

SMARTGeoHub® Central Station Software Suite

User's Manual

Version 2.2

Copyright © 2011 Geotech Instruments, LLC
Dallas, Texas

Revision History

Rev	Date	Description	By
1.0	07/1/05	Preliminary Release	
1.1	08/15/08	First Release	GD, MR
1.2	09/15/08	Major update	GD, MR
1.3	10/7/08	Minors clarifications added	MR
1.4	05/15/09	Minors clarifications added	MR
1.5	05/25/09	Minors clarifications added	MR
1.6	09/25/09	SmartSeedLink plugin documentation added	MR
1.7	02/19/10	SMARTEWConfig added, SMARTServer and SMARTGen updates	MR
1.8	03/07/11	SMARTServer Earthworm direct updates	MR
1.9	05/05/11	CD1.1 data forwarding, SMARTCal, install updates	MR
2.0	05/26/11	SMARTCal updates	MR
2.1	8/2/11	Data Security Management Addendum	MR
2.2	08/18/11	SMARTCal updates	MR

Table of Contents

ABOUT THIS DOCUMENT	V
1 GENERAL INFORMATION	1-1
1.1 SMARTGeoHub® CENTRAL STATION OVERVIEW	1-1
1.2 SYSTEM REQUIREMENTS	1-2
2 SMARTGEOHUB® CENTRAL STATION INSTALLATION	2-3
2.1 INSTALLATION	2-3
2.2 UPDATE	2-3
3 SMARTGEOHUB® CENTRAL STATION OPERATION	3-5
3.1 SMARTGeoHub® FOR SMALL AND MEDIUM NETWORKS	3-5
3.1.1 Configuration	3-5
3.1.2 SMARTGeoHub® Data Server Operation	3-5
3.1.3 SMARTGeoHub® Data Server for Infrasound and Seismo-Acoustic Networks	3-7
3.1.4 SMARTGeoHub® CD1.1 Data Forwarding	3-9
3.1.5 SMARTGen Data File Generator Module	3-11
3.1.6 SMARTCal Calibration Module	3-14
3.2 SMARTGeoHub® FOR LARGE NETWORKS	3-22
3.2.1 SMARTGeoHub® 'Flat File' Data Server	3-22
3.2.2 Database Configuration File	3-23
3.2.3 Database Configuration Operation	3-25
3.2.4 FileGen Operation	3-26
3.2.4.1 FILEGEN SETTINGS	3-27
3.2.4.2 THE SIMPLE DATA FORMAT	3-28
3.3 SMARTGeoHub® DATA INTERFACE TO EARTHWORM	3-29
3.3.1 SMARTServer Configuration Notes	3-29
3.3.2 Earthworm Configuration Notes	3-32
3.3.2.1 IMPORT_GEN_PASV	3-32
3.3.2.2 FIR (OPTIONAL)	3-33
3.3.2.3 EXPORT_SCNL_ACK (OPTIONAL)	3-36
3.3.2.4 WAVE_SERVERV (OPTIONAL)	3-37
3.3.3 SmartEWConfig Configuration Tool	3-38

3.4 SMART-24 DATA INTERFACE TO SEISCOMP/SEEDLINK	3-41
4 APPENDIX A.....	4-43
5 COMMUNICATION AND DATA SECURITY MANAGEMENT	5-47
5.1 INSTALLATION OF THE COMMUNICATION AND DATA SECURITY MANAGEMENT SOFTWARE	5-47
5.2 INITIALIZATION OF THE FORTEZZA CARD(S) FOR THE WORKSTATION	5-47
5.3 INITIALIZATION OF THE FORTEZZA CARD FOR THE SMART-24.....	5-50
5.4 CONFIGURATION OF INCOMING MESSAGE VERIFIER	5-51
5.5 STATIONKEYCOMMAND	5-52
5.5.1 <i>Generate Key Pair</i>	5-53
5.5.2 <i>Start Key Pair</i>	5-56
5.5.3 <i>Get Key Pair</i>	5-57
5.6 SMARTKEYCOMMAND	5-58

List of Figures

Figure 1. SMARTServer running window.....	3-6
Figure 2. SMARTServer infrasound version running window.....	3-8
Figure 3. SMARTCD11Forwarder running window.....	3-11
Figure 4. SMARTGen window, output in SUDS (top) or CSS 3.0 (bottom) format.	3-12
Figure 5. SMARTGeoHub® programs (SMARTServer, SMARTGen, SMARTCD11Forwarder, and SMARTCal) running on Linux.....	3-15
Figure 6. SMART24Config window for Sine Wave (top) and Pseudo Random Binary (bottom) calibrations.....	3-16
Figure 7. SMARTCal window processing a sine wave calibration.....	3-17
Figure 8. SMARTCal window processing a random binary calibration	3-18
Figure 9 SMARTCalUpdate Select Window with a sine wave calibration results (top) and random binary calibration results (bottom)	3-19
Figure 10. SMARTCalUpdate confirmation window.....	3-19
Figure 11. SMARTCal and SMARTCalUpdate calibration files	3-20
Figure 12. SMARTCalUpdate local results updated window	3-20
Figure 13. SMARTCalUpdate cleanup window.....	3-21
Figure 14. SMARTCalUpdate SMART-24 updated window.....	3-21
Figure 15. SMART-24 Real Time Data Properties window with calibration parameters	3-21
Figure 16. SMARTServer 'flat file' at start-up (top) and in normal operation (bottom)	3-23
Figure 17. FileGen program window.....	3-27
Figure 18. SMARTGeoHub® Central Station data flow diagram.	4-43
Figure 19. SMARTGeoHub® with Earthworm extensions data flow diagram.....	4-45
Figure 20. StationKeyCommand GUI	5-52
Figure 21. Generate Key Pair window for Station with DSA parameters change.....	5-54
Figure 22. Generate Key Pair window for Site.....	5-54
Figure 23. KeyPair Generated message.....	5-55
Figure 24. Start Key Pair window.....	5-56
Figure 25. Key Pair Started message	5-56
Figure 26. Get Key Pair window	5-57
Figure 27. Key Pair Retrieved message.....	5-58
Figure 28. SMARTKeyCommand Window	5-58
Figure 29. SMARTKeyCommand Generate Key Pair dialog box	5-60
Figure 30. SMARTKeyCommand Start Key Pair dialog box	5-61
Figure 31. Manual Data Security Management using Fortezza cards	5-63

List of Tables

Table 1. Terminology correspondence between this manual and traditional seismological convention.	v
Table 2. List of SMARTGeoHub® programs	2-4
Table 3 Key management programs used and corresponding section in the manual, in function of the software and hardware environment.	5-47

About This Document

This manual is designed for technical personnel installing and operating Geotech Instruments' SMARTGeoHub® Central Station Software Suite (hereafter called "SMARTGeoHub® Central Station"). The following chapters are included in this manual:

- Chapter 1, General Information about SMARTGeoHub® Central Station
- Chapter 2, Installation and Setup
- Chapter 3, SMARTGeoHub® Central Station Features and Operation

In this manual, the following terminology is used (see Table 1):

The seismic or infrasound 'Station' designates an array of instruments recording and/or transmitting data. Each instrument is referred to as a 'Site'. Each instrument can have one or several 'Channels', for example each component of a seismometer corresponds to a different channel.

