) to the application tier. They are implemented as stateless session beans and executed by the kernel.

- The communication tier: the ECaccess Server communicates with the ECCMD Server and organizes the communications between the gateways and the ECproxy server (for data transfers).

The ECaccess Server is an Application Server (AS).

2.3 Service request

ECaccess is implemented using a client-server model. The gateway runs on the Member State side, while the ECaccess Server runs on the ECMWF side:



The 6 steps for a plugin (within a gateway) to activate a service on the ECaccess Server side (and its corresponding Ecgate service) are:

a) The plugin (e.g. plugin X) accepts a connection from a Member State user (either from a FTP, HTTP/S or Telnet client): a request is sent to the gateway runtime.

b) The request is forwarded to the ECaccess Server command component to authenticate the Member State user and to activate the requested service on “Ecgate”.

c) The ECaccess Server command component checks the Member State user identity calling the Security Component: the Ecgate driver is used to access the ECaccess Certificate Authority. If the Member State user is authorized to access the service requested, a token is created and returned to the Command component.

d) The token is used by the Command component to activate the requested Service component: the Service component obtains the Member State user identity (and associated parameters) directly through the Security Component using the token.

e) The Command component returns the token (a reference of the running Service Component on the ECaccess Server) to the gateway runtime: a copy of the token is kept by the gateway to monitor and manage (in case of network issues for example) all open sessions between its plugins and the ECaccess Server.

f) The gateway runtime informs the plugin of the success of its request, providing the token. The plugin uses the token to connect to the target Service Component and to deal with its user request.

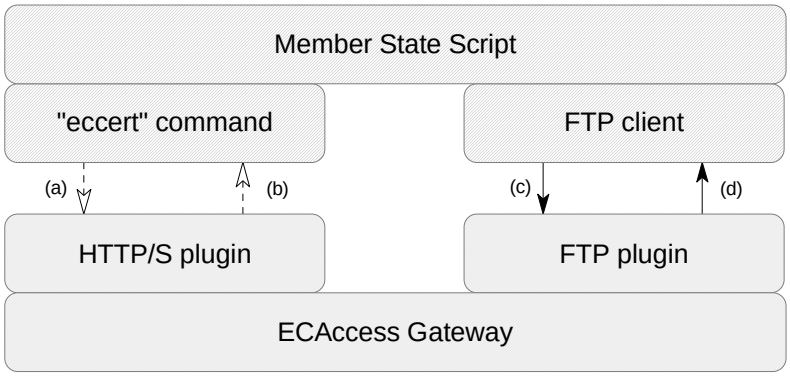
2.4 ECcert authentication

The “eccert” command can be used either to generate a temporary password (to connect to the FTP plugin), or to create a secure connection to Ecgate (bound to a temporary socket server running on a random port of the gateway) using an ECaccess certificate.

Temporary password

The “eccert” command is used to login from a standard FTP client, using both the ECMWF user identifier and the temporary password (which can be used only once and for a short period of time).

Batch authentication uses the following mechanism:



a. The script calls the “eccert” command. The “eccert” command reads the certificate from the user directory and sends it using SSL to the HTTP/S plugin.

b. The HTTP/S plugin checks the certificate using the gateway authentication mechanism. If the certificate is valid, a temporary password is created and returned to the “eccert” command, which writes it to its standard output and exits.

c. The script calls the FTP client providing the user-id and the temporary password. The FTP client connects to the FTP plugin using the temporary password.

d. The FTP plugin checks the gateway has issued this temporary password for the user-id provided. The FTP plugin accepts the connection. The script can now use the current FTP connection to process his requests.

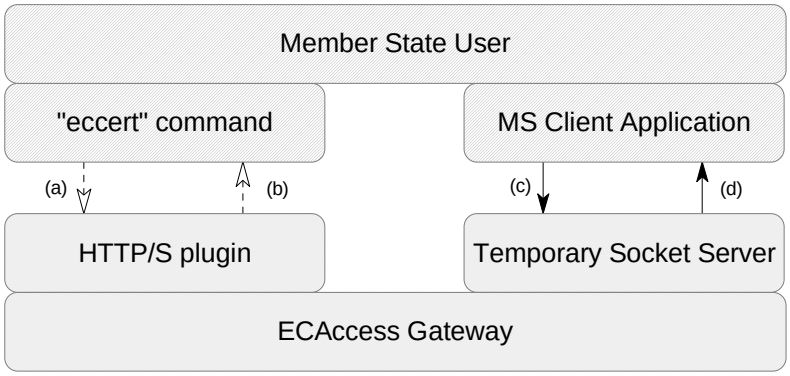
The ECaccess tools use this mechanism to authenticate the Member State user. However, the Member State user needs to create his ECaccess certificate beforehand.

A certificate is created from an ECMWF user identifier and a passcode (using a SecurID card). The certificate is saved in the user’s home directory and read when necessary by “eccert”.

Socket connector

The “eccert” command is used to access from any Member State client application to the corresponding server running at ECMWF on “Ecgate” (the temporary socket server accepts only one connection on a random port for a short period of time).

The following diagram explains the mechanism used:



a. The Member State user calls the “eccert” command providing the name of the server he wants to access on “Ecgate” (using the “-tunnel” option). The “eccert” command reads the certificate from the user directory and sends both the certificate and the service name using SSL to the HTTPS plugin.

b. The HTTPS plugin checks the certificate using the gateway authentication mechanism. If the certificate is valid, a temporary socket server is created and the random port (and the authorized user-id) is returned to the “eccert” command, which writes it to its standard output and exits.

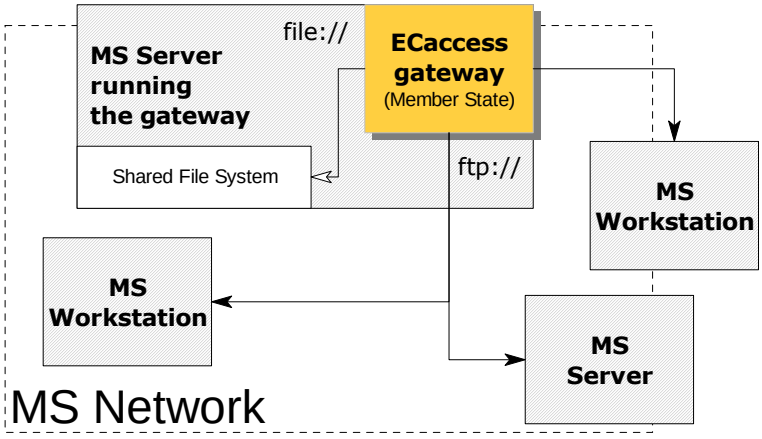
c. The Member State user calls the client application providing the user-id and the target port. The client application connects to the temporary socket server.

d. The gateway accepts the connection, opens the requested service on “ecgate” and plugs the client socket to the server application socket (it is verified that the system used by the user running the “eccert” command has the same IP address as the user running the client application). The client application can now use the connection, using its own protocol.

2.5 ECtrans file transfers

The “ectrans” command is provided for Member State users who log into their shell account at ECMWF. Its purpose is to submit a local or remote target file to the secure file transfer feature of the ECaccess Server.

For such a transfer, the ECaccess Server contacts the gateway that accesses the required file system:



The gateway receives the file transfer request from the ECaccess Server. This request includes an identifier in the format msuser@destination.

The gateway uses its database to map the specified destination to an “ectrans” association. This association includes the end location for the transfer (where the file is to be retrieved from or stored), the protocol used to access this end location and some protocol-specific options (e.g. ftp timeouts, use of temporary files, etc). The gateway then checks if the specified (local) msuser is authorized to access this association and perform the transfer.

Access protocols already provided with the gateway software are:

- FILE: access will be to the file system of the gateway Member State server: this file system can be shared with other servers or workstations using mechanisms such as NFS or SAMBA. The URL includes the full path of the directory to be accessed.

- FTP, FTPS or SFTP: access will be to a Member State workstation or server running a standard FTP, FTPS or SFTP standard server. The URL includes the login parameters (user-id and password) and the path of the directory to be accessed. Other protocol specific options can be provided through the association.

- EXEC: files can be stored (but not retrieved) on the file system of the Member State server running the gateway with a temporary name; renamed upon transfer completion to its intended name; a script is then started to process the new file.

Additional access protocols can be made available by the Member States. That will require development of in-house "ectrans" modules. These need to implement the same Java interface as the modules for the FTP, FILE and EXEC protocols described above.

3 ECACCESS INSTALLATION

ECaccess includes three packages:

- The ECaccess tools package: must be installed on each host from where ms users want to start Shell commands, or on a common file system.

- The ECaccess gateway package: must be installed on a secured platform (see the security requirements to decide on which platform the gateway should be installed).

- The ECaccess admin package: to allow remote administration of the gateway (only if the gateway package is installed).

Installing the tools package is mandatory. Installing the gateway package is optional, since you can use the ECMWF gateway.

However it is highly recommended to install the ECaccess gateway, since certain features (secure file transfer, secure tunnel to ECMWF, secure authentication, lower resource usage and web memory cache) are not available when using the ECMWF gateway.

