AutoDRM

Version 2.1

User Guide

Nanometrics INC.
SEISMOLOGICAL INSTRUMENTS
Kanata, Ontario • Canada

AutoDRM Version 2.1 User Guide

The information in this document has been carefully reviewed and is believed to be reliable Version 2.1. Nanometrics Inc. reserves the right to make changes at any time without notice to improve the reliability and function of the product.

Nanometrics Inc.
250 Herzberg Road
Kanata, Ontario, Canada K2K 2A1
Tel +1 613-592-6776
Fax +1 613-592-5929
Email info@nanometrics.ca
www.nanometrics.ca

Part number 14830R4
Release date 2008-04-10

About This User Guide

Document Scope

This user guide provides information for users who need to obtain data from or send commands to a Nanometrics data acquisition system, and system administrators who need to install, configure, and maintain AutoDRM.

- Chapter 1 Getting Started – Installation instructions and an overview of the components and functionality of AutoDRM
- Chapter 2 Configuring AutoDRM – Instructions on how to configure AutoDRM using the AutoDRM.ini file
- Chapter 3 Running AutoDRM – Information on stopping and starting AutoDRM, using the run-time commands, and monitoring the operation of AutoDRM
- Chapter 4 Using AutoDRM – A summary of AutoDRM message structure and message line descriptions.
- Appendix A Configuration File Example – An overview of the structure and an example of the AutoDRM.ini file
- Appendix B Message Syntax – An overview of the syntax for each of the supported message types and examples of message formats

Document Conventions

Essential and Supplementary Information:		
⚠	**Caution**	A Caution is essential information that explains (1) a risk of damage to equipment, data, or software where the recovery is likely to be troublesome; and (2) preventive action.
✏	**Note**	A Note is an explanation or comment that is related to the main text but is not essential information.
Links:		
	blue text	An external link; for example http://www.nanometrics.ca A link to information within the document.
Text Conventions:		
	bold text	Buttons on the graphical user interface (GUI).
	italic text	Variables such as parameter names and value placeholders
	`courier text`	File names and paths; for example ... `/nmx/user/trident.rsp`
	`courier bold text`	Input commands shown exactly as they must be entered at the prompt For example: ... and then type mkdir **`$APOLLO_LOCATION/config`**.

Contents

Chapter 1
Getting Started

Chapter 2 Configuring AutoDRM

Chapter 3 Running AutoDRM

Chapter 4 Using AutoDRM

Appendix A Configuration File Example

Appendix B Message Syntax

Tables

Chapter 1
Getting Started

AutoDRM Version 2.1 is an Automatic Data Request Manager that allows authorized users to request data from a Nanometrics data acquisition system and to send command requests, via email. Nanometrics AutoDRM Version 2.1 message formats comply with International Monitoring System (IMS), International Data Centre (IDC) conventions for AutoDRM S/H/I station basic message exchange, as defined in [IDC 3.4.1], and command request extensions defined as IMS2.0 in Attachment 6 to the Technical Terms of Reference for IMS Equipment (October 2003).

1.1 About AutoDRM

AutoDRM Version 2.1 supports IMS1.0/2.0 request and response message formats and uses email as the data return mechanism. It supports the following message conventions:

- Request lines:
  - To select data types – `HELP, STATION, CHANNEL, WAVEFORM, RESPONSE,` and `OUTAGE`.
  - To issue commands:
    - Operation change – `START_CONTINUOUS, STOP_CONTINUOUS, CALIBRATE_START,` and `CENTER_MASS`
    - Key management – `UPDATE_CRL, GENERATE_KEYPAIR,` and `START_KEYPAIR`
- Request control line to define the response message protocol – `E-MAIL`.
- Request and response environment lines:
  - To delimit the time and source of the requested data – `TIME, CHAN_LIST,` and `STA_LIST`.
  - To assign a time when the command request or response is issued – `TIME_STAMP`.
  - To define other command or response-specific constraints (see Chapter 4).
- Data type and command response lines (see Chapter 4).

1.1.1 Typical Operation

AutoDRM runs as an online service on the data acquisition network. It requires a stand-alone mail server with a mailbox for AutoDRM. It is usually best for AutoDRM to run on the NAQS Server acquisition computer because it must access the NAQS configuration files and ringbuffers. Various objects must be saved on the security token before digitally signed email messages can be exchanged (see Section 1.1.1.4 “Message Authentication” on page 3, and the SMConsole manual).

To generate responses to request messages, AutoDRM retrieves station and channel information from the `Naqs.stn` file, waveform and outage data from the NAQS ringbuffers, and sensor and digitiser response characteristics from the `*.rsp` files specified for each channel defined in the `Naqs.stn` file. AutoDRM retrieves Help information from the station help text file, if it has been installed. All AutoDRM status information is shown in the console window and is also saved into daily log files. AutoDRM optionally will log copies of request and response messages (it creates a *Message-ID* to track the message copies). For more information, see Section 1.2 “Summary of Inputs and Outputs” on page 5.

AutoDRM uses the appropriate application to implement commands at the digitiser and at the security token (for example, `CALIBRATE_START` is processed through NaqsServer).

1.1.1.1 Data Compression

For any station, `WAVEFORM` data for each channel can be sent in uncompressed format INT (as ASCII), or in CM6 compressed format. CSF (a signed compression scheme) can be used for stations or digitisers sending authenticated data in CD1.1 format. See [IDC 3.4.1] for information about these data compression schemes.

1.1.1.2 Missing Waveform Data

For `WAVEFORM` data outages, missing data are represented by an outage table giving the number of missing samples.

1.1.1.3 Message Exchange

AutoDRM can use either Post Office Protocol version 3 (POP3) or Internet Message Access Protocol version 4 (IMAP4) to receive email messages, and uses a Simple Mail Transfer Protocol (SMTP) mail service to send response email messages. The mail settings are specified with parameters in the [Interface] section of the configuration file.

A request message email can contain multiple request messages of more than one type. Each of the IMS request message blocks (bounded by `BEGIN` and `STOP`) within the email can contain one or more data requests, or one command request. As appropriate for each authorized request message, AutoDRM creates a response message containing the data responses to each of the original data requests, or executes the command and sends a response. AutoDRM obtains data from the NAQS Server configuration or data files to generate the data responses and uses the appropriate application to execute commands.

Request and response messages must use specified formats. See Chapter 4 “Using AutoDRM”, Appendix B “Message Syntax”, and [IDC 3.4.1].

AutoDRM sends each response message as one or more email messages. Data segments exceeding the user-defined maximum response message size are split into multiple parts, each of which is sent as a separate email. Each part is numbered (part 1, part 2, ... , part *n*); the final part also contains the total number of parts *n*. The maximum response message size is specified in the [Message] section *ResponseSizeLimit* parameter in the configuration file. Future-dated commands are answered with a confirmation email, followed by additional responses when the action occurs.

1.1.1.4 Message Authentication

For typical installations AutoDRM will only accept request messages sent as email in Secure/Multipurpose Internet Mail Extensions (S/MIME) format, addressed to the correct recipient (as specified in the [Interface] section *MailBox* parameter), and digitally signed by an authorized sender (a sender whose identity and access level can be verified according to information stored on the security token). If a request message cannot be verified, or if the certificate chain contained in the digital signature is not valid, AutoDRM rejects the request and does not send any acknowledgment to the originator of the message.

For installations in which clients do not have mail-signing capability you can configure AutoDRM to accept unsigned email requests by setting the configuration file [Control] section parameter *AcceptUnsigned* to `Yes`. Even if unsigned requests are accepted, signed requests will still be verified (and will be rejected if they cannot be verified).

AutoDRM will only send response email that has been signed using a private key stored on the security token. The S/MIME signature on outgoing messages will include the public key email signing certificate corresponding to the signing key.

1.1.1.4.1 S/MIME Email Format

AutoDRM will only exchange email messages using multipart/signed S/MIME format. A multipart/signed message has an email message header and a two-part message body. The first part of the body contains the message text, and the second part contains the digital signature of the message and the digital certificate of the sender. See [RFC 1847] for S/MIME standards and [RFC 2633] for the standards for adding cryptographic services to MIME data.

1.1.1.4.2 Digital Signatures and Digital Certificates

AutoDRM uses a public key infrastructure (PKI) based system for message authentication. A PKI relies on the use of a public-private key pair generated from the same algorithm as the basic tool for message and sender verification.

A key pair holder keeps their private key private, and typically has their public key posted in a public directory. The private key is used by the key pair holder to digitally sign outgoing messages. The public key is used by recipients of messages from the key pair holder to decrypt the message digital signature, and thereby verify the authenticity of the message.

A digital signature can be generated for any outgoing message, using the private key of the sender to encrypt a hash (an efficient mathematical representation of the data, derived using a hash function such as SHA-1) of the message. The public key of the sender is used by the message recipient to decrypt the hash of the received message. If the hashes match, the signature—and therefore the authenticity of the message—is verified.

A PKI provides a level of sender verification through the use of digital certificates in addition to the message verification provided by the use of key pairs. A trusted certificate authority (CA) will issue a digital certificate to an applicant whose identity has been verified. The certificate contains the name of the applicant (associated with a specific email address), a serial number, the period of time for which the certificate is valid, a copy of the public key of the applicant, and a copy of the digital signature of the CA.

AutoDRM requires that a valid certificate be included in incoming signed messages (see Section 1.1.1.4.3).

1.1.1.4.3 Certificate Verification Rules

AutoDRM requires that signed incoming emails contain a valid certificate. A certificate must meet these conditions to be considered valid:

- It follows X.509 PKI standards for version 3 certificates and for version 2 certificate revocation lists (CRLs) (See also [RFC 2459]).
- It is still active (that is, it must not have reached its expiry date, and must not be present on the CRL of the CA).
- It either matches a certificate already stored on the security token or is accessible through a certificate chain not to exceed a specified length.

  A certificate chain is an ordered list of certificates containing the certificate of the sender and some number of certificates through which a trusted certificate is eventually referenced. AutoDRM searches certificate chains when attempting to verify the authenticity of a message. The maximum length of the certificate chain is set by the configuration file *VerificationDepth* parameter (see Section 2.1.5 “[Authentication]” on page 12).