Table 1. Terminology correspondence between this manual and traditional seismological convention.

This manual	Seismological convention
Station	Network / Array
Site	Station
Channel	Component

(Page Intentionally Left Blank)

1 General Information

1.1 SMARTGeoHub® Central Station Overview

SMARTGeoHub® is a multi-threaded data acquisition system for real time operation.

Data providers are the SMART-24 Series instruments: SMART-24R® data recorder, SMART-24D® digitizer and SMART-24A® strong motion recorder. In the following text herein, all references will be generically SMART-24. Other data providers can be the DR-24 D-Series Remote Digitizers, or other digitizers communicating through supported protocols (SeedLink, Earthworm).

A data server (SMARTServer) stores seismic or infrasound array data on a backend relational database, or to data files, while multiple client applications are the user's interface to access these data. SMARTGeoHub® can also send data to other data acquisition systems using the Earthworm, SeedLink or CD1.1 protocol. A block diagram of the SMARTGeoHub® is shown in Appendix A, Figure 18.

SMARTGeoHub® Central Station includes the following software modules:

SMARTServer

This application is the data server that accepts real-time data from the SMART-24 stations and stores it in the relational database, for small and medium networks, or to 'flat' data files, for large networks, making it available to other clients. It also can forward data in real-time using Earthworm or CD1.1 protocol. This document describes in detail the SMARTServer features and operation.

SMARTDBConfig

This application is used to create the structure used by the SMARTGeoHub® Central Station database, consisting of station (or network), sites, and channels (or components), with their associated parameters. This application's features and operation are described in the SMARTDBConfig User's Manual (available separately).

SMART24Config

This application allows the user to configure the SMART-24 instruments. Via a TCP/IP or serial connection (PPP) to the SMART-24, this software lets the user setup and control all aspects of the operation of the SMART-24. This application's features and operation are described in the SMART24Config User's Manual (available separately).

SMARTGeoViewer®

This is a powerful, user-friendly software application for display of real-time seismic or infrasound data, designed to operate as either a local or remote client for the SMARTGeoHub® database server. It is a data viewer with extended zooming capabilities. This application's features and operation are described in the SMARTGeoViewer® User's Manual (available separately).

SMARTGeoSOH Monitor

This application checks the state of health (SOH) of remote SMART-24 instruments and their communication link status, displays on a graphical interface and logs the SOH alarm conditions. SMARTGeoSOHMonitor operates as either a local or remote client for SMARTGeoHub®'s DataBase Server. This application's features and operation are described in the SMARTGeoSOH User's Manual (available separately).

SMARTGen/FileGen

This application outputs the real-time data received from the remote stations as data files of fixed, user-selectable time duration. Either a simple data format can be used, or the SUDS format primarily for the use of the automated earthquake data processor, SMARTQuake®. In turn, SMARTQuake® can keep ring buffers in SUDS, miniSEED, SEG-Y, and CSS3.0 formats. This program operates as either a local or remote client for SMARTGeoHub®'s DataBase Server.

SeedLinkToSMART (optional)

This application is the data client that imports data from a SeedLink server over a TCP/IP connection, and stores it in the SMARTGeoHub®'s relational database, where it can be accessed by the user via client applications, e.g. SMARTGeoViewer® for data display. Using this software module, data from other non-Geotech digitizers which are supported by SeisComP data acquisition system can be imported into the SMARTGeoHub® Central Station.

DR2SMART (optional)

This application is the data server that accepts real-time data from the DR-24 stations, converts data from HLCP to CD1.1 protocol, and serves the data to the SMARTServer for storage to the relational database. Using this software module, data from the Geotech DR-24 D-Series Remote Digitizers can be imported into the SMARTGeoHub® Central Station.

SMARTCD11Forwarder (optional)

This application is an extension of the SMARTServer used to forward CD1.1 protocol data to external CD1.1 readers. Multiple CD1.1 destinations are supported, and LIFO data buffering for up to 14 days to recover data in case of communication outages.

SMARTCal (optional)

This application is a SMARTGeoHub® module used to automatically process calibrations and store calibration results based on SMART-24 Series internal calibration capabilities (see SMART-24 Series Manual for details).

1.2 System Requirements

The minimal hardware requirements are: a computer with Intel Pentium IV™ @ 2.26GHz processor, 1 GB RAM, with Microsoft Windows XP™ operating system. Additional resources may be required to support large seismic networks. Support for Linux and Unix operating systems is also available. The computer must support Java Virtual Machine and include a relational database (such as MySQL) that has a Java Data Base Connectivity driver.

2 SMARTGeoHub® Central Station Installation

Note: If SMARTGeoHub® software is already installed on the host computer, first save the station information data stored in the database to an XML file using SMARTDBConfig program (version 1.2 or higher).

2.1 INSTALLATION

Insert the Geotech's SMARTGeoHub® CD in the system CD drive. On a Windows platform run the **Install** program from the distribution CD root directory for a **full installation of the SMARTGeoHub® software**. In Windows 7, run the install program as administrator.

If SMARTGeoHub® software is already installed on the host computer, the above command updates with the newest version, overwriting the existing program files. Otherwise, a new installation is performed.

This procedure will create all folders and will copy all files necessary to run SMARTGeoHub®. The installation folder defaults to `c:\SmartGeoHub`, where the SMARTGeoHub® programs will be installed in different subfolders, one for each software component.

The Java Runtime Environment will be installed, or updated if an older version is already present on the computer. The "Typical" setup should be selected during the Java Runtime Environment Install Wizard.

The install procedure will also install the MySQL relational database required for SMARTGeoHub® to operate.

If MySQL database is already installed on the host computer, the existing mysql folder (with all programs and data files) is renamed as "mysql.old", and a new database server is installed in the `c:\mysql` folder. The station information data can be restored to the database from the previously saved backup XML file using a database restore tool called "restore-db" included in SMARTDBConfig software (version 1.2 or higher).

2.2 UPDATE

Run the **Update** program from the distribution CD root directory for **updating the SMARTGeoHub® software** with the newest version. In Windows 7, run the update command as administrator. The MySQL database (programs and data) is kept unchanged.

When the software installation (or update) is completed, it is recommended to create new (or update existing) shortcuts on the Desktop to point to the SMARTGeoHub® programs (see Table 2).

Also, it is recommended to set the program icons (by editing the shortcut Properties) to point to the corresponding icon files stored in the `c:\SMARTGeoHub\icons` folder, e.g. `c:\SMARTGeoHub\icons\Smart24Config.ico` for Smart24Config program.

The SMARTGeoViewer client is license protected. It has a 30-days evaluation period following installation. The permanent license key is distributed from the factory by email or phone, after the user provides the user codes obtained during product activation.