3.1 Condition of use

Please note that the software includes RSA encryption routines. The RSA encryption is used to ensure the integrity of the data and the authenticity of any request.

Your country may have import restrictions on encryption software. However, please take note of Article 5 of the "PROTOCOL ON THE PRIVILEGES AND IMMUNITIES OF THE EUROPEAN CENTRE FOR MEDIUM-RANGE WEATHER FORECASTS":

"Goods imported or exported by the Centre and strictly necessary for the exercise of its official activities shall be exempt from ... Such goods shall also be exempt from all prohibitions and restrictions on import and export."

New versions of the ECaccess software may from time to time become available. These new versions may provide new features or fixes to software problems. Due to the Client-Server nature of the software a software protocol, which may have to be changed at some time in the future, is involved.

3.2 Downloading the packages

Three packages are available as gunzip and zip compressed files for downloading from "http://www.ecmwf.int/services/ecaccess/": the ECaccess tools package, the ECaccess gateway package and the ECaccess admin package:

- ecaccess-tools-v<version>.tar.gz
- ecaccess-gateway-v<version>.tar.gz
- ecaccess-admin-v<version>.tar.gz

3.3 Installing the ECaccess Shell command tools

The tools do not need to be installed as "root". They just need to be accessible (with read and execute privileges) to Member State users who wish to use the Shell commands. It is possible to install the tools on a file system shared by various platforms.

Directory structure

Assuming the compressed tools package has been downloaded into the appropriate directory (<software-dir>), the following (Bourne shell) commands would install the tools and would make them available to all users:

```
#>cd <software-dir>; version=<version>
#>gunzip ecaccess-tools-v$version.tar.gz
#>tar –xvf ecaccess-tools-v$version.tar
#>cd ecaccess-v$version/client
#>chmod a+rx .. */. */eccert tools/.ecaccess
```

The "<software-dir>/ecaccess-v<version>/client" directory contains three subdirectories:

Name	Content
ecsrc	Contains the "eccert" source.
ecbin	Contains a subdirectory for each supported UNIX platform. These include an "eccert" binary (AIX, HP-UX, SunOS, IRIX/64, Linux).
tools	Contains the main ".ecaccess" script and a symbolic link for each Shell command.

The ".ecaccess" script will invoke the appropriate "eccert" binary (using "uname", e.g. "ecbin/AIX/eccert").

Currently, “eccert” binaries are available for AIX, HP-UX, SunOS, IRIX/64 and Linux, but can be compiled for other platforms from the source (ecsrc/eccert.c). However, “eccert” is linked with the “openssl” libraries. If you need to compile “eccert” for a new operating system and these libraries are not already installed, they can be found at http://www.openssl.org/ for a variety of operating systems.

Tools customization

The “.ecaccess” script needs to be updated with the full path of the ECaccess home directory. Edit the script “.ecaccess” in the directory “ecaccess-v<version>/client/tools”, and set the _ECACCESS_HOME parameter:

```
_ECACCESS_HOME={full-path-to-ecaccess-directory}
```

The following parameters also needs to be updated:

- _ECHOST: this is the full Internet name (e.g. “ecaccess.meteo.ms”) of the gateway host (needs to match the “hostname” parameter in the “Login” group of the gateway configuration file).
- _ECFTPPORT: this is the FTP port of the gateway (needs to match the “ftp” parameter in the “Ports” group of the gateway configuration file).
- _ECCERTPORT: this is the HTTPS port of the gateway (needs to match the “https” parameter in the “Ports” group of the gateway configuration file).

To avoid errors due to incorrect PATH but taking into account that several hosts may share the tools, you may wish to set the full path of all the commands used from the “.ecaccess” script:

```
awk={full-path-to-the-awk-command}
basename={full-path-to-the-basename-command}
cut={full-path-to-the-cut-command}
echo={full-path-to-the-echo-command}
ftp={full-path-to-the-ftp-command}
grep={full-path-to-the-grep-command}
ln={full-path-to-the-ln-command}
sed={full-path-to-the-sed-command}
uname={full-path-to-the-uname-command}
which={full-path-to-the-which-command}
```

User environment

Member State users, who wish to use these tools, should be advised to add the “<software-dir>/ecaccess-v<version>/client/tools” directory to their PATH environment variable.

3.4 Installing the gateway

In a UNIX environment, the gateway must only be installed and started as "root" if the HTTP/S, FTP and telnet server needs to be used on privileged ports (such as standard ports 80, 443, 21 and 23). Otherwise, a special user-id can be created for the gateway.

All the following operations should be performed from the account of the selected user installing the gateway.

Directory structure

To extract the gateway package in a UNIX environment:

```
#>gunzip ecaccess-gateway-v<version>.tar.gz
#>tar –xvf ecaccess-gateway-v<version>.tar
```

This command must be started from the directory where the gateway is to be installed.

To extract the gateway package in a Windows environment, use the "winzip" application.

The directory structure of "ecaccess-v<version>/gateway" is:

Name	Content
bin	Commands to start and stop the gateway.
conf	ECaccess configuration files.
db	ECaccess database files.
dist	ECaccess distribution files.
ectrans	Root directory for secure file transfers with the genericFile destination.
htdocs	Root directory for the HTTP/S plugin (web resources).
lib	ECaccess gateway and plugins libraries.
log	Log files.
sql	ECaccess sql files for the database.
tmp	Temporary files.
unimars	Configuration files for the unimars plugin.
xsl	Configuration files for the HTTP/S plugin.

The "ectrans" directory is the default directory used by the "genericFile" destination to save files on the local machine. The log directory is used by the gateway to trace runtime information. The temporary directory is used for small files created and managed by the gateway runtime for its internal use.

To use “ectrans” for keeping files retrieved from ECMWF on the server running the gateway, it is recommended to mount the “ectrans” directory to a separate partition with free disk space. It is also possible to export this directory as a NFS directory to allow access from different systems.

Gateway environment

The gateway is a Java application, so Java needs to be installed and the “java” command must be available in the PATH of the user running the gateway (the gateway currently requires Java version 1.4.2+ on either a UNIX or a Windows platform).

The gateway start-up script needs to be updated with the full path of the ECaccess home directory.

UNIX

For a UNIX operating system, edit the script “gateway” in the directory “ecaccess-v<version>/gateway/bin”, and set the ECACCESS_HOME parameter (and optionally the JAVA_OPTS parameter if you need to pass specific parameters to your Java Virtual Machine - JVM):

```
ECACCESS_HOME={full-path-to-ecaccess-directory}
JAVA_OPTS=
```

Optionally, you can also set the full path of all the commands used by the “gateway” script:

```
java={full-path-to-the-java-command}
echo={full-path-to-the-echo-command}
egrep={full-path-to-the-egrep-command}
kill={full-path-to-the-kill-command}
rm={full-path-to-the-rm-command}
sleep={full-path-to-the-sleep-command}
cat={full-path-to-the-cat-command}
ps={full-path-to-the-ps-command}
nohup={full-path-to-the-nohup-command}
uname={full-path-to-the-uname-command}
find={full-path-to-the-find-command}
```

These settings avoid errors due to incorrect PATH.

Then you can change the default settings for the administration tools by editing the configuration file “starter.properties” in the “ecaccess-v<version>/gateway/conf” directory. You can change the “starter.listenAddress/starter.port” (interface/port to listen to for the administration Web server) and the “starter.user/starter.password” (user and password to log to all the administration tools) parameters.

Windows

For Windows, export the ECACCESS_HOME and JAVA_HOME variables environment (and optionally the JAVA_OPTS parameter if you need to pass specific parameters to your Java Virtual Machine - JVM).

To set these values you can also edit the script "gateway.bat" in the directory "ecaccess-v<version>\gateway\bin" and set (or uncomment) the following lines:

```
set JAVA_HOME={full-path-to-java-directory}
set ECACCESS_HOME={full-path-to-ecaccess-directory}
```

Then you can update the "starter.properties" configuration file like explained in the previous section.

Basic configuration

The general configuration file of the gateway is located in the "ecaccess-v<version>/gateway/conf" directory. Its name is "ecmwf.properties".

Once started, the gateway will connect and authenticate to the ECaccess Server. In order to perform this action, it is necessary to obtain from ECMWF a gateway password and a gateway certificate (consult the "http://www.ecmwf.int/services/ecaccess/login.jsp" page for how to proceed).

The gateway certificate must be saved in the "ecaccess-v<version>/gateway/conf" directory. Then, edit the "ecaccess-v<version>/gateway/conf/ecmwf.properties" file and set the "hostName" parameter with the Internet name of the host where the gateway is installed (eg. "ecaccess.meteo.ms"), and set the "password" parameter with the gateway password returned by mail:

```
[Login]
hostName=ecaccess.meteo.ms
password=xxxxxxxx
```

Then, optionally you can setup the following parameters:

```
ecaccessServer=ecaccess.ecmwf.int
listenAddress=0.0.0.0
externalAddress=${Login[hostname]}
```

The "ecaccessServer" parameter defines which ECaccess Server you want to connect to. To connect to ECMWF through the Internet, use "ecaccess.ecmwf.int". To switch to the RMDCN network, change it to "msaccess.ecmwf.int".