1.1.1.4.4 Certificate and CRL Caching

AutoDRM caches certificates and CRLs for 10 minutes. Any changes to these items made using SMConsole will not be seen by AutoDRM for that period of time unless AutoDRM is restarted.

If the CRL is updated by AutoDRM (as the result of an IMS2.0 command), AutoDRM will automatically clear the cache and use the new CRL.

1.1.1.4.5 Security Token

To verify request messages and to sign response messages, AutoDRM uses security services stored on a cryptographic token conforming to Public-Key Cryptography Standard #11 (PKCS11) of RSA Laboratories.

Depending on its intended function, a security token can be configured to generate and store public-private key pairs, to download and store trusted certificates and updated CRLs from CAs, and to store an access control list (or authorization model) as a data object. See the acquisition workstation installation instructions and the SMConsole manual for information on setting up and managing security tokens. [PKCS #11] provides an overview of cryptographic tokens and details on this standard, and the manufacturer can provide specifications for the Luna 2 PKCS11 token.

The number of key pairs on a token is restricted to 2. Therefore a successful `GENERATE_KEYPAIR` command will cause the inactive key pair to be replaced with the newly generated one.

1.1.1.5 Command Authorization

Following message authentication, AutoDRM evaluates whether the command request is authorized by checking the requester information against the authorization model that is stored on the workstation token. For example, with the default access permissions, a user with the assigned role of Operator is authorized to request `STOP_CONTINUOUS` but is not authorized to request `GENERATE_KEYPAIR`. (For more information, see the SMConsole manual.)

If the request is authorized AutoDRM will accept and process the request. If it is not authorized AutoDRM will reject the request; no response is sent to the requester. Future-dated command requests (`START_KEYPAIR` and `CALIBRATE_START`) will be reauthorized immediately before execution.

1.1.1.6 Rejected Requests

A request will be rejected if the message does not pass the authentication check, the requester is not valid, or if it is not reasonable to respond (for example, to a stale request). Information about the request and the requester is logged. An acknowledgement or explanatory message is sent to the requester under some circumstances (see Table 1-1).

Table 1-1 Acknowledgement of Rejected Requests

Reason for Rejected Request	Response Sent to the Requester
The requester cannot be verified.	No
The certificate chain contained in the digital signature is not valid.	No
The requester is not authorized to make the request.	No
The authenticated IMS2.0 request is stale. That is, it was created before the time period defined by the configuration file [Message] section *RequestExpiry* parameter.	Yes
The authenticated IMS2.0 request has a `MSG_ID` that is the same as that of at least one other message. (The cache will store `MSG_IDs` for as long as is defined in *RequestExpiry*, up to a maximum of 1000 values.)	Yes

1.2 Summary of Inputs and Outputs

To generate proper responses to request messages, AutoDRM requires access to various input files. These files must either be stored on the same computer as AutoDRM or be accessible over a LAN on a shared drive. Once AutoDRM has been started, it generates log and (optional) message copy files to directories as defined in the configuration file, and maintains the message number file.

1.2.1 Input Files

- `AutoDRM.ini` – This configuration file defines the operating characteristics for the AutoDRM program (see Chapter 2 “Configuring AutoDRM”).
- `Naqs.stn` – The Naqs station file `Naqs.stn` defines the station and channel configuration for the NaqsServer data acquisition program. AutoDRM uses this file to determine which channels are available for data requests, to locate the data ringbuffers and response files for those channels, and as an information source to respond to `STATION` and `CHANNEL` requests.
- `Naqsaddr.ini` – This file is created and managed by NaqsServer. It contains address information for each of the digitisers and must be present for AutoDRM to process digitiser command requests.
- `NmxToCD11.ini` – The configuration file for the NmxToCD11 subsystem must be present if waveform data requests are to support CSF formatting.
- `Calibration.ini` – The configuration file for the calibration subsystem must be present if calibration command requests are to be supported.

- Data files
  - Naqs ringbuffers – AutoDRM obtains waveform data and outage information from the NAQS ringbuffers.
  - Instrument response (*.rsp) files – AutoDRM obtains sensor and digitiser response data from the *.rsp files specified for each channel defined in the Naqs.stn file. (For information on generating response files with the Response and Tresponse utilities, see the Playback utilities manual.)
- *helpfilename*.txt – AutoDRM retrieves Help information from the station help file (for example, stnhelp.txt) if it has been installed. Edit this text file to reflect the station characteristics. The file name and path are defined in the [Control] section *HelpFile* parameter.

1.2.2 Output Files

- AutoDRM_*yyyymmdd*.log – The log files contain diagnostic messages generated by AutoDRM and provide a summary of the program operation. The logs record all request attempts and the identity of the requester. The log files are created daily, in the directory specified in the [Control] section of the configuration file.

  Each log message has an associated type, ranked by severity (Table 1-2). Log verbosity can be configured to show only messages at or above a specified severity level. The verbosity of the log on startup is set in the [Control] section of the AutoDRM.ini file. While AutoDRM is running, you can set verbosity to a different level by using the run-time commands (Section 3.2 on page 16).

Table 1-2 AutoDRM Log Message Types

Label	Description
F	Fatal errors – Serious errors which cause immediate system shutdown.
E	Errors – Abnormal occurrences which will likely affect data integrity.
W	Warnings – Less serious abnormal occurrences.
I	Informational messages – Messages tracing the normal operation of the system.
V	Verbose messages – Detailed informational messages tracing the normal operation of the system.
D	Debug messages – Additional verbose trace messages.

- REQ_*yyyymmdd_hh_mm_ss*.txt, MAIL_*yyyymmdd_hh_mm_ss*.txt (message copy files) – Optionally, you can set AutoDRM to save complete copies of incoming email request messages (REQ_*.txt) and outgoing email response messages (MAIL_*.txt). The message copies are logged in directories specified in the [Control] section of the configuration file.

Caution The output file MsgNum.txt is used by the AutoDRM program. Do not edit this file.

- MsgNum.txt – The message number file is used to ensure the continuity of the message ID even if the AutoDRM program is stopped and restarted. The number of the last message sent is stored in MsgNum.txt. If this file is lost, the message number is initialized to the configured default (defined in the [Message] section *MessageNumber* parameter). The MsgNum.txt file is stored in the working directory (typically /nmx/user).

- Scheduled items – The directory defined in the configuration file [Control] section *ScheduleDir* parameter will contain a single file for each outstanding item (for example, `CalibrationSched1.ser`). These files will be removed automatically by the system as the items are run.

1.3 Dependencies

Calibrations require NaqsServer and DataServer to be running.

1.4 Software Requirements

Before you install AutoDRM, ensure that the installation computer meets the following software requirements:

- Java Runtime Environment version 1.5 or later

1.5 Installing AutoDRM

AutoDRM must be installed either on the NAQSServer computer or on a computer that has TCP/IP access to the NAQSServer computer and network access to the NAQS ringbuffers. It requires a stand-alone mail server with a mailbox for AutoDRM.

Note Edit the file `nmx/user/stnhelp.txt` to reflect the characteristics of your network. (The content of this file is sent in response to a `HELP` request.)

- See the installation instructions for the acquisition system workstation.

1.6 Additional References

For more information, see the following external references:

[PKCS #11] RSA Laboratories (December 1999). “PKCS #11 v2.10: Cryptographic Token Interface Standard”. http://www.rsasecurity.com/rsalabs/node.asp?id=2133

[IDC 3.4.1] Science Applications International Corporation (SAIC), Pacific-Sierra Research Corporation. (Rev 1, March 1999). “Formats and Protocols for Messages - IMS 1.0”. http://www.rdss.info/

Internet Engineering Task Force (IETF) http://www.ietf.org/rfc.html for the following documents:

[RFC 1847] Galvin, J., Crocker, S., Freed, N., and S. Murphy, “Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted”, RFC 1847, October 1995.

[RFC 1848] Crocker, S., Freed, N., Galvin, J., and S. Murphy, “MIME Object Security Services”, RFC 1848, October 1995.

[RFC 2459] Housley, R., Ford, W., Polk, W. and D. Solo, “Internet X.509 Public Key Infrastructure Certificate and CRL Profile”, RFC 2459, January 1999.

[RFC 2632] Ramsdell, B., Editor, “S/MIME Version 3 Certificate Handling”, RFC 2632, June 1999.

[RFC 2633] Ramsdell, B., Editor, “S/MIME Version 3 Message Specification”, RFC 2633, June 1999.

Chapter 2
Configuring AutoDRM

Before running AutoDRM you must edit the configuration file (`AutoDRM.ini`) to provide network connection and file location information for the email server and receiver and to provide message structure and authentication parameters.

2.1 Definition of Configuration File Sections and Parameters

The `AutoDRM.ini` file contains five required sections and two optional sections for configuring the AutoDRM subsystems.

The parameters for the required sections are mandatory unless otherwise indicated. Each required section must appear once in the following order:

- [Interface]
- [Control]
- [Message]
- [Stations]
- [Authentication]

The two optional sections, if used, must be included at the bottom of the `AutoDRM.ini` file:

- [CalibrationDefaults]
- [CalibrationStationDefaults]

An example `AutoDRM.ini` file is shown in Appendix A.

- If you edit the `AutoDRM.ini` file, you must restart AutoDRM for the change to take effect.

2.1.1 [Interface]

The [Interface] section defines the mail server and mail box characteristics. It contains the parameters described in Table 2-1.

Table 2-1 [Interface] Section Parameters

Parameter	Definition
HostMailServer	The host server used to relay email (outbound using SMTP; inbound using the protocol defined in *MailBoxProtocol*). It is identified by either its host name, or its IP address expressed in dotted decimal format *w.x.y.z*. ◆ Permitted values: a valid mail server name or IP address.
MailDomain	The mail domain on the host mail server. ◆ Permitted values: a valid mail domain on the host mail server.
MailBoxProtocol	The protocol used to receive email messages. ◆ Permitted values: POP3 or IMAP4.
MailBox	The email recipient name for this program. AutoDRM will only accept email addressed to *MailBox@MailDomain* (for example, autodrm@*MailDomain*). ◆ Permitted values: the name, as an alphanumeric string, that has been assigned to the email recipient.
MailBoxPassword	The password required to access the AutoDRM recipient mail box. ◆ Permitted values: any alphanumeric string.

2.1.2 [Control]

The [Control] section defines the locations and characteristics of various input and output files. It contains the parameters described in Table 2-2.