Table 2. List of SMARTGeoHub® programs

Program Name	Program File Name and Path	Function
SMART24Config	c:\SMARTGeoHub\SMART24Config\bin\ Smart24Config.bat	SMART-24 instrument configuration client
SMART24Reader	c:\SMARTGeoHub\SMART24Reader\ Smart24Reader.exe	Quick display & data format conversion of SMART-24 raw data files
SMARTDBConfig	c:\SMARTGeoHub\SMARTDB_Config- x.x\bin\SmartDB_Config.bat	Database configuration for real-time operation client
SMARTServer	c:\SMARTGeoHub\SMARTServer- x.xx\bin\runServer.bat or	Real-time data acquisition for small & medium networks
	c:\SMARTGeoHub\SMARTToEarth\bin\ runSmartToEarth.bat or	SmartServer with Earthworm data exporter
	c:\SMARTGeoHub\SMARTServerFF\bin\ runServer.bat	SmartServer for large networks (“flat files” database model)
SMARTGeoViewer	c:\SMARTGeoHub\SMARTGeoViewer- x.xx\SmartGeoViewer.bat	Real-time waveform display client
SMARTGeoSOH	c:\SMARTGeoHub\SMARTGeoSOH_Monit or-x.x\bin\SOH_Monitor.bat	SMART-24 real-time SOH data monitoring client
SMARTGen	c:\SMARTGeoHub\SMARTGen\bin\ SmartGen.bat or	Real-time data files creator client for small & medium networks
	c:\SMARTGeoHub\FileGen\bin\FileGen.bat	Real-time data files creator client for large networks
SMARTCD11Forwarder (optional)	c:\SMARTCD11Forwarder\bin\ SMARTCD11Forwarder.bat	Real-time CD1.1 data forwarder (typically for use in infrasound or seismo- acoustic networks)

3 SMARTGeoHub® Central Station Operation

3.1 SMARTGeoHub® for Small and Medium Networks

The SMARTGeoHub® relational database model is used on small and medium networks involving less than 100 channels. Its main advantage consists in providing simple, easy to use program interfaces to the user.

3.1.1 Configuration

Remote sites connect to the SMARTGeoHub® Central Station over a TCP/IP network. They communicate through a well-defined protocol that authenticates pre-defined clients. In order for the SMARTGeoHub® to accept connections from remote sites, the seismic or infrasound array network must be configured in the database, using the SMARTDBConfig program. The user should refer to the SMARTDBConfig User's Manual for a description of the database configuration operations.

The site specific parameters set in the database with SMARTDBConfig must agree with corresponding parameters set up on the respective SMART-24 remote station (e.g. site and channel (or component) names, and IP address). The user should refer to the SMART24Config User's Manual for a description of the SMART-24 instrument configuration operations.

3.1.2 SMARTGeoHub® Data Server Operation

The SMARTServer is used to accept incoming real-time data from SMART-24 instruments via a TCP/IP network, using the CD-1.1 frame format and protocol described in the IDC documentation "Formats and Protocols for Continuous Data CD-1.1, 3.4.3 Revision 0.3".

The SMARTServer application can accept SMART-24 connections after the remote SMART-24 sites are configured to send CD1.1 real-time data to the SMARTGeoHub® Central Station computer, identified by its IP address over the TCP/IP network, and after the SMARTGeoHub® database was configured with the relevant station (or network) and site parameters.

To start the SMARTServer, double-click with the mouse on its shortcut on the computer's Desktop, or execute the program file (given here as an example):

```
c:\SMARTGeoHub\SMARTServer\bin\runServer.bat
```

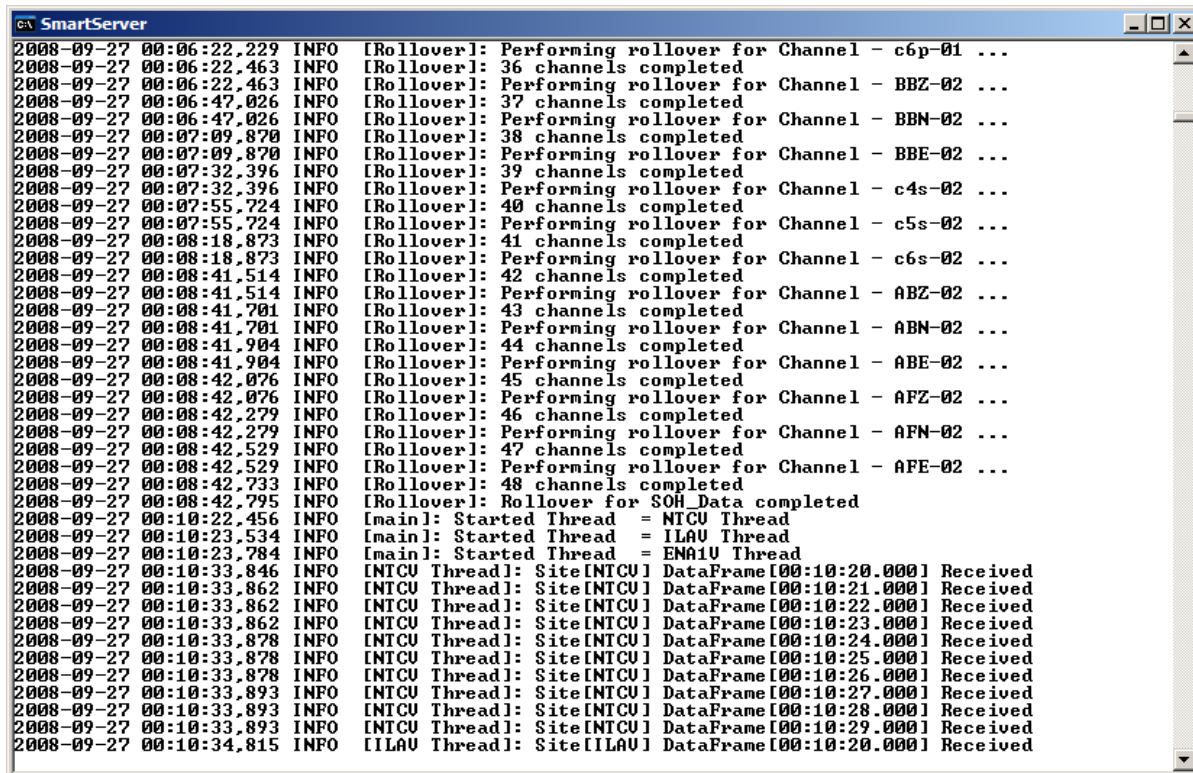
SMARTServer is the primary program required to run in order for real-time data to be acquired into the SMARTGeoHub® system. Depending on the specific system application, other programs can be required to run as well, for example SMARTGen for generating continuous data files, and/or SMARTQuake® for automated earthquake data processing. In such cases, typically, a program file `c:\SMARTGeoHub\StartAll.bat` is provided for starting all the programs that are critical for performing real-time, automated data acquisition and processing. Also, a program file `c:\SMARTGeoHub\StopAll.bat` is provided for stopping the SMARTServer and all the other real-time data acquisition and processing tasks.

Initially, when started, SMARTServer executes a data clean-up operation, deleting old data from the SMARTGeoHub®'s MySQL database. Typically the most recent 1 to 3 days of data are kept into the database. SMARTServer performs the data clean-up operation daily, always deleting the oldest data day in the database.

SMART-24 remote sites attempt to establish a data connection with the SMARTGeoHub® typically every two minutes. Upon receipt of a SMART-24 data connection request, the SMARTServer validates the site name and IP address of the sender against the site names and IP addresses registered in the local database. If the site is registered, SMARTServer will acknowledge and commence to receive the data frames from the remote site, and store them into the system.