The "externalAddress" parameter make it possible to setup the ECaccess plugins to listen to a specific network interface (this can be useful if your system has more than one network interface and you want to restrict access to one of these). You can also set it on a per plugin basis by going through the "ecmwf.properties" file and updating the "listenAddress" parameter of each plugin.

The "externalAddress" parameter allows specifying which address should be used by the ECaccess Server to connect to your ECaccess gateway (through the call back port defined below). This parameter can be useful if you are doing NAT (Network Address Translation) on your network.

Then, you can set up the port parameters:

```
[Ports]
ftp=9021
```

```
http=9080
https=9443
telnet=9023
ssh=9022
callBack=9000
unimars=9108
database=9090
```

The HTTP/S, ftp, ssh and telnet servers are probably already running on the system. If the above settings are modified to use standard ports (80, 443, 21, 22 and 23), don't forget to first stop the existing HTTP/S, ftp, ssh and telnet applications. Otherwise, the gateway plugins won't be allowed to bind these ports.

Note that SSL is only used to secure the login process. Once authenticated, you will be redirected to the HTTP server. However, if you don't want to use the HTTP server and want all exchanges between the server and the browser to be secured (not only authentication), comment the "http" parameter:

```
#http=9080
```

This way, all exchanges between browsers and the HTTP server will be encrypted.

The data channel between ECaccess and ECMWF (which is used for data transfers and interactive sessions with ssh or telnet) is secured by an SSL connection. The algorithm used to encrypt the connection can be configured with the following parameter:

```
[ProxySocket]
cipherSuites=SSL_RSA_WITH_NULL_MD5
```

By default SSL is used to authenticate (exchange of certificates) and then a MD5 data integrity check is applied to connection. If you want full encryption of all the traffic through the data channel change this parameter to "SSL_RSA_WITH_RC4_128_SHA".

Finally, you may want to update the content of the text files used respectively by the telnet server (in the gateway/conf/telnet directory), the FTP server (in the gateway/conf/ftp directory) and the ssh server (in the gateway/conf/ssh directory) to display various user messages (during login, help, etc.).

Database configuration

The gateway includes an embedded database engine: HSQLDB is a relational database supporting a subset of ANSI-92 SQL. In some case it may be useful to couple the ECaccess gateway with an external database (e.g. to centralize the administration).

Database mode

The ECaccess database mode (SERVER, INTERNAL or EXTERNAL) is configured in the "DataBase" group of the "ecmwf.properties" configuration file.

Server mode

By default, the ECaccess embedded database is started internally (not as a server and it is therefore not accessible from the outside):

```
[DataBase]
repository=${ecmwf.dir}/gateway/sql
driver=org.hsqldb.jdbcDriver
level=2.0
protocol=jdbc
subProtocol=hsqldb
#server=hsqldb://${Login[hostName]}:${Ports[database]}
alias=${ecmwf.dir}/gateway/db/ecaccess
user=sa
password=
logECtrans=false
logEvents=true
purge=120
```

If the "logECtrans" parameter is set to "true" it will activate the record of the ECtrans transfers in the database (this information can be later used to monitor the transfer rates, frequency and total of transfers per user using the database monitoring tool). The "logEvents" parameter record the actions performed by each user on the Gateway (this information is then accessible by the users themselves through the Web interface).

If you want the gateway to act as a database SERVER, then uncomment the "server" parameter.

Internal mode

The ECaccess embedded database is started internally and is not accessible externally (not even by the database manager). To configure the database gateway in the INTERNAL mode, leave the "server" parameter commented out:

```
#server=hsqldb://${Login[hostName]}:${Ports[database]}
```

External mode

The ECaccess gateway uses an external database. ECaccess has been designed to seamlessly integrate with other relational databases that provide JDBC support. ECaccess uses only JDBC 1.0 API calls and an SQL subset to avoid problems with limitations of several JDBC drivers.

A typical setting to configure the ECaccess gateway to use an EXTERNAL DB2 database would be:

```
[DataBase]
…
dbms=Db2
driver=COM.ibm.db2.jdbc.app.DB2Driver
…
subProtocol=db2
alias=//[hostname]:[port]/[database name]
user=user
password=password
…
```

To specify the platform of the target RDBMS, the “dbms” parameter needs to be added. Its value must be one of: Hsqldb, Db2, Informix, MsAccess, MsSQLServer, MySQL, Oracle, PostgreSQL, Sybase or Sapdb.

The appropriate value for the “alias” parameter should be available in the target RDBMS documentation. The JDBC driver class named in the “driver” parameter must be available in one of the Java Archive files (JAR) in the “gateway/lib/ext” directory.

The ECaccess gateway requires several internal tables in the target RDBMS. These are created by the “admin/sql/dbcreate.sql” script as shown in the appendix “Database diagram” (the ECaccess admin package described in the next section).

If the execution of the script “dbcreate.sql” within the database fails, then it is necessary to adjust the types for the database used. E.g., the selected database may use the BOOLEAN type rather than the BIT type (which is used by HSQLDB to represent Boolean values). These adjustments must be done in “admin/sql/dbcreate.sql” and in the file “gateway/conf/repository.xml”. Note that similar adjustements might be required with the other “*.sql” files in the sql directory.

Note that the date and time format in the database need to be set to “yyyy-MM-dd” and “HH:mm:ss”.

3.5 Installing ECadmin

To start the database and log manager remotely from any platform, ECaccess provides a package called ECadmin. Like the gateway, ECadmin is a Java application and can therefore be started either on a UNIX or Windows platform. This section explains how to install the ECadmin package.

Note that using two different versions of Java for the gateway and the administration tools may be an issue under certain circumstances.

The installation process described in this section must be repeated on each workstation from which ECaccess administration tasks are to be performed.

The ECadmin distribution is available within the ECaccess gateway distribution. To extract the Ecadmin package in a UNIX environment:

```
#>gunzip ecaccess-admin-v<version>.tar.gz
#>tar –xvf ecaccess-admin-v<version>.tar
```

This command must be started from the directory where ECadmin is to be installed.

To extract the ECadmin package in a Windows environment, use the "winzip" application.

The directory structure of "ecaccess-v<version>/ecadmin" is:

Name	Content
bin	Commands to start and stop the database and log managers.
conf	ECadmin configuration files.
lib	ECadmin libraries.
sql	SQL scripts used by the database manager.
log	Log files.
tmp	Temporary files.

In order to use the ECadmin tools, the ECaccess database mode must be configured as EXTERNAL or SERVER. It is also recommended to set the "logECtrans" and "logEvents" parameters to "true" in the "ecmwf.properties" file (see previous section).

UNIX

For a UNIX operating system, edit the script "ecadmin" in the directory "ecaccess-v<version>/admin/bin", and set the ECACCESS_HOME parameter:

```
ECACCESS_HOME={full-path-to-ecaccess-directory}
```

Optionally, you can also set the full path of all the commands used by the "gateway" script:

```
java={full-path-to-the-java-command}
echo={full-path-to-the-echo-command}
egrep={full-path-to-the-egrep-command}
nohup={full-path-to-the-nohup-command}
find={full-path-to-the-find-command}
```

These settings avoid errors due to incorrect PATH.

Windows

For Windows, export the ECACCESS_HOME and JAVA_HOME environment variables (and optionally the JAVA_OPTS parameter if you need to pass specific parameters to your Java Virtual Machine - JVM).

To set these values you can also edit the script “ecadmin.bat” in the directory “ecaccess-v<version>\admin\bin” and set (or uncomment) the following lines:

```
set JAVA_HOME={full-path-to-java-directory}
set ECACCESS_HOME={full-path-to-ecaccess-directory}
```

4 STARTING AND TESTING

This chapter describes how to start the gateway, how to check it is properly installed and that it is running.

The following section assumes the gateway is installed and the commands will be started using the gateway shell account (which can be "root" or a specific user).

4.1 Starting the gateway

To start the gateway, go to the "ecaccess-v<version>/gateway/bin" directory and run the following command:

```
gateway start
```

To stop the gateway, run the following command:

```
gateway stop
```

In general, it is preferable to automatically start and stop the gateway daemon when the system is either started or stopped as shown in the following sections with the UNIX and Windows procedures.

UNIX set-up

Simply copy the "gateway" script to your "init.d" directory and create the symbolic links to the appropriate run levels. Examples for the Linux Redhat and SuSE distribution are shown below.

All the following commands should be run from the shell account of the "root" user.

Linux RedHat

For Linux Redhat use the following:

```
#>cp $ECACCESS_HOME/gateway/bin/gateway /etc/rc.d/init.d/.
#>chmod u+x /etc/rc.d/init.d/gateway
```

And then use "chkconfig" to create symbolic links to appropriate run levels (in this case levels 2, 3, 4 and 5):

```
#>chkconfig –level 2345 gateway on
```

The ECaccess gateway will now restart every time the system is rebooted.