Table 2-2 [Control] Section Parameters*

Parameter	Definition
StationFile	The path and name of the NAQS station file `Naqs.stn`.
AddressFile	The path and name of the Naqs station address mapping file `naqsaddr.ini`.
CalibrationFile	The path and name of the Calibrate configuration file.
CD11File	The path and name of the NmxToCD11 configuration file.
HelpFile	The path and name for the help file to be sent in response to a `HELP` request (typically stnhelp.txt). ▸ Edit the help file to provide accurate information about your installation.
StartContScript	The file path and name of the `set_primary` script, which will start continuous data transmission when run.
StopContScript	The file path and name of the `set_auxiliary` script, which will stop continuous data transmission when run.
RequestCache	The file path and name for the cache for future-dated request information.
AutoDRMLogDir	The path for the AutoDRM log.
RequestLogDir	The path for the request messages log directory.
ResponseLogDir	The path for the response messages log directory.
ScheduleDir	The path for the scheduled items directory.

Table 2-2 [Control] Section Parameters* (Continued)

Parameter	Definition
AcceptUnsigned	This indicates whether or not unsigned data requests should be accepted. It might be useful to accept unsigned requests on installations that do not have any security concerns to eliminate the need for all users to have mail signing capability. Signed requests will still be verified. ◆ Permitted values: Yes, No.
Verbosity	The severity level of the log messages displayed at startup. You can change the setting for a particular session (see Section 3.2 “Using the Run-Time Commands” on page 16). ◆ Permitted values: DEBUG, VERBOSE, and INFO.

* Paths and file names must not contain spaces. Locations must be accessible locally or through a TCP/IP network connection. Optionally, use a period (.) for the current directory.

2.1.3 [Message]

The [Message] section defines response message characteristics. It contains the parameters described in Table 2-3.

Table 2-3 [Message] Section Parameters

Parameter	Definition
MessageIdPrefix	Each response message is identified by a message ID (in the `MSG_ID` line) in the form *Prefix-Number. MessageIdPrefix* defines the prefix to be used in the value for the message ID. ◆ Permitted values: any alphanumeric string without spaces.
MessageNumber	Each response message is identified by message ID (in the `MSG_ID` line) in the form *Prefix-Number. MessageNumber* defines the default initial response message number. The message number is incremented for each response sent. AutoDRM maintains a file MsgNum.txt containing the current message number (Section 1.2.2 on page 6), and normally uses the value from that file. The MessageNumber value is used only if MsgNum.txt cannot be read. ◆ Permitted values: any positive integer.
ResponseSizeLimit	The maximum size in megabytes of an email response message. Responses larger than this limit will be sent as numbered parts in multiple email messages. ◆ Permitted values: any number from 0.4 to 2. Recommended maximum is 1. ([IDC 3.4.1] page 4 recommends a maximum email message size of 1 Mbyte).
RequestExpiry	The maximum age of a request (in hours) before it is considered stale. A stale request is rejected (For more information, see Section 1.1.1.6 “Rejected Requests” on page 5). ◆ Permitted values: any positive integer.

2.1.4 [Stations]

The [Stations] section defines the station information. It contains the parameters described in Table 2-4.

Table 2-4 [Stations] Section Parameters

Parameter	Definition
StationCode	The array station name for this instance of AutoDRM. You can specify this name in the `STA_LIST` environment line to obtain information for all stations. ◆ Permitted values: any valid station code of 3 to 5 characters. (See also [IDC 3.4.1]).
CFSiteName	The Central Facility site name (for example, CF-01-01). If this parameter is missing from the file, AutoDRM will use CF-01-01 by default.
StartDate	The date the station started operating. This is used to define the Start Date field in Station and Channel data. ◆ Permitted values: a date in format *yyyy/mm/dd*.
EndDate	The projected date on which the station will be decommissioned. ◆ Permitted values: a date in format *yyyy/mm/dd*.

2.1.5 [Authentication]

The [Authentication] section defines token access characteristics. It contains the parameters described in Table 2-5. (For more information, see the SMConsole manual.)

Table 2-5 [Authentication] Section Parameters*

Parameter	Definition
TokenID	The serial number of the PKCS11 token. ◆ Permitted values: the serial number, or the word `any`. • If `TokenID = any` AutoDRM will use the first token it finds; it will not search all possible tokens.
PIN	The user password needed to log in to the PKCS11 token user account (For more information, see the SMConsole manual). ◆ Permitted values: a valid PIN
VerificationDepth	The maximum permitted length of the chain of certificates that will be searched in order to authenticate the message. In all cases, the signing certificate must chain through to a self-signed CA certificate stored on the token. ◆ Permitted values: the recommended value is 0, where: • 0 – The certificate used to sign the email must be stored on the token. • 1 – The certificate of the CA that issued the certificate used to sign the email must be stored on the token. • 2 – The certificate used to sign the email can chain through any number of issuing local CAs, which must be included in the email or be present on the token, as long as the final global CA in the chain is stored on the token.

* The parameter *KeyID* has been removed starting with version 2.01. AutoDRM will ignore it if it is included in the configuration file.

2.1.6 [CalibrationDefaults]

The [CalibrationDefaults] section defines the default calibration settings for all stations. This section is optional and each parameter is also optional. This section contains the parameters and default settings described in Table 2-6.

Note For more information on calibration settings, see the Calibrate user guide.

Table 2-6 [CalibrationDefaults] Section Parameters (Optional)

Parameter	Default Setting	Definition
Ton	*30*	The number of seconds after the calibration coil is engaged to wait before starting calibration signal. ◆ Permitted values: any positive integer > = 0
SwRamp	*30*	The number of seconds for the sine wave to ramp up to full amplitude signal. The number of seconds for the sine wave to ramp down from full amplitude signal. ◆ Permitted values: any positive integer > = 0
SwNfft	*100*	The number of samples used in each fast Fourier transform window during analysis. ◆ Permitted values: any positive integer > = 32 • To optimize efficiency and accuracy, you should specify a number with many small prime factors.
SwDec	*1*	The factor by which the digitizer sample rate is decimated before analysis. ◆ Permitted values: 1,2,3,4,5,10,20,25,30,40,50,100
PrbUnitWidth	*2.0*	The PRB signal unit width in seconds. ◆ Permitted values: any positive value > = 0
PrbNfft	*100*	The number of samples used in each fast Fourier transform window during analysis. ◆ Permitted values: any positive integer > = 32 • To optimize efficiency and accuracy, you should specify a number with many small prime factors.
PrbDec	*1*	The factor by which the digitizer sample rate is decimated before analysis. ◆ Permitted values: 1,2,3,4,5,10,20,25,30,40,50,100
PulseDelay	*60*	The number of seconds after the pulse ends to include in analysis. ◆ Permitted values: any positive integer > = 0
PulseNfft	*100*	The number of samples used in each fast Fourier transform window during analysis. ◆ Permitted values: any positive integer > = 32 • To optimize efficiency and accuracy, you should specify a number with many small prime factors.
PulseDec	*1*	The factor by which the digitizer sample rate is decimated before analysis. ◆ Permitted values: 1,2,3,4,5,10,20,25,30,40,50,100

2.1.7 [CalibrationStationDefaults]

The [CalibrationStationDefaults] section defines the default calibration settings for a specific station. Use one section per station for an unlimited number of stations. This section is optional and only the StationName parameter is required: The rest of the parameters are optional. This section contains the parameters and default settings described in Table 2-7.

Note For more information on calibration settings, see the Calibrate user guide.

Table 2-7 [CalibrationStationDefaults] Section Parameters (Optional)

Parameter	Default Setting	Definition
StationName	*STN01*	The station name as defined in the [Station] section of the `Naqs.stn` file. ◆ Permitted values: any valid station code of 3 to 5 characters.
Ton	*300*	The number of seconds after the calibration coil is engaged to wait before starting calibration signal. ◆ Permitted values: any positive integer > = 0
SwRamp	*240*	The number of seconds for the sine wave to ramp up to full amplitude signal. The number of seconds for the sine wave to ramp down from full amplitude signal. ◆ Permitted values: any positive integer > = 0
SwNfft	*100*	The number of samples used in each fast Fourier transform window during analysis. ◆ Permitted values: any positive integer > = 32 • To optimize efficiency and accuracy, you should specify a number with many small prime factors.
SwDec	*1*	The factor by which the digitizer sample rate is decimated before analysis. ◆ Permitted values: 1,2,3,4,5,10,20,25,30,40,50,100
PrbUnitWidth	*2.0*	The PRB signal unit width in seconds. ◆ Permitted values: any positive value > = 0
PrbNfft	*100*	The number of samples used in each fast Fourier transform window during analysis. ◆ Permitted values: any positive integer > = 32 • To optimize efficiency and accuracy, you should specify a number with many small prime factors.
PrbDec	*1*	The factor by which the digitizer sample rate is decimated before analysis. ◆ Permitted values: 1,2,3,4,5,10,20,25,30,40,50,100
PulseDelay	*60*	The number of seconds after the pulse ends to include in analysis. ◆ Permitted values: any positive integer > = 0
PulseNfft	*100*	The number of samples used in each fast Fourier transform window during analysis. ◆ Permitted values: any positive integer > = 32 • To optimize efficiency and accuracy, you should specify a number with many small prime factors.
PulseDec	*1*	The factor by which the digitizer sample rate is decimated before analysis. ◆ Permitted values: 1,2,3,4,5,10,20,25,30,40,50,100

Chapter 3
Running AutoDRM

This chapter provides information on starting and stopping AutoDRM, using the run-time commands, and monitoring the operation of AutoDRM.

3.1 Starting and Stopping AutoDRM

You can start and stop AutoDRM locally or remotely via a telnet session.

3.1.1 Starting and Stopping AutoDRM Locally

In a typical network AutoDRM will be set up to start automatically using scripts (on Solaris and Linux) or the NmxWatchdog utility (on Windows). It can also be started manually from the command line. AutoDRM must be shut down properly to release its system resources.