As soon as data are received by the SMARTServer, they become available to other SMARTGeoHub® client applications, e.g. for data display with the SMARTGeoViewer® program. The user should refer to the SMARTGeoViewer® User's Manual for a description of the data display operations.

The SMARTServer logs the site name and data frame time in its console window (shown as an example in Figure 1), and to log files stored in a log folder (e.g. `c:\SMARTGeoHub\SMARTServer\logs`). The maximum number of log files to keep, their maximum size and the logging level is set in the configuration file `c:\SMARTGeoHub\SMARTServer\conf\log4j.properties`; the parameters to set are `log4j.appender.drfa.MaxBackupIndex` (e.g. to 100), `log4j.appender.drfa.MaxFileSize` (e.g. to 1000KB), and `log4j.rootLogger` (to INFO or DEBUG) respectively.



```

SmartServer
2008-09-27 00:06:22,229 INFO [Rollover]: Performing rollover for Channel - c6p-01 ...
2008-09-27 00:06:22,463 INFO [Rollover]: 36 channels completed
2008-09-27 00:06:22,463 INFO [Rollover]: Performing rollover for Channel - BBZ-02 ...
2008-09-27 00:06:47,026 INFO [Rollover]: 37 channels completed
2008-09-27 00:06:47,026 INFO [Rollover]: Performing rollover for Channel - BBN-02 ...
2008-09-27 00:07:09,870 INFO [Rollover]: 38 channels completed
2008-09-27 00:07:09,870 INFO [Rollover]: Performing rollover for Channel - BBE-02 ...
2008-09-27 00:07:32,396 INFO [Rollover]: 39 channels completed
2008-09-27 00:07:32,396 INFO [Rollover]: Performing rollover for Channel - c4s-02 ...
2008-09-27 00:07:55,724 INFO [Rollover]: 40 channels completed
2008-09-27 00:07:55,724 INFO [Rollover]: Performing rollover for Channel - c5s-02 ...
2008-09-27 00:08:18,873 INFO [Rollover]: 41 channels completed
2008-09-27 00:08:18,873 INFO [Rollover]: Performing rollover for Channel - c6s-02 ...
2008-09-27 00:08:41,514 INFO [Rollover]: 42 channels completed
2008-09-27 00:08:41,514 INFO [Rollover]: Performing rollover for Channel - ABZ-02 ...
2008-09-27 00:08:41,701 INFO [Rollover]: 43 channels completed
2008-09-27 00:08:41,701 INFO [Rollover]: Performing rollover for Channel - ABN-02 ...
2008-09-27 00:08:41,904 INFO [Rollover]: 44 channels completed
2008-09-27 00:08:41,904 INFO [Rollover]: Performing rollover for Channel - ABE-02 ...
2008-09-27 00:08:42,076 INFO [Rollover]: 45 channels completed
2008-09-27 00:08:42,076 INFO [Rollover]: Performing rollover for Channel - AFZ-02 ...
2008-09-27 00:08:42,279 INFO [Rollover]: 46 channels completed
2008-09-27 00:08:42,279 INFO [Rollover]: Performing rollover for Channel - AFN-02 ...
2008-09-27 00:08:42,529 INFO [Rollover]: 47 channels completed
2008-09-27 00:08:42,529 INFO [Rollover]: Performing rollover for Channel - AFE-02 ...
2008-09-27 00:08:42,733 INFO [Rollover]: 48 channels completed
2008-09-27 00:08:42,795 INFO [Rollover]: Rollover for SOH_Data completed
2008-09-27 00:10:22,456 INFO [main]: Started Thread = NTCU Thread
2008-09-27 00:10:23,534 INFO [main]: Started Thread = ILAU Thread
2008-09-27 00:10:23,784 INFO [main]: Started Thread = EN01U Thread
2008-09-27 00:10:33,846 INFO [NTCU Thread]: Site[NTCU] DataFrame[00:10:20.000] Received
2008-09-27 00:10:33,862 INFO [NTCU Thread]: Site[NTCU] DataFrame[00:10:21.000] Received
2008-09-27 00:10:33,862 INFO [NTCU Thread]: Site[NTCU] DataFrame[00:10:22.000] Received
2008-09-27 00:10:33,862 INFO [NTCU Thread]: Site[NTCU] DataFrame[00:10:23.000] Received
2008-09-27 00:10:33,878 INFO [NTCU Thread]: Site[NTCU] DataFrame[00:10:24.000] Received
2008-09-27 00:10:33,878 INFO [NTCU Thread]: Site[NTCU] DataFrame[00:10:25.000] Received
2008-09-27 00:10:33,878 INFO [NTCU Thread]: Site[NTCU] DataFrame[00:10:26.000] Received
2008-09-27 00:10:33,893 INFO [NTCU Thread]: Site[NTCU] DataFrame[00:10:27.000] Received
2008-09-27 00:10:33,893 INFO [NTCU Thread]: Site[NTCU] DataFrame[00:10:28.000] Received
2008-09-27 00:10:33,893 INFO [NTCU Thread]: Site[NTCU] DataFrame[00:10:29.000] Received
2008-09-27 00:10:34,815 INFO [ILAU Thread]: Site[ILAU] DataFrame[00:10:20.000] Received
  
```

Figure 1. SMARTServer running window

3.1.3 SMARTGeoHub® Data Server for Infrasound and Seismo-Acoustic Networks

Typically for use in infrasound or seismo-acoustic networks, a version of the SMARTServer (identified by version number 3.0 or higher) can generate station CD1.1 data frames by assembling the CD1.1 data frames received from the SMART-24 instruments, digitally sign data at the hub (using a hardware authentication token) and forward CD1.1 data to external CD1.1 receivers, via a program module called SMARTCD11Forwarder.

SMARTServer must be configured before attempting to run initially in this mode. The SMARTServer configuration parameters specific to infrasound/seismo-acoustic operation should be changed by editing the configuration file `viewer-opts.conf`, located e.g. at:

```
c:\SmartGeoHub\SmartServer\conf\viewer-opts.conf.
```

The parameter file settings of interest are contained in the 'Infrasound_Info' section:

[Infrasound_Info]

<code>cd11.generate=1;</code>	Defines whether generating station CD1.1 data frames for all SMART-24 digitizers reporting is enabled ('1') or disabled ('0'). Example implies enabled, so that station CD1.1 frames are created for forwarding to external CD1.1 readers.
<code>station.sign=1;</code>	Defines whether digital signing of the station CD1.1 data frames at the hub is enabled ('1') or disabled ('0'). A hardware authentication token is used for data digital signing. Example implies enabled, so that station CD1.1 frames are signed before being forwarded to external CD1.1 readers.
<code>auth.keyid=1;</code>	Defines the identifier of the certificate with the public key required to verify the digital signature, if non-zero authentication is used to verify communications. Example implies '1', meaning authentication verification is enabled with an authentication key id value of 1.
<code>infrasound.dataframesize=10;</code>	Defines the data frame size in seconds for all SMART-24 digitizers reporting. Valid values are 10, 20 or 30 seconds. Example implies '10', meaning the digitizer's data frame size (set with SMART24Config software) is 10 seconds.
<code>frame.destination=0;</code>	Defines the identifier of the destination of the frame (8-character ASCII) set by the SMARTServer in the station CD1.1 data frame header. Example implies '0' (default).
<code>token.luna=1;</code>	Defines whether a LunaPCM authentication token is used ('1') or a Fortezza card ('0') for digital data signing. Example implies '1' (LunaPCM).
<code>forwarder.recoverqueue=6;</code>	Defines how many data frames are resent for forwarding, when re-establishing communication between SMARTServer and SMARTCD11Forwarder after it has been interrupted. Example implies '6', or 1 minute when using 10-seconds data frames.