Linux SuSE

For Linux SuSE use the following:

```
#>cp $ECACCESS_HOME/gateway/bin/gateway /etc/init.d/.
#>chmod u+x /etc/init.d/gateway
```

And then manually create the symbolic start links (called when entering a run level) and stop links (called when leaving a run level) to run levels 2, 3, 4 and 5:

```
#>ln -s /etc/init.d/gateway /etc/init.d/rc2.d/S22gateway
#>ln -s /etc/init.d/gateway /etc/init.d/rc2.d/K22gateway
#>ln -s /etc/init.d/gateway /etc/init.d/rc3.d/S22gateway
#>ln -s /etc/init.d/gateway /etc/init.d/rc3.d/K22gateway
#>ln -s /etc/init.d/gateway /etc/init.d/rc4.d/S22gateway
#>ln -s /etc/init.d/gateway /etc/init.d/rc4.d/K22gateway
#>ln -s /etc/init.d/gateway /etc/init.d/rc5.d/S22gateway
#>ln -s /etc/init.d/gateway /etc/init.d/rc5.d/K22gateway
```

Note the start and stop links include a number in their link name to control the order of the service starts and stops. In this example the service number is set to 22 but might be different for another system.

The ECaccess gateway will now restart every time the system is rebooted.

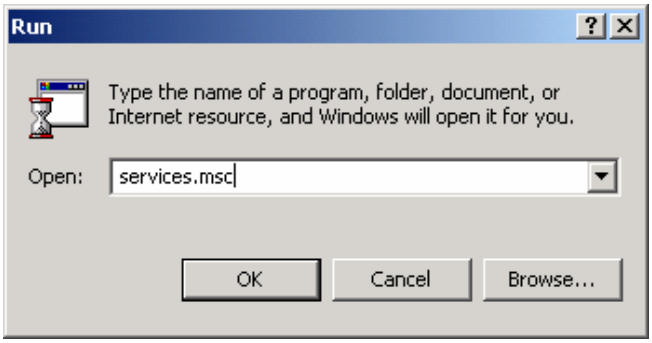
Windows set-up

Before setting up the ECaccess gateway as a Windows service, you need to ensure that the gateway has not been started already. If necessary, shut it down using the "stop" option of the "gateway" script (as explained above).

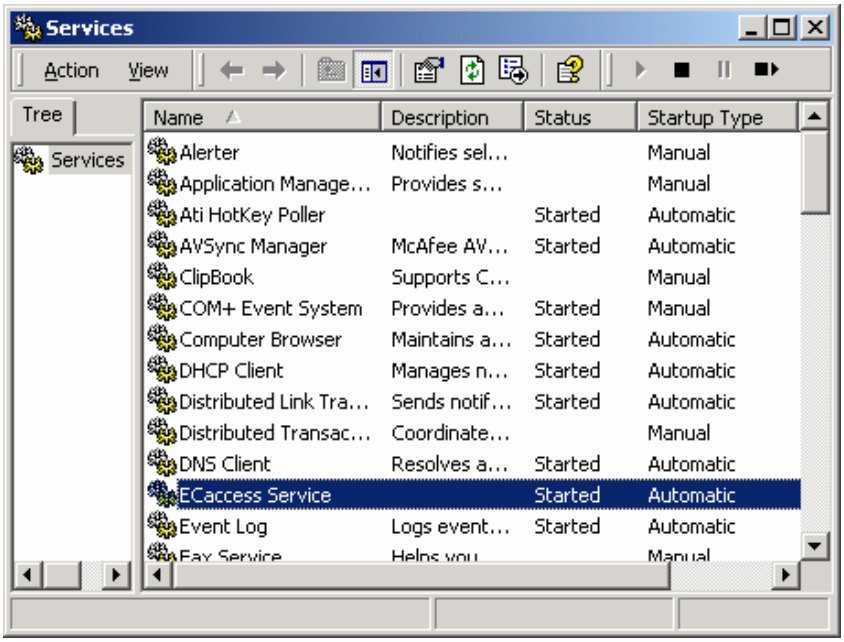
To set up the ECaccess gateway as a Windows service and manage it (start and stop) from the services application, run the following command from the "ecaccess-v<version>/gateway/bin" directory:

```
gateway install
```

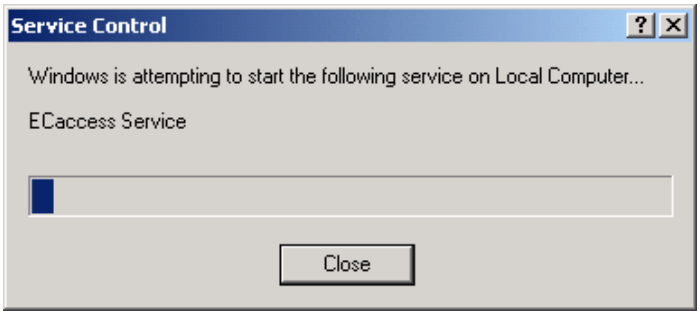
Then open the "Start" menu of Windows and select the "Run" option:



Enter "services.msc" and click the "OK" button. The services application is started:



Select the “ECaccess Service” in the list with the right button of your mouth and select the “Start” option to start the ECaccess gateway:



The ECaccess gateway is now started and will restart every time the system is rebooted.

4.2 Administering the gateway

This requires the ECaccess gateway to have been started (procedure explained in the previous section).

The ECaccess software provides a facility to administer the gateway through a Web interface. To access the administration site, point a Web browser to the following address:

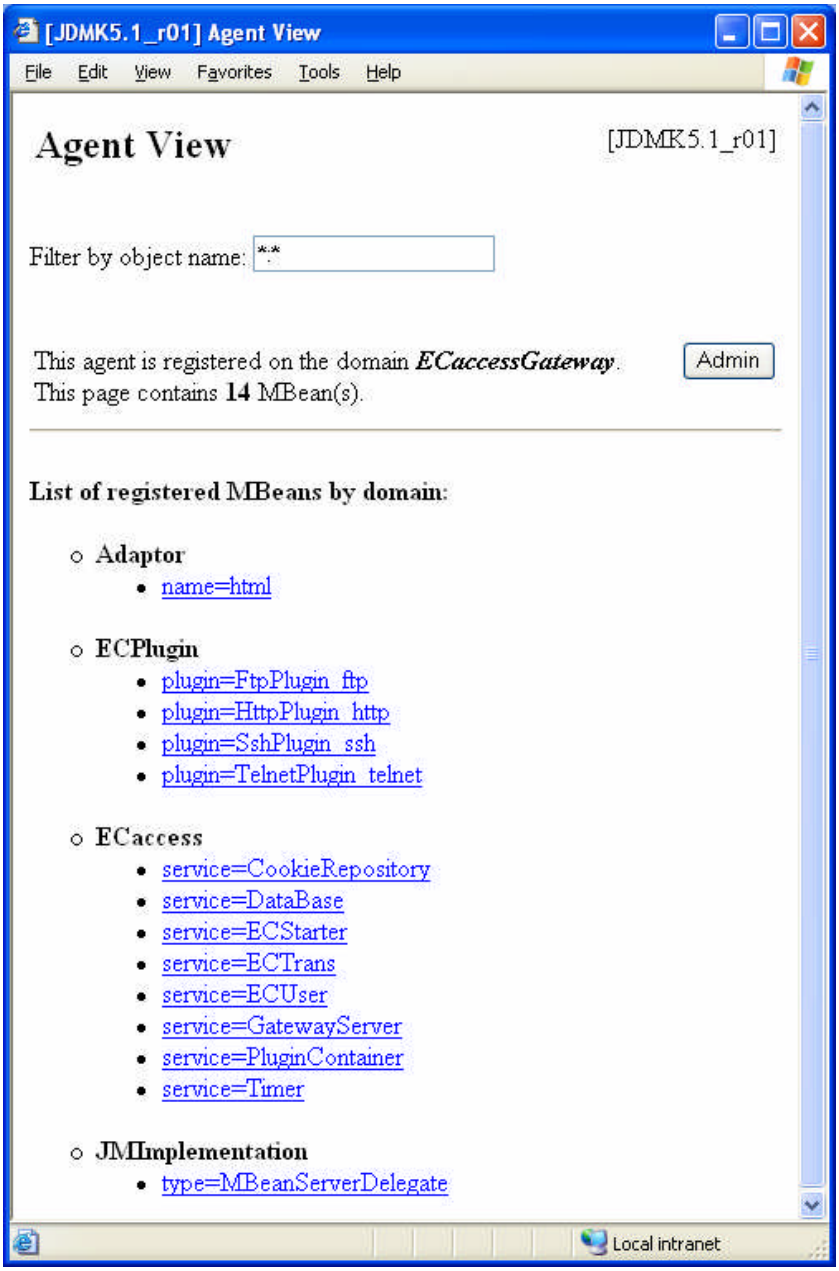
- "http://[starter.listenAddress]:[starter.port]/"

Where "starter.listenAddress" and "starter.port" can be found in the "gateway/conf/starter.properties" configuration file). Note that by default the "127.0.0.1" address is defined, which means that you can access this web site only from the server itself (this requires having a web browser installed on this system).