- To start AutoDRM manually from the command line, type `autodrmconsole` in any terminal window.
- To stop AutoDRM, type `stop` or `quit` in the AutoDRM terminal window.
- To set AutoDRM to be started and monitored automatically on Windows by the NmxWatchdog utility, add the following entry to the `watchdog.ini` file:

```
[ WatchEntry n ]
ProgramTitle = AutoDRM
ProgramPathname = "java -cp c:\nmx\bin\AutoDRM.jar
ca.nanometrics.autodrm.AutoDRM"
WorkingDirectory = "c:\nmx\user"
ExitAction = Restart
PingsSemaphore = true
StartDelay = 6s
```

3.1.2 Stopping and Restarting AutoDRM Remotely

You can stop and restart AutoDRM remotely via a telnet session. This allows you to change the `.ini` file and validate the changes by restarting AutoDRM.

On Linux and Solaris:

- To stop AutoDRM remotely, type `Nmxkill /nmx/user/AutoDRM`
- To restart AutoDRM remotely, type `Nmxrestart /nmx/user/AutoDRM`

On Windows:

- To stop AutoDRM remotely, type `Nmxkill c:\nmx\user\AutoDRM`
- To restart AutoDRM remotely, type `Nmxrestart c:\nmx\user\AutoDRM`

3.2 Using the Run-Time Commands

AutoDRM supports a basic keyboard interface for entering run-time commands, with the options described in Table 3-1. The commands are case-insensitive.

Table 3-1 AutoDRM Run-Time Commands

To...	Type...
Display all log messages in the log file; set the log verbosity to DEBUG	`D`
Suppress debug messages in the log file; set the log verbosity to VERBOSE	`V`
Suppress debug and verbose messages in the log file; set the log verbosity to INFO	`I`
Toggle the mail debugger option to show/hide connection protocols.	`M`
Toggle to save (or not save) a copy of all request and response messages to file. These files are stored in the directory set in [Control] section *R*LogDir* parameters.	`F`

Stop AutoDRM and exit. Exit AutoDRM using either of these commands; both ensure that all files are closed and system resources are released.	`QUIT`
	or
	`STOP`

3.3 Updating the Token Configuration

AutoDRM caches certificates and CRLs for 10 minutes, and access control lists are cached indefinitely. Any changes to these items made using SMConsole will not be seen by AutoDRM for that period of time unless AutoDRM is restarted.

- Restart AutoDRM after making changes to certificates, CRLs, or the access control list (authorization model) to ensure that the cache reflects the current token configuration.

3.4 Monitoring the Operation of AutoDRM

AutoDRM generates log messages that trace the operation of the program. It displays these messages in the terminal window and writes them to the current `AutoDRM .log` file. You can set the level of detail (the verbosity) of the information to be displayed and recorded.

- To view the log, open the file `AutoDRM_`*date*`.log` in a text editor. The log file location is set with the configuration file [Interface] section *AutoDRMLogDir* parameter.
- To set the verbosity of log messages on startup, edit the [Control] section of the `AutoDRM.ini` configuration file.
- To change the verbosity of log messages while AutoDRM is running, use the run-time commands.

3.5 Troubleshooting AutoDRM

To set up AutoDRM for troubleshooting:

1. Set the log verbosity to Verbose (`V`), to provide a detailed trace of AutoDRM activity.
2. Toggle the mail debugger option (`M`) to show/hide connection protocols.
3. Toggle on the save-to-file option (`F`). With this option enabled, AutoDRM will save complete copies of all incoming requests to timestamped files in the configured directories. These files contain complete Internet headers and can be examined with any text editor to aid in solving mail problems.

For more information, see Section 3.2 “Using the Run-Time Commands” on page 16.

3.5.1 AutoDRM Is Not Receiving Email

If AutoDRM is not receiving email and there is no other message on the console window:

- Check that the mail protocol is correct (this is set with the [Interface] section *MailBoxProtocol* parameter), and that the POP3 or IMAP4 service is running at the mail host.

3.5.2 AutoDRM Cannot Verify Email

If AutoDRM rejects a signed email because it cannot be verified, it will print a log message indicating why the verification failed.

If Mail is rejected because a valid certificate is not stored on the security token:

- Install the necessary certificates on the token using the SMConsole program, and ensure that the [Authentication] section *VerificationDepth* parameter is set to an appropriate value.

3.5.3 AutoDRM Cannot Find Data

If AutoDRM cannot find data

- Ensure that the path name in the `Naqs.stn` file matches the path and name of the ringbuffer file.

3.5.4 AutoDRM Sends an Email but It Is Not Received

If AutoDRM is sending email and there is no error message on the console window, ensure that the [Interface] section *HostMailServer* parameter is specified properly:

- Check that the server name or IP address is correct.

3.5.5 AutoDRM Rejects Requests

If AutoDRM rejects requests, ensure that the request message is formatted correctly, and note request rejection conditions (For more information, see Section 1.1.1.6 “Rejected Requests” on page 5):

- Command requests must be specified as IMS2.0. (Data requests can be sent as either IMS1.0 or IMS2.0, but IMS2.0 requests will have the `MSG_ID` checked against previous ones; the request will then be parsed as IMS1.0.)
- For IMS2.0 command requests, the `TIME_STAMP` and `MSG_ID` lines are checked.
- The system will check the requester against the authorization model (or access control list) using the Subject Distinguished Name for the certificate. There must be a user set up with this name in the authorization model using SMConsole. Unsigned requests use the unsigned *userid* in the authorization model. (For more information, see the SMConsole manual.)

Chapter 4
Using AutoDRM

AutoDRM accepts data and command requests from authorized senders in the form of properly addressed email messages using S/MIME multipart/signed format (For more information, see Section 1.1.1.3 on page 2).

- To request data from or submit command requests to a Nanometrics data acquisition system through AutoDRM, send a digitally-signed, properly structured email request message to the address of the server of the system you wish to query.

This chapter provides a summary of AutoDRM message structure and message line descriptions. For more information, see Appendix B "Message Syntax".

4.1 Message Types

AutoDRM Version 2.1 supports the following message types:

- Request messages:
  - Data request
  - Help (a special case of data request that does not require standard message lines)
  - Command request
- Response messages:
  - Data
  - Command response

4.2 Message Structure Overview

- More than one request message (that is, an IMS request message bounded by `BEGIN` and `STOP`) and more than one message type can be included in an email message.
- One message type can be included in a request message.
- One command request can be included in a request message.
- More than one data request can be included in a request message.
- Message lines must be left-justified.
- Message lines are not case-sensitive.

All messages (except for HELP requests) require standard messages lines to format the message and specify the request or response. These include:

- Message format lines
  - BEGIN – start of the message
  - MSG_TYPE – the type of message (for example, request, data, command request, command response)
  - MSG_ID – a two-part identifier, user-defined for request messages and AutoDRM-constructed for response messages
  - REF_ID – used in response messages to identify the original request message, using the value of MSG_ID from the request message
  - STOP – end of the message
- Request control line – used in request messages to specify the response protocol as email
- Request environment lines – specific by message type, these define parameters of the request (for example, stations and timeframes)
- Request lines – to select one or more of the supported data types or to submit a command

4.2.1 Request Message Structure

All messages (except for HELP requests) must have BEGIN as the first line of the message body. This line is used to identify the message format version (for example, IMS1.0). HELP constitutes a complete request message and does not require any other message lines.

Lines two and three must be MSG_TYPE and MSG_ID respectively. These lines identify the type of message and the string used to identify the original message in the response. The message identification MSG_ID contains a user-assigned ID_string code and, optionally, a source code. These are separated by a space.

The fourth line should be the request control line E-MAIL, to identify the response protocol as email and to specify the destination email address for the response. This line is optional; if it is not included, AutoDRM uses the email header information to determine the address for the response.

The fifth through *n*th lines are environment and request lines specific to the message type (see Table 4-1 “Data Request Message Lines” on page 23 and Table 4-3 “Command Request Message Lines” on page 24).

- Environment lines define parameters of the request (for example, defining the source of requested data or the time at which to implement a calibration command). Precision for time environment definitions is as defined in [IDC 3.4.1] (For more information, see the message examples in Appendix B)
- Request lines specify the type of data or command request (for example, WAVEFORM, UPDATE_CRL)

Some command requests include an enclosure between the request and STOP (for example, a certificate request).

The last line of the message must be STOP.

AutoDRM ignores any information in a message following the first instance of STOP, so all requests must be bounded by one set of BEGIN and STOP commands.

4.2.1.1 Request Message Example

```
From: sender@senderdomain
Message-ID: message ID created automatically to track logged copies
To: autodrm@autodrmdomain
Subject: data request format example
MIME-Version: 1.0
Content-Type: multipart/signed; protocol="application/x-pkcs7-signature";
micalg=sha1; boundary="----145CF85120FDED29F5B4B42AC8DF7032"

This is an S/MIME signed message

------145CF85120FDED29F5B4B42AC8DF7032
Content-Type: text/plain

begin ims1.0
msg_type request
msg_id Example_1
e-mail datarecipient@recipientdomain
time 2008/5/12 00:03:00.000 to 2008/5/12 00:15
sta_list *z
waveform ims1.0 : cm6
outage
stop
------145CF85120FDED29F5B4B42AC8DF7032
Content-Type: application/x-pkcs7-signature; name="smime.p7s"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7s"

signature

------145CF85120FDED29F5B4B42AC8DF7032--
```

4.2.2 Response Message Structure

Response messages follow essentially the same format as request messages. The `MSG_TYPE` typically is either data or a command response. Additional response message lines include `REF_ID` (a reference number identifying the original request message), and `DATA_TYPE` for a data request (identifying the type of data listed in the section of the message immediately following). Formats for the data output comply with the specifications in [IDC 3.4.1].

4.2.2.1 Response Message Example

```
Message-ID: message ID created automatically to track logged copies
Date: Tue, 15 April 2008 11:38:10 -0400 (EDT)
From: autodrm@autodrmdomain
To: datarecipient@recipientdomain

Mime-Version: 1.0
Content-Type: multipart/signed; protocol="application/x-pkcs7-signature";
micalg=sha1;
boundary="----=_Part_5_3086625.1027956676305"

------=_Part_5_3086625.1027956676305
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit

begin ims1.0
msg_type data
msg_id Response_Ex_1
ref_id Example_1
```

```
data_type LOG IMS1.0
 begin ims1.0
 msg_type request
 msg_id Example_1
 e-mail your email address
 time 2008/05/12 00:03:00.000 to 2008/5/12 00:15
 sta_list *z
 waveform ims1.0 : cm6
 outage
 stop

DATA_TYPE WAVEFORM IMS1.0:CM6
compressed waveform data

DATA_TYPE OUTAGE IMS1.0
outage data
stop

------=_Part_5_3086625.1027956676305
Content-Type: application/x-pkcs7-signature; name="smime.p7s"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7s"
Content-Description: S/MIME Cryptographic Signature

signature
------=_Part_5_3086625.1027956676305--
```

4.3 Message Line Descriptions

This section includes message line descriptions for supported message types. See Appendix B for message syntax and examples.