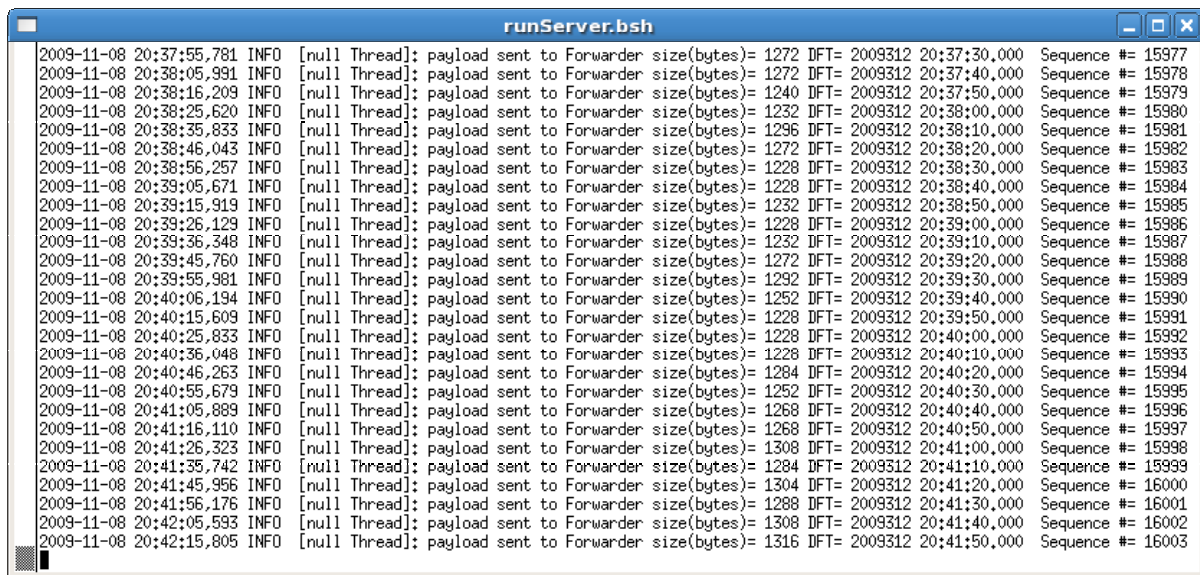
`maxnumber.stations=3;` Defines the number of SMART-24 digitizers reporting. Example implies 3 sites. In this case entries for all 3 sites must be present in the [Infrasound_Info] section, similar to the examples given below for the first site.

`sta.port.1=11656;` Defines the data port address assigned to the SMART-24 digitizer for data reception, and must agree with the corresponding site data port stored in the database. Example implies site number one. Each SMART-24 would have a unique data port assignment and would be identified numerically sequential, i.e. `sta.port.2=...`, `sta.port.n=...`, where `n= maxnumber.stations`. The recommended parameter values are `10000 + SMART-24 serial number`. Thus the example implies serial number 1656.

`cd11Queue.serverip=192.168.100.108;` Defines the IP address of the host computer where SMARTCD11Forwarder program resides, typically the same computer where SMARTGeoHub® (and SMARTServer) resides too. SMARTServer will forward CD1.1 data to the SMARTCD11Forwarder at this IP address, and data port address specified below, which in turn will queue and forward the data to external CD1.1 reader(s).

`cd11Queue.serverport=9010;` Defines the data port address to connect and send data to the SMARTCD11Forwarder program. The default value of 9010 is recommended to be used.

This version of SMARTServer logs the CD1.1 station data frame time, sequence number and size (in bytes) in its console window, as shown in Figure 2.



```

runServer.bsh
2009-11-08 20:37:55,781 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1272 DFT= 2009312 20:37:30,000 Sequence == 15977
2009-11-08 20:38:05,991 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1272 DFT= 2009312 20:37:40,000 Sequence == 15978
2009-11-08 20:38:16,209 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1240 DFT= 2009312 20:37:50,000 Sequence == 15979
2009-11-08 20:38:25,620 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1232 DFT= 2009312 20:38:00,000 Sequence == 15980
2009-11-08 20:38:35,833 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1296 DFT= 2009312 20:38:10,000 Sequence == 15981
2009-11-08 20:38:46,043 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1272 DFT= 2009312 20:38:20,000 Sequence == 15982
2009-11-08 20:38:56,257 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1228 DFT= 2009312 20:38:30,000 Sequence == 15983
2009-11-08 20:39:05,671 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1228 DFT= 2009312 20:38:40,000 Sequence == 15984
2009-11-08 20:39:15,919 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1232 DFT= 2009312 20:38:50,000 Sequence == 15985
2009-11-08 20:39:26,129 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1228 DFT= 2009312 20:39:00,000 Sequence == 15986
2009-11-08 20:39:36,348 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1232 DFT= 2009312 20:39:10,000 Sequence == 15987
2009-11-08 20:39:45,760 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1272 DFT= 2009312 20:39:20,000 Sequence == 15988
2009-11-08 20:39:55,981 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1292 DFT= 2009312 20:39:30,000 Sequence == 15989
2009-11-08 20:40:06,194 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1252 DFT= 2009312 20:39:40,000 Sequence == 15990
2009-11-08 20:40:15,609 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1228 DFT= 2009312 20:39:50,000 Sequence == 15991
2009-11-08 20:40:25,833 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1228 DFT= 2009312 20:40:00,000 Sequence == 15992
2009-11-08 20:40:36,048 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1228 DFT= 2009312 20:40:10,000 Sequence == 15993
2009-11-08 20:40:46,263 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1284 DFT= 2009312 20:40:20,000 Sequence == 15994
2009-11-08 20:40:55,679 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1252 DFT= 2009312 20:40:30,000 Sequence == 15995
2009-11-08 20:41:05,889 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1268 DFT= 2009312 20:40:40,000 Sequence == 15996
2009-11-08 20:41:16,110 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1268 DFT= 2009312 20:40:50,000 Sequence == 15997
2009-11-08 20:41:26,323 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1308 DFT= 2009312 20:41:00,000 Sequence == 15998
2009-11-08 20:41:36,742 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1284 DFT= 2009312 20:41:10,000 Sequence == 15999
2009-11-08 20:41:45,956 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1304 DFT= 2009312 20:41:20,000 Sequence == 16000
2009-11-08 20:41:56,176 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1288 DFT= 2009312 20:41:30,000 Sequence == 16001
2009-11-08 20:42:05,593 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1308 DFT= 2009312 20:41:40,000 Sequence == 16002
2009-11-08 20:42:15,805 INFO [null Thread]: payload sent to Forwarder size(bytes)= 1316 DFT= 2009312 20:41:50,000 Sequence == 16003
  
```

Figure 2. SMARTServer infrasound version running window

3.1.4 SMARTGeoHub® CD1.1 Data Forwarding

This application is an extension of the SMARTServer infrasound version used to forward CD1.1 protocol data to external CD1.1 readers. Multiple CD1.1 destinations are supported, and LIFO data buffering for up to 14 days to recover data in case of communication outages.