If you want to bind all the interfaces available on the server then replace "127.0.0.1" by "0.0.0.0".

You will be prompted for the admin "User Name" and "Password", which are by default "admin" and "admin" (same as the "user" and "password" values in the "Admin" group of the "ecmwf.properties" file).

Then, the "Agent View" is displayed in the browser window:



Each plugin administration interface can be accessed using its link in the “ECPlugin” group. Common services of the ECaccess gateway (and the gateway itself) can be accessed through the links in the “ECaccess” group.

To ensure that the gateway is authenticated and registered at ECMWF, click the “GatewayServer” service of the “ECaccess” group and check the MBean attributes “Connected” and “Registered”. They should be both set to “true” indicating the gateway is logged into the ECaccess server at ECMWF.

Then, click “Back to Agent View” and check each plugin status by clicking on the corresponding link (eg. FtpPlugin_ftp, HttpPlugin_http, TelnetPlugin_telnet, SshPlugin_ssh). The MBean attribute “Status” should read “ON”.

Each manageable component of ECaccess is called Mbean (for Management Bean of the JMX specification). By browsing the administration site it is possible to find the management interface of each Mbean to read (and optionally write) their attribute values (parameters) and perform predefined operations (such as start and stop for the plugins).

There is also an MBean which is dedicated to the Database. This bean display the main parameters related to the Database and also provide an SQL interface through the "execUpdate" and "execeSelect" buttons. It is possible to check for example the list of ECMWF users defined in the Database by executing the following "execSelect" request:

"select * from ECUSER"

The list of tables in the Database is provided in an Appendix of this document.

Moreover, ECaccess also includes a database and a log manager for administration of the ECaccess database (a compliant SQL database) and to browse the log.

The Database Manager

The Database Manager can be started from any workstation, on which the ECaccess admin package has been installed. Run the following command from the "ecaccess-v<version>/admin/bin" directory:

```
ecadmin start-dbmanager
```

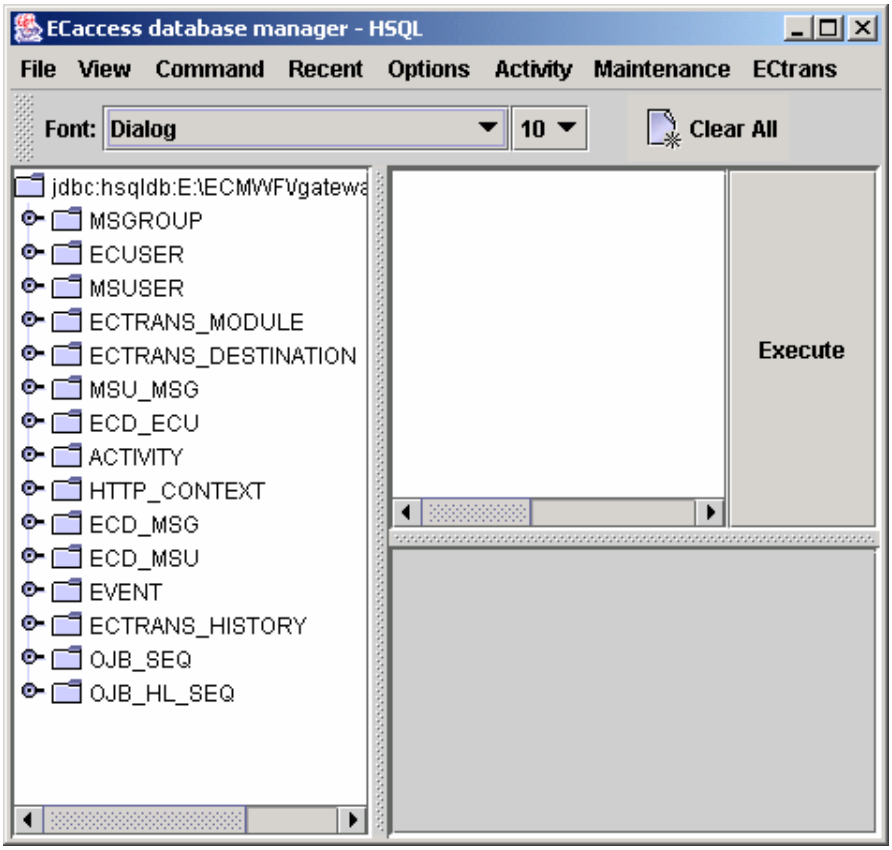
When prompted, provide the admin user and password (as specified in the "gateway/conf/starter.properties" script), the hostname of the server running the ECaccess gateway and the port of the HTTPS plugin (as specified in the "gateway/conf/ecmwf.properties" file).

A window (divided into three panes) will appear on the desktop:

- The left pane shows the tables and their fields as defined in the database. To expand tables or fields, click the handle on the left. Expanded fields show the type and whether null-values are allowed.

- The upper right pane is an editable text area for SQL requests. These can be initialised with the "File -> Open Script …" menu option, or the Activity, Maintenance or Ectrans menu options. The "Execute" button on the right will initiate the request shown.

- The lower right pane will display the result of the SQL request either as a table ("View -> Result as table") or as text ("View -> Result as text"). The result can be saved using the "File -> Save result …" option.



The “Clear All” button in the icon bar will reset the request and result panes.

The database manager can be used for administration of the database, monitoring the activity of the gateway or managing the ECtrans facility.

The Activity menu has two options:

- The “Service activity” option will generate and execute an SQL request for showing the activity of a selected or all services. You will be prompted for the names of the service to be monitored (select “%” for all services), the maximum number of result lines (default value set to 5000), the date to be monitored (in the format “YYYY-MM-DD”) and the order of the result (by time).

- The “ECtrans activity” option will generate and execute an SQL request for showing Ectrans transfers. You will be prompted for the name of the Member State user to be monitored (enter “%” for all Member State users), the date to be monitored (in the format “YYYY-MM-DD”) and the order of the result (by time).

The ECtrans menu has several options (please note that the new Gateway allow users to manage themselves their own associations, therefore this menu is not useful unless you have already created associations with previous releases of the Gateway and want to continue maintaining them by yourself):

- The “New module” option: to declare a previously developed in-house ECtrans module in the database. You will be prompted for the name (to be referred to by ECtrans destinations), the Java class name (in the format “myPackage.MyModule”), the location of the Java Archive file (JAR) containing the compiled (binary) code (the variable “${ecmwf.dir}” can be used to specify a path relative to the gateway installation directory) and the status (enabled or disabled) of the module.

- The “New destination” option: to declare a new ECtrans destination in the database. You will be prompted for the name of this destination (to be used in the “-remote msuser@destination” ectrans command option), the name of the ECtrans module to be associated with this destination (“genericFtp” and “genericFile” are available by default), the URL (see the section “ECtrans configuration” for more details) and the status (enabled or disabled) of the destination.

- The “New MS group” option: to create a new group of Member State users. You will be prompted for the name, a comment (e.g. “Project XXXXX”) and the status (enabled or disabled) of this group.

- The “New user association” option: to manually associate a Member State user with an ECMWF user (normally this would be done by the ECMWF users themselves, using the ECaccess Web interface). You will be prompted for the Member State user name (which is a local login name), the ECMWF user name to be associated with the Member State user, the hostname of the Member State user workstation, the destination directory and a comment. The password for this Member State user will not be set and the Member State user will be disabled. The ECMWF user will need to connect to the ECaccess Web site, set the password and activate the Member State user before using it (note that the password is stored encrypted in the database).

- The “Remove module” option: to remove the declaration of an ECtrans module and all its dependencies from the database (ECtrans history and destinations). You will be prompted for the name of the declaration.

- The “Remove destination” option: to remove the declaration of an ECtrans destination and all its dependencies from the database (ECtrans history and destinations). You will be prompted for the name of the declaration.

- The “Remove MS group” option: to remove a Member State group and all its dependencies from the database. You will be prompted for the Member State group name.

- The “Remove user association” option: to remove an ECMWF-Member State user association and all its dependencies from the database (ECtrans history). You will be prompted for the ECMWF and the Member State user names.

- The “MS group remove user” option: to remove a Member State user from a Member State group. You will be prompted for the Member State group name and the Member State user name.

- The “MS group add user” option: to add a Member State user to a Member State group. You will be prompted for the Member State group name and the Member State user name.

- The “Grant an MS group” option: to authorize a Member State group to use a specific destination. You will be prompted for the ECtrans destination name and the Member State group name.

- The “Grant an MS user” option: to authorize a Member State user to use a specific destination. You will be prompted for the ECtrans destination name and the Member State user name.

- The “Grant an EC user” option: to authorize an EC user to use a specific destination. You will be prompted for the ECtrans destination name and the ECMWF user names.

The Maintenance menu has four options:

- The “Create the database” option: to create the database from scratch (create all the tables and indexes). A diagram of the ECaccess database can be found in the appendix of this document.

- The “Init the database” option: to initialise the database with the default ECtrans modules and destinations.