4.3.1 Help Request

A help request retrieves the content of the station *HelpFile*. It does not require the basic message lines used for other types of requests, just the word `help` in the message body. Help request messages do not have to be signed.

For example:

```
From: your email address
To: AutoDRM email address
Subject:
---------------------message boundary
help
```

4.3.2 Data Request

Table 4-1 describes format, control, environment, and request lines for data request messages.

Table 4-1 Data Request Message Lines

Line label	Parameters	Description	Line type
BEGIN	ims1.0 | ims2.0	Required. Indicates the start of message and the message format version number. (If the version number is IMS2.0, the message MSG_ID is checked against previous ones for duplication.)	Format
MSG_TYPE	request	Required. Indicates the type of message.	Format
MSG_ID	*ID_string [source]*	Required. The user-defined two-part identifier for the message. The two parts are separated by a space. It must include a unique *ID_string*, up to 20 alphanumeric characters, and optionally can include a code identifying the message source, up to 16 alphanumeric characters (for example, the station network code).	Format
E-MAIL	*[address]*	Optional. Specifies the response protocol as email, and the destination address. If an address is not specified, the default is the sender address in the email message header.	Control
TIME	*[date time]*	Required. Defines the period for which data are requested. Not all fields need to be completed. The default is the time the request message is sent. Format is *yyyy/mm/dd hh:mm:ss.sss* to *yyyy/mm/dd hh:mm:ss.sss*	Environment
STA_LIST	*[station code]*	Optional. Defines the stations within the network for which data are requested. The wildcard character is an asterisk (*). The default is all stations in the network.	Environment
CHAN_LIST	*[channel code]*	Optional. Defines the channels for which data are requested; these are from the STA_LIST stations. The wildcard character is an asterisk (*). The default is the vertical channel for the defined stations: *z.	Environment
STATION	ims1.0 | ims2.0*	Retrieves station information for one or more stations.	Request
CHANNEL	ims1.0 | ims2.0*	Retrieves channel information for one or more data channels.	Request
WAVEFORM	ims1.0 | ims2.0 [: *compression*]	Retrieves waveform data for one or more channels. Supported data compression formats are uncompressed format INT (the default), CM6, and CSF for authenticated data in CD1.1 format.	Request
RESPONSE	ims1.0 | ims2.0*	Retrieves response characteristics for one or more data channels.	Request

Table 4-1 Data Request Message Lines (Continued)

Line label	Parameters	Description	Line type
OUTAGE	ims1.0 | ims2.0[*]	Retrieves information about data outages for one or more channels.	Request
STOP		Required. Indicates the end of the message.	Format

[*] The IMS version must be the same as that used in the BEGIN Format line.

4.3.3 Data Response

Table 4-2 describes format and data type lines for data response messages.

Table 4-2 Data Response Message Lines

Line label	Parameters	Description	Line type
BEGIN	ims1.0 | ims2.0	Indicates the start of message and the message format version number.	Format
MSG_TYPE	data	Indicates the type of message.	Format
MSG_ID	*MessageIdPrefix-MessageIdNumber*	AutoDRM assigns the two-part identifier for the message.	Format
REF_ID	*Request_Msg_ID [part sequence information]*	Identifies the original request message. For multipart responses it includes part sequence information.	Format
DATA_TYPE	*requested type* ims1.0 | ims2.0	Identifies the type of data included in the message, followed by the data.	Data type
STOP		Indicates the end of the message.	Format

4.3.4 Command Request

Table 4-3 describes format, control, environment, and request lines for command request messages.

Note Timestamp (TIME_STAMP) and message ID (MSG_ID) values are checked for IMS2.0 command request messages, and MSG_ID is checked for data request messages specified as IMS2.0. Values that are invalid will cause AutoDRM to reject the request (For more information, see Section 1.1.1.6 on page 5).

Table 4-3 Command Request Message Lines

Line label	Parameters	Description	Line type
BEGIN	ims2.0	Required. Indicates the start of message and the message format version number. (If the version number is IMS1.0, the command request is rejected.)	Format
MSG_TYPE	command_request	Required. Indicates the type of message.	Format

Table 4-3 Command Request Message Lines (Continued)

Line label	Parameters	Description	Line type
`MSG_ID`	*ID_string [source]*	Required. The user-defined two-part identifier for the message. The two parts are separated by a space. It must include a unique *ID_string*, up to 20 alphanumeric characters, and optionally can include a *source* code identifying the message source, up to 16 alphanumeric characters (for example, the station network code).	Format
`E-MAIL`	*[address]*	Optional. Specifies the response protocol as email, and the destination address. If an address is not specified, the default is the sender address in the email message header.	Control
`TIME_STAMP`	*date [time]*	Required. Assigns a date and time when the command request is issued. A blank value will cause the request to be rejected. Format is *yyyy/mm/dd hh:mm:ss.sss*	Environment
`STA_LIST`	*[station code]*	Optional. Defines the stations (instruments and central facility workstation) within the network for which the command is requested. The wildcard character is an asterisk (`*`). The default is all stations in the network, including the central facility where applicable.	Environment
`CHAN_LIST`	*[channel code]*	Optional. Defines the channels for which the command is requested; these are from the `STA_LIST` stations. The wildcard character is an asterisk (`*`). The default is the vertical channel for the defined stations: `*z`.	Environment
`AUTH_ID`	*active_keypair_ID*	Optional for `GENERATE_KEYPAIR` and `START_KEYPAIR`. `AUTH_ID` either specifies an authentication unit by its active key pair ID or redefines the currently active key pair ID, depending on the command request: ◆ For `GENERATE_KEYPAIR`, it specifies the authentication unit on which to run the command. When this line is included, the command will reject updates to any authentication units with an active key pair ID not matching the specified ID. ◆ For `START_KEYPAIR`, it defines the new key ID to assign to the active key pair. When this line is not included, the newly started key pair will retain the ID of the previously active key pair.	Environment
`DURATION`	*seconds*	Required for `CENTER_MASS`. The duration of the mass centring signal.	Environment

Table 4-3 Command Request Message Lines (Continued)

Line label	Parameters	Description	Line type
`START_TIME`	*date [time]*	Required for `CALIBRATE_START` and optional for `START_KEYPAIR`. The default for `START_KEYPAIR` is the time the request is received. Format is *yyyy/mm/dd hh:mm:ss*	Environment
`SENSOR`	`yes | no`	Required for `CALIBRATE_START`. Indicates whether the sensor should be included in the calibration.	Environment
`TYPE`	`pulse | random | sine`	Required for `CALIBRATE_START`. The calibration signal will either be sine wave, random (pseudo random binary), or pulse signal.	Environment
`CALIB_PARAM`	*parameters*	1 to *N* required for `CALIBRATE_START`. The required parameters for the three different signal types are as follows: ◆ pulse • first signal pulse: duration • subsequent signal pulses: duration, delay ◆ random – duration ◆ sine – frequency, duration The Amplitude parameter is optional for all three signal types.	Environment
`UPDATE_CRL`		Update the CRL stored at the CPCSS. Include the updated CRL.	Request
`GENERATE_KEYPAIR`		Generate a new key pair on digitiser or central facility workstation authentication units. Optionally, include the DSA parameters for the key pair.	Request
`START_KEYPAIR`		Start using the specified key pair. Include the certificate in the request.	Request
`CALIBRATE_START`		Calibrate one or more channels. The START_TIME can be any valid date that is future-dated a maximum of 31 days or post-dated a maximum of 1 hour. For more information, see the Calibrate user guide. Requires environment variables `START_TIME, SENSOR, TYPE`. ✧ Only one calibration can be running at any one time. Any calibrations scheduled to start while another calibration is running will be rejected.	Request
`CENTER_MASS`		Centre mass on one or more elements. Requires environment variable `DURATION`.	Request
`START_CONTINUOUS`		Turn on primary station continuous data transmission.	Request
`STOP_CONTINUOUS`		Turn off primary station continuous data transmission.	Request
`STOP`		Indicates the end of the message.	Format

4.3.5 Command Response

Table 4-4 describes format, environment, and response lines for command response messages.

Table 4-4 Command Response Message Lines

Line label	Parameters	Description	Line type
BEGIN	ims2.0	Indicates the start of message and the message format version number.	Format
MSG_TYPE	command_response	Indicates the type of message.	Format
MSG_ID	*MessageIdPrefix-MessageIdNumber*	AutoDRM assigns the two-part identifier for the message.	Format
REF_ID	*Request_Msg_ID*	Identifies the original request message.	Format
TIME_STAMP	*date time*	Assigns a date and time when the command response is issued.	Environment
STA_LIST	*[station code]*	Defines the stations within the network to which the response applies. The wildcard character is an asterisk (*).	Environment
CHAN_LIST	*[channel code]*	Defines the channels to which the response applies; these are from the STA_LIST stations. The wildcard character is an asterisk (*).	Environment
AUTH_ID	*keypairID*	Defines the authentication unit identified in the request.	Environment
CRL_UPDATED		Response to UPDATE_CRL request. It confirms that the CRL is updated.	Response
KEYPAIR_GENERATED		Response to GENERATE_KEYPAIR request. It confirms that a key pair has been generated, and includes a certificate request for the new key pair.	Response
KEYPAIR_STARTED		Response to START_KEYPAIR request. It indicates when the key pair was started, and includes the certificate.	Response
KEYPAIR_CONFIRM		First response to future-dated START_KEYPAIR requests.	Response
CALIBRATE_CONFIRM		First response to CALIBRATE_START requests.	Response
START_TIME	*date time* | not confirmed	Indicates either the actual time for which the action has been scheduled, or not confirmed if the requested action cannot be scheduled.	Response
PROBLEM_ENCOUNTERED	*description*	Response to a command request that did not execute as expected.	Response
COMMAND	*name of command*	Part of an error response describing what command failed.	Response
CALIBRATE_RESULT		Second response to a successful CALIBRATE_START request, it includes the calibration result (IN_SPEC, and optionally CALIB, CALPER).	Response
CALIB	*value*	Calibration factor.	Response

Table 4-4 Command Response Message Lines (Continued)

Line label	Parameters	Description	Line type
`CALPER`	*value*	Calibration period.	Response
`IN_SPEC`	`yes | no`	Indicates whether the channel response is within specifications.	Response
`MASS_CENTERED`		Response to `CENTER_MASS` request.	Response
`CONTINUOUS_STARTED`		Response to `START_CONTINUOUS` request.	Response
`CONTINUOUS_STOPPED`		Response to `STOP_CONTINUOUS` request.	Response
`STOP`		Indicates the end of the message.	Format

Appendix A
Configuration File Example

A.1 General Structure

The AutoDRM configuration file is structured as an .ini file, a format which is designed to be readable and editable in any text editor.