SMARTCD11Forwarder must be first configured by editing the configuration file `configuration.xml`, located e.g. at:

```
c:\SMARTCD11Forwarder\bin\configuration.xml
```

The settings of interest in this file are:

<code>keep.days="8"</code>	Defines the size of the saved data queues (in days). Example implies 8 days, meaning all data frames and forwarded data status information are stored for 8 days. In case of communication outages, data frames older than 'keep.days' will not be sent.
<code>id="SMART"</code>	Defines the queues identifier (ASCII) used to name the queue files. Example implies "SMART". The queues are saved on the hard disk in the folder <code>c:\SMARTCD11Forwarder\bin\queues</code> , as daily files with the file name containing the queue identifier followed by calendar date (year, month and day of month), and extensions <code>.ndx</code> and <code>.que</code> , e.g. <code>SMART-20110426.ndx</code> , <code>SMART-20110426.que</code> . The queue files are not maintained automatically and it is recommended the old queues to be cleaned manually, e.g. once every 1-3 months.

For each CD1.1 destination there is a corresponding line beginning with "forward" and containing the following parameters:

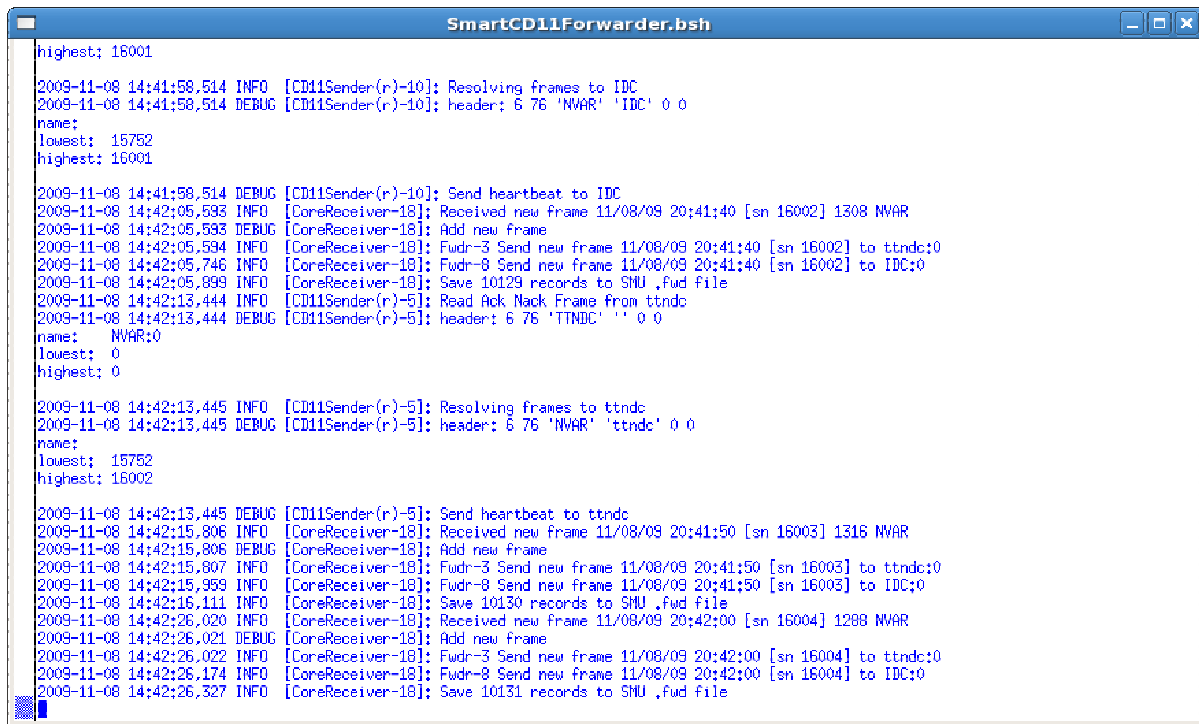
<code>enabled="true"</code>	Defines whether this CD1.1 destination is enabled ('true') or disabled ('false') for data sending to it. Example implies 'true', meaning CD1.1 data are forwarded by the SMARTCD11Forwarder to the corresponding CD1.1 receiver.
<code>source="SMART"</code>	Defines the CD1.1 source identifier (ASCII), or data frame creator stored in the frame headers, and must agree with the station name stored in the database. Example implies 'SMART'.
<code>consumer="IDC"</code>	Defines the CD1.1 consumer (receiver) identifier (ASCII), used for logging purposes on the hub. Example implies 'IDC'.
<code>destination="IDC"</code>	Defines the CD1.1 destination identifier (ASCII), stored in the frame headers as "frame destination" (see IDC documentation on formats and protocols for continuous data). Example implies 'IDC'.
<code>ip="192.168.100.56"</code>	Defines the IP address of the destination CD1.1 reader computer. SMARTCD11Forwarder will forward CD1.1 data to the external CD1.1 reader at this IP address, using the connection data port address specified below.
<code>port="8000"</code>	Defines the data port address to connect for sending data to the external CD1.1 reader.

MaxFrames2Store="6"	Defines how many data frames are resent when re-establishing communication between SMARTCD11Forwarder and the external CD1.1 reader after it has been interrupted. Example implies '6', which for 10-seconds data frames represents 1 minute of data.
KeyID="0"	Defines the identifier of the certificate with the public key required to verify the digital signature; if non-zero non-data frames generated by SMARTCD11Forwarder are digitally signed, and authentication is used to verify communications. Set to zero to disable digital data signing of non-data frames by SMARTCD11Forwarder. <i>Note that only one forwarder destination can have data signing enabled (non-zero value of the KeyID).</i> Example implies '0', meaning data signing (and authentication verification) is disabled for this CD1.1 destination.
SendHeartbeat="0"	Defines the send heartbeat interval (in seconds), and the role of the SMARTCD11Forwarder in sending heartbeats: if this value is '0' the SMARTCD11Forwarder program does not initiate sending heartbeats, but responds to heartbeats received from the CD1.1 reader with Ack-Nack frames; if this value is non-zero, e.g. '60', the SMARTCD11Forwarder sends heartbeats (Ack-Nack frames) every 60 seconds. Example implies '0' meaning heartbeats are initiated by the CD1.1 reader and responded by SMARTCD11Forwarder.