- The “Delete the database” option: to delete the content of all the tables of the database (including the ECtrans set-up).

- The “Drop the database” option: to remove the database by removing all the tables and indexes. Note that the ECaccess gateway cannot run without the database.

All the options described above (in the “Activity”, “ECtrans” and “Maintenance” menus) correspond to files located in the admin/sql directory. When an option is selected, the SQL request found in the corresponding option file is loaded by the database manager and processed.

For example, the “ECtrans -> New module” option corresponds to the following set of directives:

```
##
## Menu options
##
#menu "ECtrans"
#name "New module"
#group "new"
#id "0"

##
## Prompt for variables
##
#prompt "name;The name for the new ECtrans module;"
#prompt "class;The Java class name;myPackage.MyModule"
#prompt "archive;The JAR file including the class $class"
#prompt "active;Activate the module $name?;true|false"

##
## Confirmation
##
```

```
#confirm "Do you really want to create the module $name?"

##
## The SQL request
##
INSERT INTO ECTRANS_MODULE
  (ECM_NAME,ECM_CLASSE,ECM_ARCHIVE,ECM_ACTIVE)
VALUES
  ('$name','$class','$archive','$active')
```

The "menu" directive associates an option with a menu; the "name" directive labels this option; the "group" directive optionally specifies a name used for grouping options with separation bars; and the "id" directive defines the order in which the options appear in the menu.

The "prompt" directives describe the parameters. The argument of the "prompt" directive is a list of two or three elements separated by semicolons. The first element is the name of the parameter; the second is the prompt to be displayed; and the third (optional) element is the default for the parameter. A list of values is separated by "|" can also be provided which will restrict the user to select from this list only.

The optional "confirm" directive will cause prompting for confirmation before proceeding.

Finally, the request is built (substituting the parameters with the specified values) and shown in the request pane of the window.

While the request is still displayed in the request pane, it is possible (in case of an error) to edit the request and to restart the modified request by clicking the "Execute" button. Successful requests are kept and can be recalled using the "Recent" menu.

Using the mechanism described above, new menus and new options can be added to the administration interface. Renewing the menus requires restarting the administration interface: Use the File -> Exit option and rerun "ecadmin start-dbmanager".

Several database managers can be active at the same time.

The Log Manager

The log manager can be started from any workstation, on which the ECaccess admin package has been installed. Run the following command from the "ecaccess-v<version>/admin/bin" directory:

```
ecadmin start-logmanager
```

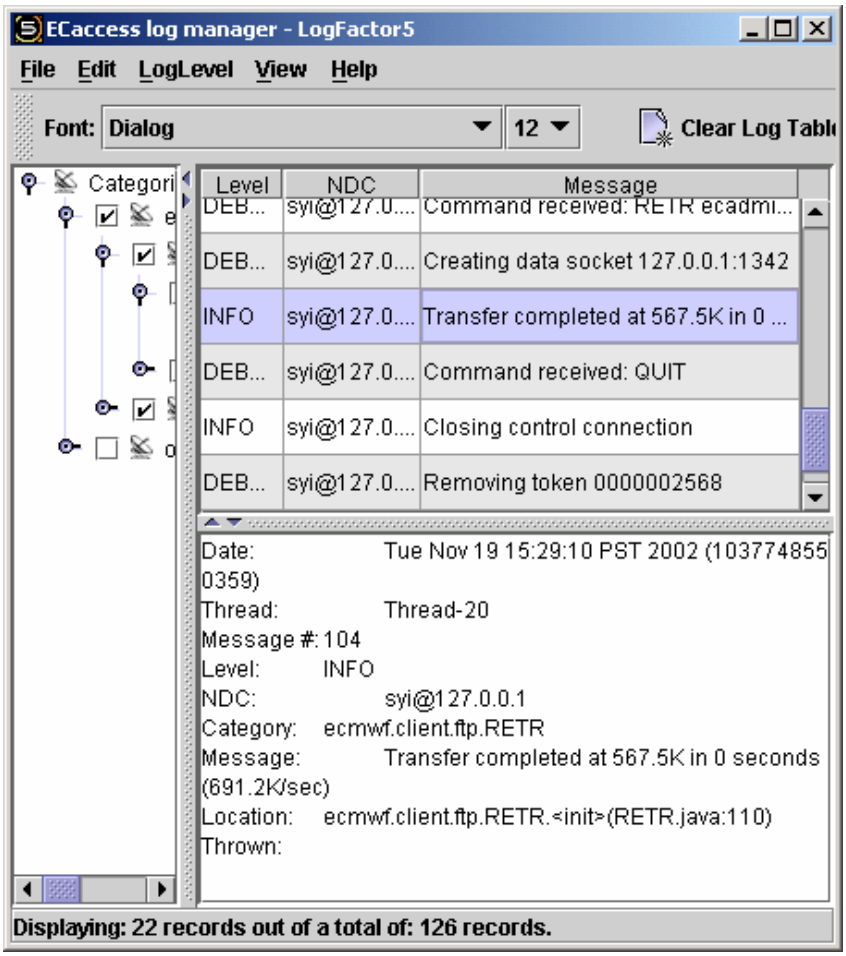
When prompted, provide the admin user and password (as specified in the "gateway/conf/starter.properties" script) and the hostname of the server running the ECaccess gateway and the port of the HTTPS plugin (as specified in the "gateway/conf/ecmwf.properties" file). The following window will appear on the desktop:

The window is divided into three panes:

- The left pane shows the application components of ECaccess (including the external software modules of the ECaccess application). To expand components click the handle on the left. To activate login for a component (or a sub-component) just tick the checkbox to the left of the component name. It is recommended to tick only the "ecmwf" components, since the other components cannot be managed (except for debugging).

- The upper right pane shows log entries for the selected components. Entries are summarized in a single line. The columns to be displayed can be (un)ticked in the "View" menu.

- The lower right pane shows the details for a log entry selected with the mouse in the upper right pane.



In the "LogLevel" menu the logging levels to be shown can be (un)ticked. Also, with the "LogLevel -> Configure LogLevel Colours" option, the colours to be used for each level can be configured (e.g. red for fatal errors and green for info).

Log messages can also be diverted to other targets (files or syslog daemons for example); for more information refer to chapter 4.5 "Checking the logs".

Several log managers can be active at the same time.

4.3 ECtrans configuration

This section is useful if you plan to develop and install your own ECtrans transfer module, or if you have already set-up ECtrans associations in the past with the database manager and you still want to manage these associations yourself (the new Gateway allow users to manage themselves their own associations).

ECtrans uses three tables in the database: the ECTRANS_MODULE table contains pointers to the compiled (binary) module code; the ECTRANS_DESTINATION table protocol parameters; and the ECTRANS_HISTORY table the ECtrans transfer history.

ECtrans destination

When declaring a new destination within the Database Manager (option "ECtrans -> New destination"), one of the parameters prompted for is a URL. The gateway uses it to determine the protocol-specific options to be passed to the ECtrans module. The URL can include the following references in its text:

- Member State user references: $msuser[name], $msuser[comment], $msuser[dir], $msuser[host], $msuser[passwd]. Member State users will need to provide settings for one or more Member State users through the "ECtrans set-up" facility of the WEB interface

- ECMWF user references from the database: $ecuser[name], $ecuser[uid], $ecuser[gid], $ecuser[dir], $ecuser[shell], $ecuser[comment].

- References to "ectrans" command options: $location is the location directory specified within the "target" option (in the format [location/]filename); $remote is the value of the "remote" option (in the format msuser@destination).

```
#>ectrans –gateway ecaccess.meteo.ms \
          -remote msuser@destination \
          –source ./source-filename
          -target [location/]filename
```

- A reference to Java system properties in the format ${VARIABLE_NAME} or any other JAVA variable (set with "-D[VARIABLE_NAME]=[variable]") in the start-up script "gateway/bin/gateway", such as the ECaccess distribution directory ${ecmwf.dir}.

The destination table also includes the ECD_RESTRICT and ECD_RESOLVE fields. If ECD_RESTRICT is set to false then the destination is made available to any ECMWF user; and if ECD_RESOLVE is set to true, the path set in $location can include ".." as a path element.

ECtrans authorization

Declared destinations can be authorized to either an ECMWF user (and all its Member State users), or to a Member State user or to a group of Member State users. These authorizations can be set up through the "ECtrans" menu of the database manager.

ECtrans checks the ECD_MSU table if the Member State user is authorized for the requested destination. Otherwise, ECtrans checks the ECD_MSG table if the Member State user belongs to an authorized Member State group or the ECD_ECU table if the ECMWF user, who has created the Member State user, is authorized. If none of these checks grant access, the transfer is aborted.

Note that the first connection of an ECMWF user to the gateway causes its registration in the database MSUSER table. From then on any Member State user creation or authorization association will reference it.

Nevertheless, for ECMWF user not in the database yet, Member State users can be created manually with the Web management interface, service "ECUser" from the ECaccess group: in the MBean operations section, provide the "ecuser" (ECMWF login) parameter in "Description of importRegisteredUser" section and click the "importRegisteredUser" button to import the ECMWF user information into the database.