These files consist of a number of sections, each containing several parameters:

- Sections are identified by a name enclosed in square brackets (for example, [Interface]).
- Each parameter is defined on a separate line following the section identifier. The format for defining the parameter is as follows: *ParameterName* = *Value*.

The following example shows the section that defines the network connections for AutoDRM:

```
[Interface]

HostMailServer = 199.71.138.13       // name or IP address for host email server
MailDomain = test.nanometrics.ca     // mail domain
MailBoxProtocol = pop3               // protocol for connection to mail box
MailBox = autodrm                    // name of the mail box on the HostMailServer
MailBoxPassword = autodrm            // password for mail box login
```

A.1.1 Data Order and Default Values

All parameters for a given section must be defined after the section identifier for that section and before any other section identifier.

AutoDRM does not provide any default settings for the parameters in the required sections. Therefore, you must fully define every parameter in each required section. Parameters must be defined in the order that they are listed in the parameter description sections (For more information, see Chapter 2).

AutoDRM does provide default settings for the parameters in the optional sections. Therefore, you do not have to define any of the parameters in those optional sections if you want to accept the default values. Parameters must be defined in the order that they are listed in the parameter description sections (For more information, see Chapter 2).

A.1.2 White Space and Comments

The inifile reader ignores white space and blank lines, so you can add white space anywhere within a configuration file to improve readability.

The double slash // is a comment delimiter. You can add comments anywhere in a file, to add descriptive information and to remove parameters or sections temporarily from the file.

For example:

```
// This is a full line comment
[Station]                          // a comment can follow a section header
StationCode = CA01                // a comment can follow a parameter definition
```

A.1.3 Configuration File Error Detection

AutoDRM parses the configuration files on startup. If it detects any errors (unrecognized fields or illegal values), it will print an error message and stop. Illegal values are values which are undefined or out of range for a particular parameter (For more information, see Chapter 2).

- To resume, fix the file using a text editor, and then restart AutoDRM.

The most common cause of unrecognized fields are as follows:

- Misspelled parameter names – Check the spelling carefully, and note that parameter names are case-sensitive.
- Missing names – If a parameter appears out of order or in the wrong section, it will not be recognized.
- Duplicated names – If a parameter name appears more than once, only the first instance will be recognized.

A.2 Example AutoDRM Configuration File

```
[Interface]

HostMailServer = localhost           // name or IP address for host email server
MailDomain = ws01-ca01.gci.ctbto.org // mail domain
MailBoxProtocol = pop3               // protocol for connection to mail box
MailBox = autodrm                    // name of the mail box on the HostMailServer
MailBoxPassword = autodrm            // password for mail box login

[Control]

StationFile = /nmx/user/Naqs.stn             // Naqs station file
AddressFile = /nmx/user/naqsaddr.ini         // Naqs station address mapping file
CalibrationFile = /nmx/user/autodrm/Calibration.ini  // Calibration configuration 
file ** DO NOT MODIFY **
CD11File = /nmx/user/NmxToCD11.ini           // CD1.1 configuration file
HelpFile = /nmx/user/autodrm.hlp             // AutoDRM help file
StartContScript = /nmx/bin/set_primary       // Script to start continuous data 
transmission
StopContScript = /nmx/bin/set_auxiliary      // Script to stop continuous data 
transmission
RequestCache = /nmx/user/AutodrmRequests.ser // Cache for future-dated request 
information
AutoDRMLogDir = /nmx/log/AutoDRMLogs        // AutoDRM log path
RequestLogDir = /nmx/log/AutoDRMLogs/AutoDRMRequests    // Request log path, '.' for 
current directory
```

```
ResponseLogDir = /nmx/log/AutoDRMLogs/AutoDRMResponses  // Response log path, '.' for
current directory
ScheduleDir = /nmx/user/autodrm/scheduleItems         // Directory for storing
schedule items

** DO NOT MODIFY **

AcceptUnsigned = Yes                         // accept unsigned requests
Verbosity = VERBOSE                           // DEBUG,  VERBOSE or INFO

[Message]

MessageIdPrefix = WS01-CA01   // the prefix of messageID, string without space
MessageNumber = 0           // Initial message number as the suffix of messageID, int
ResponseSizeLimit = 1.0      // response message size limit in Mbytes if specified
size is > 2MB or < 0.4 MB
RequestExpiry = 24          // maximum age of a request before it is considered stale
in hours

[Stations]

StationCode = CA01             // array station code for this AutoDRM
StartDate = 1970/01/01        // start date for all stations
EndDate = 2050/01/01          // end date for all stations

[Authentication]

TokenID = any                // tokenID for the token you use ("any" = reads s/n from
token)
PIN = CTBTO                 // the pin code to login token
KeyID = 1                   // the ID to find the private key for signing/
authentication
VerificationDepth = 1         // number of chained certificates received in E-mail
enough to verify the message

[CalibrationDefaults]// optional section; each parameter is also optional

Ton = 30   // seconds after calibration coil is engaged to wait to start calibration
SwRamp = 30       // seconds to ramp up and down to full amplitude signal duration
SwNfft = 100     // size of nFFT window in samples
SwDec= 1         // amount to decimate signal (allowed values:
1,2,3,4,5,10,20,25,30,40,50,100)
               // decimation will default to nearest value if not one of the above
PrbUnitWidth = 2.0       // seconds of minimum width
PrbNfft = 100            // size of nFFT window in samples
PrbDec = 1              // amount to decimate signal (allowed values:
1,2,3,4,5,10,20,25,30,40,50,100)

PulseDelay = 60        // seconds after pulse to include in analysis
PulseNfft = 100        // size of nFFT window in samples
PulseDec = 1           // amount to decimate signal (allowed values:
1,2,3,4,5,10,20,25,30,40,50,100)
```

```
[CalibrationStationDefaults]  // station specific defaults;  1 sections per station,
no limit on number of stations

StationName = STN01   // required; the defaults in this section apply only to this
station
Ton = 300            // seconds after calibration coil is engaged to wait to start
calibration

SwRamp = 240       // seconds to ramp up and down to full amplitude signal duration
SwNfft = 100       // size of nFFT window in samples
SwDec= 1           // amount to decimate signal (allowed values:
1,2,3,4,5,10,20,25,30,40,50,100)

PrbUnitWidth = 2.0          // seconds of minimum width
PrbNfft = 100               // size of nFFT window in samples
PrbDec = 1                  // amount to decimate signal (allowed values:
1,2,3,4,5,10,20,25,30,40,50,100)

PulseDelay=60               // seconds after pulse to include in analysis
PulseNfft=100               // size of nFFT window in samples
PulseDec = 1                // amount to decimate signal (allowed values:
1,2,3,4,5,10,20,25,30,40,50,100)
```

Appendix B
Message Syntax

This section shows the syntax for each of the supported message types and examples of message formats. For more detailed information on request message structure (for example, message line continuation) refer to [IDC3.4.1].

B.1 Appendix Conventions

The message syntax conventions for this Appendix are as follows:

Symbol Name	Symbol	Used to Indicate
angle brackets	< >	A rule (defined following the message syntax)
brackets	[ ]	An optional entry
pipe	|	Alternative elements
asterisk	*	The possibility of many elements (* indicates zero to many, 1* indicates one to many)

B.2 Help Messages

A help request retrieves the content of the station help file *HelpFile*. It does not require the basic message lines used for other types of requests, just the word `help` in the message body. For example:

```
From: your email address
To: AutoDRM email address
Subject:
---------------------message boundary
help
```

B.3 Data Messages

More than one data request can be included in an email message. Formats for the data output in data response messages comply with the specifications in [IDC 3.4.1]. Examples of all data output formats are provided in Appendix B of [IDC 3.4.1].

B.3.1 Data Request

Syntax

```
BEGIN IMS1.0 | IMS2.0
MSG_TYPE REQUEST
MSG_ID id_string [source]
[EMAIL address]
TIME <time> to <time>
[STA_LIST station_code]
[CHAN_LIST channels]
1* <request_line>
STOP
```

where:

```
<request_line> =
   CHANNEL [ims_format] | WAVEFORM ims_format [: compression] |
   RESPONSE [ims_format] | OUTAGE [ims_format]
<time> = yyyy[/mm[/dd[ hh[:mm[:ss[.sss]]]]]]
```

Example

```
begin ims1.0
msg_type request
msg_id Example_2
e-mail test@nanometrics.ca
time 2008/04/05 10:00:00.000 to 2008/04/05 10:02:00.000
sta_list *
waveform ims1.0 : cm6
outage
stop
```

B.3.2 Data Response

Syntax

```
BEGIN IMS1.0
MSG_TYPE DATA
MSG_ID id_string [source]
REF_ID ref_str [ref_src] PART # OF n
DATA_TYPE LOG IMS1.0
Original request information
1* <requested_data>
STOP
```

or

```
BEGIN IMS2.0
MSG_TYPE DATA
MSG_ID id_string [source]
REF_ID ref_str [ref_src] PART # OF n
1* <requested_data>
STOP
```

where:

```
<requested_data> = DATA_TYPE requested_data_type ims_format
                   formatted data
```

B.4 Command Messages

Command parsing has these restrictions on submitted requests:

- Only one command per request message is supported.
- All of the command parameters must precede the command keyword in the email. The only item that can follow the keyword is an enclosure (for example, a certificate) where appropriate for the command.
- Enclosures must follow the command keyword and precede the `STOP` keyword.