SMARTCD11Forwarder program should only be started after SMARTServer is already running. Typically both SMARTServer and SMARTCD11Forwarder applications are started automatically, by running the StartAll shortcut. Otherwise SMARTCD11Forwarder can be started by double-clicking on the SMARTCD11Forwarder shortcut that points to the following program file:

c:\SMARTCD11Forwarder\bin\SMARTCD11Forwarder.bat

The program window logs the data frames received from the SMARTServer, and forwarded to each of the CD1.1 destinations (see Figure 3). This information is also stored to daily log files stored in the folder c:\SMARTCD11Forwarder\logs; the current log file is named 'CD11Reader.log' and the older log files are renamed with the extension given by their calendar date (year, month and day of month), e.g. CD11Reader.log.2011-04-26. The log files are not maintained automatically and it is recommended the old logs to be cleaned manually, e.g. once every 1-3 months.



```

highest; 16001

2009-11-08 14:41:58.514 INFO [CD11Sender(r)-10]: Resolving frames to IDC
2009-11-08 14:41:58.514 DEBUG [CD11Sender(r)-10]: header: 6 76 'NVAR' 'IDC' 0 0
name:
lowest: 15752
highest: 16001

2009-11-08 14:41:58.514 DEBUG [CD11Sender(r)-10]: Send heartbeat to IDC
2009-11-08 14:42:05.593 INFO [CoreReceiver-18]: Received new frame 11/08/09 20:41:40 [sn 16002] 1308 NVAR
2009-11-08 14:42:05.593 DEBUG [CoreReceiver-18]: Add new frame
2009-11-08 14:42:05.594 INFO [CoreReceiver-18]: Fudr-3 Send new frame 11/08/09 20:41:40 [sn 16002] to ttndc:0
2009-11-08 14:42:05.746 INFO [CoreReceiver-18]: Fudr-8 Send new frame 11/08/09 20:41:40 [sn 16002] to IDC:0
2009-11-08 14:42:05.899 INFO [CoreReceiver-18]: Save 10129 records to SHU .fud file
2009-11-08 14:42:13.444 INFO [CD11Sender(r)-5]: Read Ack Nack Frame from ttndc
2009-11-08 14:42:13.444 DEBUG [CD11Sender(r)-5]: header: 6 76 'TTNDC' '' 0 0
name: NVAR:0
lowest: 0
highest: 0

2009-11-08 14:42:13.445 INFO [CD11Sender(r)-5]: Resolving frames to ttndc
2009-11-08 14:42:13.445 DEBUG [CD11Sender(r)-5]: header: 6 76 'NVAR' 'ttndc' 0 0
name:
lowest: 15752
highest: 16002

2009-11-08 14:42:13.445 DEBUG [CD11Sender(r)-5]: Send heartbeat to ttndc
2009-11-08 14:42:15.806 INFO [CoreReceiver-18]: Received new frame 11/08/09 20:41:50 [sn 16003] 1316 NVAR
2009-11-08 14:42:15.806 DEBUG [CoreReceiver-18]: Add new frame
2009-11-08 14:42:15.807 INFO [CoreReceiver-18]: Fudr-3 Send new frame 11/08/09 20:41:50 [sn 16003] to ttndc:0
2009-11-08 14:42:15.959 INFO [CoreReceiver-18]: Fudr-8 Send new frame 11/08/09 20:41:50 [sn 16003] to IDC:0
2009-11-08 14:42:16.111 INFO [CoreReceiver-18]: Save 10130 records to SHU .fud file
2009-11-08 14:42:26.020 INFO [CoreReceiver-18]: Received new frame 11/08/09 20:42:00 [sn 16004] 1288 NVAR
2009-11-08 14:42:26.021 DEBUG [CoreReceiver-18]: Add new frame
2009-11-08 14:42:26.022 INFO [CoreReceiver-18]: Fudr-3 Send new frame 11/08/09 20:42:00 [sn 16004] to ttndc:0
2009-11-08 14:42:26.174 INFO [CoreReceiver-18]: Fudr-8 Send new frame 11/08/09 20:42:00 [sn 16004] to IDC:0
2009-11-08 14:42:26.327 INFO [CoreReceiver-18]: Save 10131 records to SHU .fud file

```

Figure 3. SMARTCD11Forwarder running window

3.1.5 SMARTGen Data File Generator Module

The SMARTGen program can be used to write raw data in near real-time as data files of fixed duration, for the use of an automated processing software (e.g. SMARTQuake®). The data files are formatted using SUDS format (Banfill, 2003), or CSS 3.0 database.

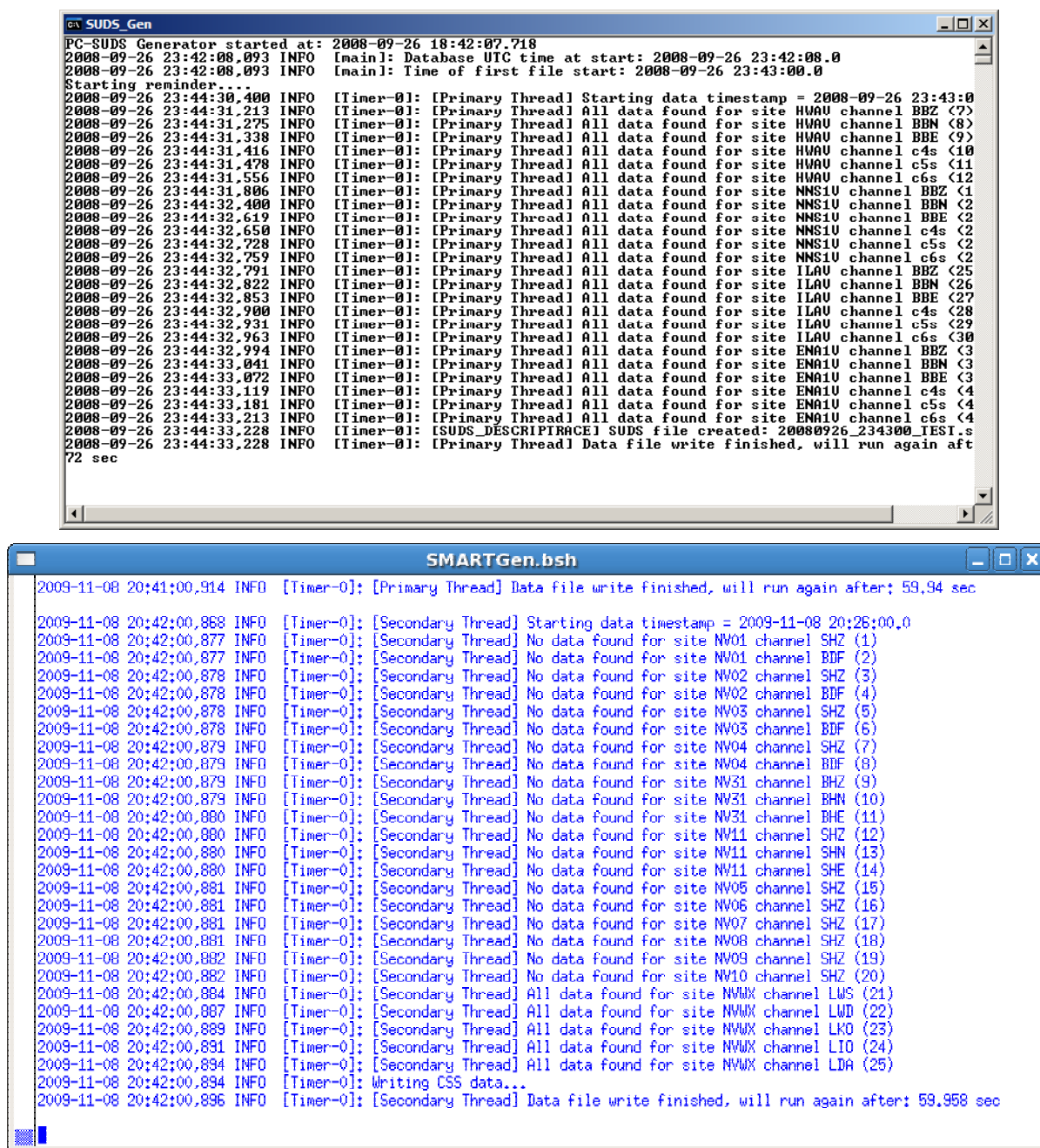
SMARTGen program should only be started after SMARTServer is already running. Typically both SMARTServer and SMARTGen applications (and optionally other programs as well, e.g. SMARTCal) are started automatically by running the `StartAll` shortcut. Otherwise SMARTGen can be started by double-clicking on the SMARTGen shortcut that points to the following program file:

```
c:\SMARTGeoHub\SMARTGen\bin\SMARTGen.bat.
```