ECtrans modules

This section presents the ECtrans modules provided with the gateway software and explains how to develop an in-house ECtrans module for specific requirements.

The ECtrans modules provided with the gateway software are "file", "ftp", "ftps", "sftp" and "exec" (see also "2.5 ECtrans file transfers"). The Jar file for these modules is "gateway/lib/ectrans/ectrans.jar".

A new ECtrans module, which has been developed, must be declared (this can be done with the database manager option "ECtrans -> New module"). Subsequent declarations of ECtrans destinations can refer to it (see "The Database Manager").

Such a new ECtrans module must implement the same Java interface as the "file", "ftp", "ftps", "sftp" and "exec" modules. For the development of a specific ECtrans module, a basic knowledge of the Java development language and a Java Development Kit are essential.

The following is the source code of the Java abstract class to be extended by each ECtrans module:

```
package ecmwf.common.ectrans;

import java.io.IOException;…

public abstract class TransferModule {
 …
 public abstract void connect(
   String location,
   ECtransSetup setup) throws Exception;
 public void check(
   long sent,
```

```
   String checksum) throws IOException { … }
 public abstract void close() throws IOException;
 public abstract void del(String name) throws IOException;
 public InputStream get(
   String name,
   long size) throws IOException;
 public OutputStream put(
   String name,
   long posn,
   long size) throws IOException;
 public abstract long size(String name) throws IOException;
 …
}
```

The ECaccess gateway first calls the connect() method, then the size() method if the resume or append option is used and then gets an output stream through the put() method. Parameters provided to the put() method are the name of the target file, the starting position within the file and the expected size of the file. The gateway then writes, flushes and closes the data stream through the close() method of the ECtrans module (the check() method is also called just before the close() method to allow additional checking). The del() method can be called if the erase option has been selected.

Note that the get() method is called by the gateway if the "-get" option is used in the "ectrans" command. In this case, the transfer direction is from the gateway to ECMWF.

The source code for the "ftp", "ftps", "sftp", "file" and "exec" modules can be found in the "ectrans.jar" archive.

Suppose you want to develop a new ECtrans module called "MyModule". First you have to provide the implementation of the module (in the "MyModule.java" file) and then compile the module:

```
#>javac –classpath $ECACCESS_HOME/gateway/lib/ext/gateway.jar \
      ecmwf/common/ectrans/module/MyModule.java
```

Then add to the archive file the new ECtrans module:

```
#>jar –uvf $ECACCESS_HOME/gateway/lib/ectrans/ectrans.jar \
      ecmwf/common/ectrans/module/MyModule
```

It is not necessary to restart the gateway.

Note that if the new module implementation needs extra libraries, they must be either added to the "gateway/lib/ext" directory or added to the module archive file path. When libraries are added to the "gateway/lib/ext" directory they are not automatically reloaded. Changes to these libraries require a gateway restart. The gateway log feature is based on the "log4j" Java package. This package allows the module to record messages in the log file. For more information about these logging facilities, see section "4.5 Checking the logs" or refer to http://jakarta.apache.org/log4j/docs.

4.4 Checking the gateway

The gateway consists of a set of services. The gateway is running fine if all services are functional.

The first section shows how to monitor FTP, HTTP/S and telnet servers. The second section shows how to check the "ectrans" module configuration.

Control servers

In order to avoid local network issues, all the following procedures should be started from the gateway server.

If the gateway services can be reached from the local host but not from a distant host, first check that there are no firewalls involved.

To test the ECaccess batch tools, which use the ECaccess FTP and HTTPS services, create an ECaccess certificate:

```
#>eccert –verbose
echost: ecaccess.meteo.ms
ecport: 9443
eccert: /home/xyz/.eccert.crt
Certificate request
ECMWF user identifier: xyz
Passcode from your SecurID card:
Certificate saved (855 bytes)
```

The verbose mode shows the HTTP/S server running on "ecaccess.meteo.ms" has been contacted on port "9443", the certificate request has been sent and a new certificate has been received (and saved).

Then try an "ec" command such as "ecls":

```
#>ecls local
HP-UX
OSF1
SunOS
#>
```

If you don't get the expected result, check the environment parameters (described in the section 5.1 of the "ECaccess User's Manual"). Alternatively refer to the last section of this chapter to read the gateway logs.

To test the telnet server, just try logging in with a telnet command:

```
#>telnet ecaccess.meteo.ms 9023
```

The following is the expected result:

```
Authorized access only.
*********************************************
   For further information, read the ECaccess
   documentation at:
   -> http://www.ecmwf.int/services/ecaccess/

   You can also use ECaccess to load/download
   files from your EChome, ECscratch  or ECfs
   directories using the ECaccess FTP server:
```

```
   -> ftp://uid@ecaccess.ecmwf.int/

   Use your UID and the SecurID code to login!
*******************************************
TelnetPlugin v<version>
login: xyz
Passcode:XXXXXX
```

If you don't get this kind of dialog, check the port. The port is specified in the "ecmwf.properties" configuration file. If you still don't get the expected result, refer to the last section of this chapter below to read the gateway logs.

Check ECtrans

The ECtrans configuration can be tested using the administration Web interface, going to the "ECtrans" service in the "ECaccess" group.

First, to check whether the ECtrans feature is enabled, take a look at the "Activated" MBean attribute. It should be set to "true". The feature can be enabled or disabled setting the value to true or false and clicking the apply button at the end of the "MBean attributes" section.

The operations available in the "MBean operations" section all use the same parameters:

- ecuser: the ECMWF login of the user, under which should be started the "ectrans" command.
- remote: the "-remote" option of the "ectrans" command in the format msuser@destination.
- target: the target file name in the format [location]/filename.

The operations available are:

- getURL: to get check the Member State user is attached to the EC user and is allowed to access this destination. The real URL is given in return.
- del: to delete the file (if check is successful).
- size: to get the size of the file (if check is successful).

To activate one of these operations, fill in the parameters and click the associated button.

4.5 Checking the logs

The logs are managed by the gateway and are stored by default in the "gateway/log/gateway.log" file.

The log feature of the gateway is based on the "log4j" Java package and can log messages at five priority levels:

- DEBUG: debugging messages, this should be suppressed in production.
- INFO: messages similar to the verbose mode of many applications.
- WARN: warning messages, which are logged to some log but the application is able to overcome the problem.
- ERROR: application error messages, which are logged to some log. But the application may still be able to continue, such as when an administrator supplied incorrect configuration parameters and the application falls back to using some hard coded default values.
- FATAL: critical messages, after logging of which the application quits abnormally.

The default logging configuration of the gateway can be modified in the "gateway/conf/log4j.properties" file. This file is periodically checked (each 30s) and automatically reloaded if necessary.

For example, to change the INFO priority level (default) to DEBUG, edit the logging configuration file and change the line:

```
level=INFO
```

to:

```
level=DEBUG
```

The default target of the log output is a file, but can also be a remote UNIX syslog daemon or the console (among many other output target, but this is out of the scope of this document and more details can be found at http://jakarta.apache.org/log4j/docs).

For example, to log to the local syslog daemon, edit the logging configuration file and change the line:

```
appenders=RollingFile
```

to:

```
appenders=RollingFile,Syslog
```

One or many output target can be specified at a time.

To log to a remote syslog daemon just change the following line:

```
ostname=localhost
```

to the name of the remote host running the syslog daemon.

5 UPGRADING

When a new version of the ECaccess Gateway become available, it is possible to use a script to automate the upgrade. This script is available for download from the following address:

- "http://www.ecmwf.int/services/ecaccess/download"

The script will automate the download of the new Gateway software. The existing configuration will be copied over to the new Gateway and the current Gateway will be stopped. It will then be up to the administrator to start the new Gateway and also to make it a boot time, it this is set up.

To do the upgrade the administrator should simply:

- Move the upgrade script called "upgrade-v<version>.sh" to the "gateway/bin" directory on the existing Gateway.

- Run the upgrade script as the UID under which the Gateway runs, while the Gateway is still running.

In case of problem, a copy of the output from the upgrade script should be sent to "ecaccess@ecmwf.int".