Errors encountered during parsing (for example, invalid date formatting) will return an error log identical to those for data requests. Problems occurring later in processing will be reported as described below.

B.4.1 Calibrate Start

The command request `CALIBRATE_START` is issued to calibrate IMS seismic, hydro-acoustic, and infrasound stations. It indicates the time of the calibration and details how the calibration should be conducted.

The command response is in two parts. The first part confirms the calibration (for example, that it is scheduled), the second part contains the calibration results.

B.4.1.1 Calibrate Start Request

Calibration command parsing imposes this restriction on submitted requests:

- The type declaration must precede the calibration parameters
- If only one pulse signal is specified, the only required parameter is duration. If multiple pulse signals are specified, then each subsequent signal specified after the first signal requires a delay parameter. The delay parameter specified for the second signal is applied to the first signal, the delay parameter specified for the third signal is applied to the second signal, and the delay value for the final signal is the default PulseDelay value defined in the `AutoDRM.ini` file.

  For example:

  ```
  calib_param 1.0
  calib_param 2.0 60.0
  calib_param 3.0 90.0
  ```

  In the above example,

  - The delay between the first and second signal is 60.0 seconds.
  - The delay between the second and third signals is 90.0 seconds.
  - The delay after the last signal is the default value from `AutoDRM.ini`: 60 seconds.

B.4.1.1.1 Syntax

```
BEGIN IMS2.0
MSG_TYPE COMMAND_REQUEST
MSG_ID id_string [source]
[E-MAIL address]
TIME_STAMP <time>
START_TIME <time>
[STA_LIST station_code]
[CHAN_LIST channels]
SENSOR yes | no
<pulse_calibration> | <random_calibration> | <sine_calibration>
CALIBRATE_START
STOP


where

<time> = yyyy[/mm[/dd[ hh[:mm[:ss[.sss]]]]]]
<pulse_calibration> =
   TYPE pulse
   CALIB_PARAM duration (seconds) [amplitude (volts)]
   * CALIB_PARAM duration (seconds) delay (seconds) [amplitude (volts)]
<random_calibration> =
   TYPE random
   1* CALIB_PARAM duration (seconds) [amplitude (volts)]
<sine_calibration> =
   TYPE sine
   1* CALIB_PARAM frequency (hertz) duration (seconds) [amplitude (volts)]
```

B.4.1.1.2 Pulse Example

```
begin ims2.0
msg_type command_request
msg_id Example_1
e-mail test@nanometrics.ca
time_stamp 2008/05/27 14:30
start_time 2008/05/27 14:37
sta_list Tr236
sensor yes
type pulse
calib_param 1.0
calib_param 2.0 60.0
calib_param 3.0 90.0
calibrate_start
stop
```

B.4.1.1.3 Random Example

```
begin ims2.0
msg_type command_request
msg_id Example_1
e-mail test@nanometrics.ca
time_stamp 2008/05/27 14:30
start_time 2008/05/27 14:37
sta_list Tr236
sensor yes
type random
calib_param 1.0
calibrate_start
stop
```

B.4.1.1.4 Sine Example

```
begin ims2.0
msg_type command_request
msg_id Example_1
e-mail test@nanometrics.ca
time_stamp 2008/05/27 14:30
start_time 2008/05/27 14:37
sta_list Tr236
sensor yes
type sine
calib_param 5 1.0
calibrate_start
stop
```

B.4.1.2 Calibrate Confirmation Response

Calibrations are scheduled by station. If there is a critical error preventing this command from being processed at all, a single error email will be sent out with the requested station list and channel list. Otherwise, a separate confirmation or critical error email will be sent out for each station as they are scheduled. If the scheduler rejects a particular calibration, it will have a start time of not confirmed. All calibrations that did not have a critical error will subsequently generate a results response by channel (even those that were rejected by the scheduler).

Syntax

```
BEGIN IMS2.0
MSG_TYPE COMMAND_RESPONSE
MSG_ID id_string [source]
REF_ID ref_str [ref_src]
TIME_STAMP yyyy/mm/dd hh:mm:ss
<indiv_response> | <indiv_error_message> | <overall_error_message>
STOP
```

where:

```
<indiv_response> = STA_LIST single station_code
                   CHAN_LIST requested channels for that station
                   START_TIME yyyy/mm/dd hh:mm:ss | not_confirmed
                   CALIBRATE_CONFIRM
<indiv_error_message> = STA_LIST single station_code
                        CHAN_LIST requested channels for that station
                        COMMAND CALIBRATE_START
                        PROBLEM_ENCOUNTERED problem description
<overall_error_message> = STA_LIST requested station_code
                          [CHAN_LIST requested channels]
                          COMMAND CALIBRATE_START
                          PROBLEM_ENCOUNTERED problem description
```

B.4.1.3 Calibrate Results Response

When the calibration has completed, a result email will be sent for each instrument channel that was calibrated.

Syntax

```
BEGIN IMS2.0
MSG_TYPE COMMAND_RESPONSE
MSG_ID id_string [source]
REF_ID ref_str [ref_src]
TIME_STAMP yyyy/mm/dd hh:mm:ss
STA_LIST single station name
[CHAN_LIST single channel]
CALIBRATE_RESULT
<normal_response> | PROBLEM_ENCOUNTERED problem description
STOP
```

where:

```
<individual_values> = CALIB value
                      CALPER value
<normal_response> = IN_SPEC yes | no
                    <individual_values> | system response in IMS2.0
```

B.4.2 Center Mass

The command request CENTER_MASS is issued to centre the mass at IMS seismic or hydro-acoustic stations.

B.4.2.1 Center Mass Request

Syntax

```
BEGIN IMS2.0
MSG_TYPE COMMAND_REQUEST
MSG_ID id_string [source]
[E-MAIL address]
TIME_STAMP <time>
[STA_LIST station_code]
[CHAN_LIST channels]
DURATION seconds
CENTER_MASS
STOP
```

where:

```
<time> = yyyy[/mm[/dd[ hh[:mm[:ss[.sss]]]]]]
```

Example

```
begin ims2.0
msg_type command_request
msg_id Example_2
e-mail test@nanometrics.ca
time_stamp 2008/05/27 14:35
sta_list Tr236
duration 1
center_mass
stop
```

B.4.2.2 Center Mass Response

Mass centring is done by station. If there is a critical error preventing this command from being processed at all, a single error email will be sent out with the requested station list and channel list. Otherwise, a separate confirmation or error email will be sent out for each station as it is centred. The system will allow up to 20 seconds for the centring to complete, after which time it will abort the attempt and issue an error email for that station.

Syntax

```
BEGIN IMS2.0
MSG_TYPE COMMAND_RESPONSE
MSG_ID id_string [source]
REF_ID ref_str [ref_src]
TIME_STAMP yyyy/mm/dd hh:mm:ss
<indiv_response> | <indiv_error_message> | <overall_error_message>
STOP
```

where:

```
<indiv_response> = STA_LIST single station_code
                   CHAN_LIST requested channels for that station
                   MASS_CENTERED
<indiv_error_message> = STA_LIST single station_code
                        CHAN_LIST requested channels for that station
                        COMMAND CENTER_MASS
                        PROBLEM_ENCOUNTERED problem description
<overall_error_message> = STA_LIST requested station_code
                          [CHAN_LIST requested channels]
                          COMMAND CENTER_MASS
                          PROBLEM_ENCOUNTERED problem description
```

B.4.3 Generate Keypair

The command request `GENERATE_KEYPAIR` is issued to generate new key pairs at instrument and Central Facility workstation security tokens (only the CF workstation running AutoDRM can be updated by this command). The DSA parameters can optionally be enclosed in the email, otherwise default values will be used.

B.4.3.1 Generate Keypair Request

Syntax

```
BEGIN IMS2.0
MSG_TYPE COMMAND_REQUEST
MSG_ID id_string [source]
[E-MAIL address]
TIME_STAMP <time>
[STA_LIST station_code]
[CHAN_LIST channels]
[AUTH_ID active_keypair_ID]
GENERATE_KEYPAIR
[PEM DSA parameters]
STOP
```

where:

```
<time> = yyyy[/mm[/dd[ hh[:mm[:ss[.sss]]]]]]
```

Example

```
begin ims2.0
msg_type command_request
msg_id Example_3
e-mail test@nanometrics.ca
time_stamp 2008/04/16 14:28
generate_keypair
stop
```

Example

```
begin ims2.0
msg_type command_request
msg_id Example_3a
e-mail test@nanometrics.ca
time_stamp 2008/04/16 14:28
sta_list STN01, CF*
generate_keypair
stop
```

B.4.3.2 Generate Keypair Response

Key pairs are generated by the instruments and the Central Facility workstation. For instruments, if there is a critical error preventing this command from being processed at all, a single error email will be sent out with the requested station list and channel list. Otherwise, a separate confirmation or error email will be sent out for each instrument as its key pair is generated. The response to a successful execution of the command request will include a Certificate Request for the new key pair. Similarly, for CF workstations a confirmation or error email will be sent out as its key pair is generated. The response to a successful request will include a Certificate Request for the new key pair.

Syntax

```
BEGIN IMS2.0
MSG_TYPE COMMAND_RESPONSE
MSG_ID id_string [source]
REF_ID ref_str [ref_src]
TIME_STAMP yyyy/mm/dd hh:mm:ss
<indiv_response> | <indiv_error_message> | <overall_error_message>
STOP
```

where:

```
<indiv_response> = STA_LIST instrument or workstation name
                   [AUTH_ID requested_Keypair_ID]
                   KEYPAIR_GENERATED
                   PEM certificate request
<indiv_error_message> = STA_LIST instrument or workstation name
                        [AUTH_ID requested_Keypair_ID]
                        COMMAND GENERATE_KEYPAIR
                        PROBLEM_ENCOUNTERED problem description
<overall_error_message> = STA_LIST requested station_code
                          [CHAN_LIST requested channels]
                          [AUTH_ID requested_Keypair_ID]
                          COMMAND GENERATE_KEYPAIR
                          PROBLEM_ENCOUNTERED problem description
```

B.4.4 Start Continuous

The command request START_CONTINUOUS is issued to enable and start all CD1.*x* senders configured to execute on the workstation running AutoDRM.