6 SECURITY

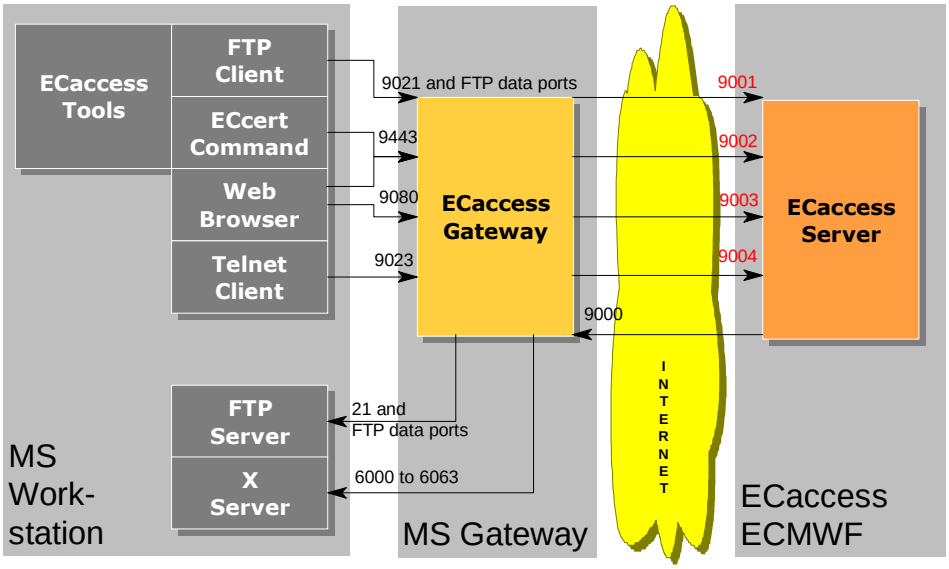
The gateway needs to access other computers and to start daemons:

- A number of servers, which listen on a number of ports, are started by default: you might want to firewall these ports and/or to close the corresponding server (when possible).

- A number of ports of the ECaccess Server must be accessible to the gateway: you must authorize connection to these ports.

Ports bound by the gateway are shown in the first section. The second section shows connections required by the gateway to the ECaccess Server.

The following diagram describes the streams between a Member State workstation, the ECaccess gateway and the ECaccess server:



All ports on the Member State gateway are configurable (default values are shown in black on the diagram). Ports on the ECMWF ECaccess server are not configurable (in red).

6.1 Open ports

Ports used by the gateway are configured in the gateway general configuration file.

Default ports are shown in the following table, but can be modified in the “ecmwf.properties” file:

Port	Purpose	Open to
9080	HTTP plugin (standard: 80)	Member State users
9443	HTTP/S plugin (standard: 443)	
9021	FTP plugin (standard: 21)	
9022	SSH plugin (standard: 22)	
9023	Telnet plugin (standard: 23)	
9082	Administration Web site	Administrator
9090	Database port (in SERVER mode)	
9000	Call back port	ECaccess Server

The call back port is mandatory and is used by the ECaccess Server to send notifications to the gateway.

6.2 Connections

The gateway must access the ECaccess Server and the Member State workstations (for X Window, FTP, FTPS and/or SFTP):

Port	Host	Purpose
9001	ECaccess Server	Naming service.
9002		Naming server.
9003		Data channel.
9004		Command channel.
6000-6063	Member State workstations	X Window System.
21 (and data ports) or 22		FTP/S and/or SFTP servers (ECtrans)

All connections to the ECaccess Server are mandatory.

6.3 Security manager

It is possible to use the "gateway/conf/ecmwf.policy" file to set-up a security policy for the ECaccess gateway. During its execution, when the gateway requests access to a critical system resource (such as file I/O and network I/O) the application invokes a special Access Controller module that evaluates (thanks to the "ecmwf.policy" file) the request and decides if it should be granted of denied. This policy file can be modified by hand or using the "policytool" application that comes with the Java package. Entries in the "ecmwf.policy" file use the standard java.policy file format as described in the Java documentation.

For example, to restrict the gateway file access to the ECaccess directory, change the line:

```
Permission java.io.FilePermission
      "<<ALL FILES>>",
      "read,write,delete,execute";
```

to:

```
Permission java.io.FilePermission
      "${ecmwf.properties}/-",
      "read,write,delete,execute";
```

Or, to restrict the gateway network access to the ECaccess server and to a sub-domain (say "meteo.ms"), change the line:

```
Permission java.net.SocketPermission
      "*:*",
      "connect,accept,listen,resolve";
```

to:

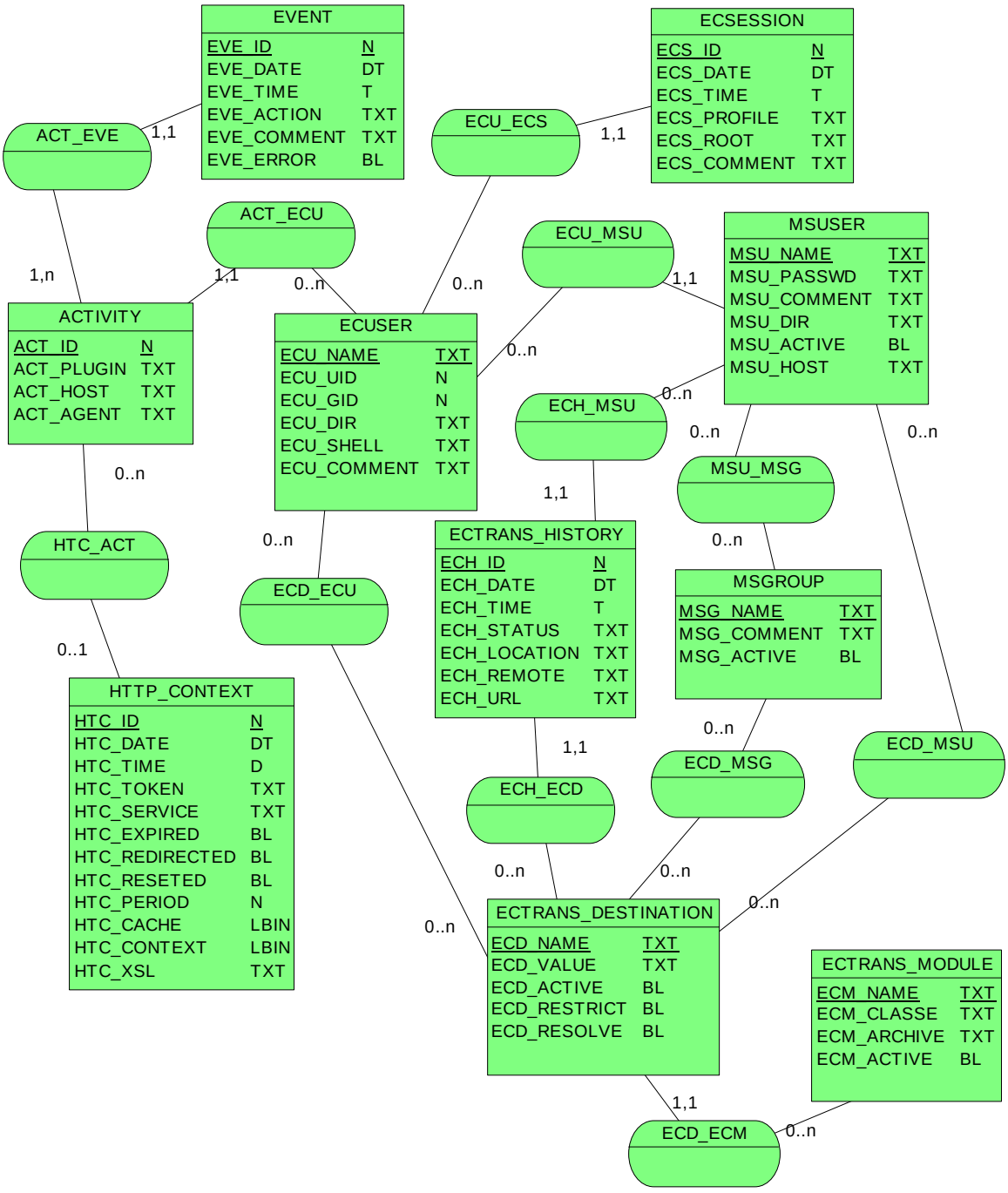
```
Permission java.net.SocketPermission
      "localhost:*",
      "connect,accept,listen,resolve";
Permission java.net.SocketPermission
      "*.meteo.ms:*",
      "connect,accept,listen,resolve";
Permission java.net.SocketPermission
      "ecaccess.ecmwf.int:9000-90004",
      "connect,accept,listen,resolve";
```

To find more information on how to configure this file, please refer to the following document:

- http://java.sun.com/j2se/1.4/docs/guide/security/spec/security-specTOC.fm.html

7 APPENDIX

7.1 Database diagram



7.2 Database tables

The tables of the ECaccess database are:

Table name	Purpose
MSGROUP	Member State group of users
MSUSER	Member State users
MSU_MSG	Member State users and Member State group associations
ECUSER	ECMWF users
ECU_MSU	EC users and Member State users associations
ECTRANS_MODULE	ECtrans modules
ECTRANS_DESTINATION	ECtrans destinations
ECD_MSG	Member State groups and destination associations
ECD_MSU	Member State user and destination associations
ECD_ECU	EC users and destination associations
ECD_ECM	Modules and destination associations
EVENT	Events for an activity
ACTIVITY	Activities for an EC users
ACT_ECU	EC users and activities associations
ACT_EVE	Events and activities associations
ECTRANS_HISTORY	ECtrans history for and Member State user
ECH_MSU	Member State user and history associations
ECH_ECD	Destinations and history associations
HTTP_CONTEXT	HTTP context for Web sessions
HTC_ACT	Activities and contexts associations
OJB_SEQ	Reserved for internal usage
OJB_HL_SEQ	