B.4.4.1 Start Continuous request

Syntax

```
BEGIN IMS2.0
MSG_TYPE COMMAND_REQUEST
MSG_ID id_string [source]
[E-MAIL address]
TIME_STAMP <time>
START_CONTINUOUS
STOP
```

where:

```
<time> = yyyy[/mm[/dd[ hh[:mm[:ss[.sss]]]]]]
```

Example

```
begin ims2.0
msg_type command_request
msg_id Example_4
e-mail test@nanometrics.ca
time_stamp 2008/04/22 17:28
start_continuous
stop
```

B.4.4.2 Start Continuous response

Syntax

```
BEGIN IMS2.0
MSG_TYPE COMMAND_RESPONSE
MSG_ID id_string [source]
REF_ID ref_str [ref_src]
TIME_STAMP yyyy/mm/dd hh:mm:ss
CONTINUOUS_STARTED | <error_message>
STOP
```

where:

```
<error_message> = COMMAND START_CONTINUOUS
                  PROBLEM_ENCOUNTERED problem description
```

B.4.5 Start Keypair

The command request START_KEYPAIR is issued to make a key pair active at the security token on an instrument or a Central Facility workstation. A certificate for the key pair must be enclosed in the email.

B.4.5.1 Start Keypair Request

Key pairs can be started immediately or at a future date. Start Keypair command parsing will impose this restriction on submitted requests:

- The station and channel list must resolve to a single instrument or workstation.

Syntax

```
BEGIN IMS2.0
MSG_TYPE COMMAND_REQUEST
MSG_ID id_string [source]
[E-MAIL address]
TIME_STAMP <time>
[START_TIME <time>]
[STA_LIST station_code]
[CHAN_LIST channels]
[AUTH_ID active_keypair_new_id]
START_KEYPAIR
PEM X509 certificate
STOP
```

where:

```
<time> = yyyy[/mm[/dd[ hh[:mm[:ss[.sss]]]]]]
```

Example

```
begin ims2.0
msg_type command_request
msg_id Example_6
e-mail test@nanometrics.ca
time_stamp 2008/04/27 14:28
sta_list STN31
chan_list BHZ
start_time 2008/04/27 15:26
start_keypair
-----BEGIN CERTIFICATE-----
MIIC8zCCArACAiE5MAsGByqGSM44BAMFADCBgzELMAkGA1UEBhMCY2ExEDAOBgNV
BAgTB09udGFyaW8xDzANBgNVBAcTBk90dGF3YTENMAsGA1UEChMEQWNtZTEQMA4G
A1UECxMHV2lkZ2V0czESMBAGA1UEAxMJUHJlc2lkZW50MRwwGgYJKoZIhvcNAQkB
Fg1wcmVzQGFjbWUuY29tMB4XDTA0MDQyNjE4MTkxOVoXDTA1MDQyNjE4MTkxOVow
PDEOMAwGA1UEChMFQ1RCVE8xDDAKBgNVBAsTA0lNUzEOMAwGA1UEAxMFU1ROMzEx
DDAKBgNVBAcTA05NWDCCAbcwggErBgcqhkjOOAQBMIIBHgKBgQD87GGC6yBrQ8A+
NsDq2r/1agwued70S8jy5TaZCW0f8nDxWXhddWkh2/+Xc64ISDtmL8B991Ev9osu
VWX9eYLiDCRIMquhIcwHmcwJ8tVBTV85ZiETZfUbg+n/zMs9iM3yOPfCc5Exynqt
/2Yv7B+w4dMRpAQmA3b9AR/gDQIEwwIVANOAc1O1HF9xsirD0MfjlBSPztxhAoGA
QuN3jm7DGw2wems3DX+2+0oLym3qrDcfatvL66ON33akfDw9eSdqDlec5ONHGA/Z
3kkCImMYNjFZqLn4FEiJVqxZQ15IyllHNjNqGTZM+lJYCYojmaew0mHqib99zl2B
g4/iYaaU8YD1b/jW9Gd9m4j6GGN/Ln6/ClDR7QBglmlWd4LN8xHm7nnnbXr2qwmD
XXYvcmGrOIQ1IvUICKze+MMAEneOCnnPMAsGByqGSM44BAMFAAMwADAtAhRHZ3oS
1zV3jGO6SPmE5cR6YhSEoAIVAL7ORvHFUXYs11u87pRWAby1NMkU
-----END CERTIFICATE-----
stop
```

B.4.5.2 Start Keypair Confirmation Response

If a key pair is scheduled to be made active at some future time, this initial response is returned. Note that when the start time is not confirmed, there will be no subsequent results response.

Syntax

```
BEGIN IMS2.0
MSG_TYPE COMMAND_RESPONSE
MSG_ID id_string [source]
REF_ID ref_str [ref_src]
TIME_STAMP yyyy/mm/dd hh:mm:ss
STA_LIST station_code
[CHAN_LIST channels]
[AUTH_ID requested_Keypair_ID]
START_TIME yyyy/mm/dd hh:mm:ss | not confirmed
KEYPAIR_CONFIRM | <error_message>
STOP
```

where:

```
<error_message> = COMMAND START_KEYPAIR
                  PROBLEM_ENCOUNTERED problem description
```

B.4.5.3 Start Keypair Results Response

When a key pair is being made active, this is the response that is sent. This is either the response to the initial request if the request is not future-dated, or is created as a separate response when processing begins at the requested time.

Syntax

```
BEGIN IMS2.0
MSG_TYPE COMMAND_RESPONSE
MSG_ID id_string [source]
REF_ID ref_str [ref_src]
TIME_STAMP yyyy/mm/dd hh:mm:ss
STA_LIST station_code
[CHAN_LIST channels]
[AUTH_ID requested_Keypair_ID]
KEYPAIR_STARTED | <error_message>
STOP
```

where:

```
<error_message> = COMMAND START_KEYPAIR
                  PROBLEM_ENCOUNTERED problem description
```

B.4.6 Stop Continuous

The command request `STOP_CONTINUOUS` is issued to stop all CD1.*x* senders configured to execute on the workstation running AutoDRM.

B.4.6.1 Stop Continuous request

Syntax

```
BEGIN IMS2.0
MSG_TYPE COMMAND_REQUEST
MSG_ID id_string [source]
[E-MAIL address]
TIME_STAMP <time>
STOP_CONTINUOUS
STOP
```

where:

```
<time> = yyyy[/mm[/dd[ hh[:mm[:ss[.sss]]]]]]
```

Example

```
begin ims2.0
msg_type command_request
msg_id Example_1
e-mail test@nanometrics.ca
time_stamp 2008/04/22 17:28
stop_continuous
stop
```

B.4.6.2 Stop Continuous response

Syntax

```
BEGIN IMS2.0
MSG_TYPE COMMAND_RESPONSE
MSG_ID id_string [source]
REF_ID ref_str [ref_src]
TIME_STAMP yyyy/mm/dd hh:mm:ss
CONTINUOUS_STOPPED | <error_message>
STOP
```

where:

```
<error_message> = COMMAND STOP_CONTINUOUS
                  PROBLEM_ENCOUNTERED problem description
```

B.4.7 Update CRL

The command request `UPDATE_CRL` is issued to load a certificate revocation list onto the workstation security token. The security token on the workstation running AutoDRM is automatically the target of this command.

B.4.7.1 Update CRL request

Update CRL command processing will impose this restriction on submitted requests:

- There must be a self-signed certificate for the principal that issued the CRL already installed on the workstation security token.

Syntax

```
BEGIN IMS2.0
MSG_TYPE COMMAND_REQUEST
MSG_ID id_string [source]
[E-MAIL address]
TIME_STAMP <time>
UPDATE_CRL
PEM X509 certificate revocation list
STOP
```

where:

```
<time> = yyyy[/mm[/dd[ hh[:mm[:ss[.sss]]]]]]
```

Example

```
begin ims2.0
msg_type command_request
msg_id Example_1
e-mail test@nanometrics.ca
time_stamp 2008/04/28 13:28
update_crl
-----BEGIN X509 CRL-----
MIIBHzCB3TALBgcqhkjOOAQDBQAwgYMxCzAJBgNVBAYTAmNhMRAwDgYDVQQIEwdP
bnRhcmlvMQ8wDQYDVQQHEwZPdHRhd2ExDTALBgNVBAoTBEFjbWUxEDAOBgNVBAsT
c0BhY21lLmNvbRcNMDMxMjExMTcwMTIyWhcNMDQwMTEwMTcwMTIyWjAqMBMCAiE1
Fw0wMzEyMTExNjU5MzBaMBMCAiE2Fw0wMzEyMTExNzAwNDJaMAsGByqGSM44BAMF
AAMwADAtAhRHsjCsbpX7x1EfUVNYYbD3TuE08QIVAO3RhCnL3d5Q7G2GCwa97RKt
ASOQ
-----END X509 CRL-----
stop
```

B.4.7.2 Update CRL Response

Syntax

```
BEGIN IMS2.0
MSG_TYPE COMMAND_RESPONSE
MSG_ID id_string [source]
REF_ID ref_str [ref_src]
TIME_STAMP yyyy/mm/dd hh:mm:ss
CRL_UPDATED | <error_message>
STOP
```

where:

```
<error_message> = COMMAND UPDATE_CRL
                  PROBLEM_ENCOUNTERED problem description
```

About Nanometrics

Nanometrics leads the world in the development of digital technology and networks for seismological and environmental studies. The award-winning Canadian exporter was the first company to produce a fully-integrated satellite system specially designed for studying and monitoring earthquakes.

Nanometrics has customers on every continent in more than 200 different countries. Our customers have used our technology to establish and grow research networks across every environment in the world from the frozen tundra of Canada's north to the arid deserts of the Middle East to the jungles of South America. Many of these include mission-critical national and regional networks that demand the highest possible data quality and availability.

Contacting Nanometrics

Nanometrics Inc.
250 Herzberg Road
Kanata, Ontario, Canada K2K 2A1
Phone: +1 613-592-6776
Fax: +1 613-592-5929
Email: info@nanometrics.ca
Web: www.nanometrics.ca

Contacting Technical Support

If you need technical support please submit a request on the Nanometrics technical support site or by email or fax. Include a full explanation of the problem and related information such as log files.

Support site:	http://support.nanometrics.ca
Email:	techsupport@nanometrics.ca