Depending on the selected configuration, immediately at program start or at the beginning of next minute marks the start time of the first file. The data are extracted from the SMARTGeoHub® database periodically, with a user programmable delay versus real-time to allow for communications delays, and written to data files. The data files contain all data channels recorded by the SMARTGeoHub® for the operator specified recording time period.

SMARTGen relies on the local computer time to determine when to run, therefore it requires the time of the host computer's clock to be close (within few seconds) to the SMART-24 digitizers' time. It is strongly recommended for the host computer's clock to be automatically synchronized with a time server (either a Internet time server, or local).

SMARTGen logs the site and channels names, the data file name and its start date and time in the program's console window (shown as an example in Figure 4).



```

PC-SUDS Generator started at: 2008-09-26 18:42:07.718
2008-09-26 23:42:08.093 INFO [main]: Database UTC time at start: 2008-09-26 23:42:08.0
2008-09-26 23:42:08.093 INFO [main]: Time of first file start: 2008-09-26 23:43:00.0
Starting reminder....
2008-09-26 23:44:30.400 INFO [Timer-01: [Primary Thread] Starting data timestamp = 2008-09-26 23:43:00
2008-09-26 23:44:31.213 INFO [Timer-01: [Primary Thread] All data found for site HVAU channel BBZ <7>
2008-09-26 23:44:31.275 INFO [Timer-01: [Primary Thread] All data found for site HVAU channel BBN <8>
2008-09-26 23:44:31.338 INFO [Timer-01: [Primary Thread] All data found for site HVAU channel BBE <9>
2008-09-26 23:44:31.416 INFO [Timer-01: [Primary Thread] All data found for site HVAU channel c4s <10>
2008-09-26 23:44:31.478 INFO [Timer-01: [Primary Thread] All data found for site HVAU channel c5s <11>
2008-09-26 23:44:31.556 INFO [Timer-01: [Primary Thread] All data found for site HVAU channel c6s <12>
2008-09-26 23:44:31.806 INFO [Timer-01: [Primary Thread] All data found for site NNS1U channel BBZ <1>
2008-09-26 23:44:32.400 INFO [Timer-01: [Primary Thread] All data found for site NNS1U channel BBN <2>
2008-09-26 23:44:32.619 INFO [Timer-01: [Primary Thread] All data found for site NNS1U channel BBE <2>
2008-09-26 23:44:32.650 INFO [Timer-01: [Primary Thread] All data found for site NNS1U channel c4s <2>
2008-09-26 23:44:32.728 INFO [Timer-01: [Primary Thread] All data found for site NNS1U channel c5s <2>
2008-09-26 23:44:32.759 INFO [Timer-01: [Primary Thread] All data found for site NNS1U channel c6s <2>
2008-09-26 23:44:32.791 INFO [Timer-01: [Primary Thread] All data found for site ILAU channel BBZ <25>
2008-09-26 23:44:32.822 INFO [Timer-01: [Primary Thread] All data found for site ILAU channel BBN <26>
2008-09-26 23:44:32.853 INFO [Timer-01: [Primary Thread] All data found for site ILAU channel BBE <27>
2008-09-26 23:44:32.900 INFO [Timer-01: [Primary Thread] All data found for site ILAU channel c4s <28>
2008-09-26 23:44:32.931 INFO [Timer-01: [Primary Thread] All data found for site ILAU channel c5s <29>
2008-09-26 23:44:32.963 INFO [Timer-01: [Primary Thread] All data found for site ILAU channel c6s <30>
2008-09-26 23:44:32.994 INFO [Timer-01: [Primary Thread] All data found for site ENA1U channel BBZ <3>
2008-09-26 23:44:33.041 INFO [Timer-01: [Primary Thread] All data found for site ENA1U channel BBN <3>
2008-09-26 23:44:33.072 INFO [Timer-01: [Primary Thread] All data found for site ENA1U channel BBE <3>
2008-09-26 23:44:33.119 INFO [Timer-01: [Primary Thread] All data found for site ENA1U channel c4s <4>
2008-09-26 23:44:33.181 INFO [Timer-01: [Primary Thread] All data found for site ENA1U channel c5s <4>
2008-09-26 23:44:33.213 INFO [Timer-01: [Primary Thread] All data found for site ENA1U channel c6s <4>
2008-09-26 23:44:33.228 INFO [Timer-01: [SUDS_DESCRIPTOR] SUDS file created: 20080926_234300_TES1.s
2008-09-26 23:44:33.228 INFO [Timer-01: [Primary Thread] Data file write finished, will run again aft
72 sec

```

```

SMARTGen.bsh
2009-11-08 20:41:00.914 INFO [Timer-0]: [Primary Thread] Data file write finished, will run again after: 59.94 sec
2009-11-08 20:42:00.868 INFO [Timer-0]: [Secondary Thread] Starting data timestamp = 2009-11-08 20:26:00.0
2009-11-08 20:42:00.877 INFO [Timer-0]: [Secondary Thread] No data found for site NV01 channel SHZ (1)
2009-11-08 20:42:00.877 INFO [Timer-0]: [Secondary Thread] No data found for site NV01 channel BDF (2)
2009-11-08 20:42:00.878 INFO [Timer-0]: [Secondary Thread] No data found for site NV02 channel SHZ (3)
2009-11-08 20:42:00.878 INFO [Timer-0]: [Secondary Thread] No data found for site NV02 channel BDF (4)
2009-11-08 20:42:00.878 INFO [Timer-0]: [Secondary Thread] No data found for site NV03 channel SHZ (5)
2009-11-08 20:42:00.878 INFO [Timer-0]: [Secondary Thread] No data found for site NV03 channel BDF (6)
2009-11-08 20:42:00.879 INFO [Timer-0]: [Secondary Thread] No data found for site NV04 channel SHZ (7)
2009-11-08 20:42:00.879 INFO [Timer-0]: [Secondary Thread] No data found for site NV04 channel BDF (8)
2009-11-08 20:42:00.879 INFO [Timer-0]: [Secondary Thread] No data found for site NV31 channel BHZ (9)
2009-11-08 20:42:00.879 INFO [Timer-0]: [Secondary Thread] No data found for site NV31 channel BBN (10)
2009-11-08 20:42:00.880 INFO [Timer-0]: [Secondary Thread] No data found for site NV31 channel BHE (11)
2009-11-08 20:42:00.880 INFO [Timer-0]: [Secondary Thread] No data found for site NV11 channel SHZ (12)
2009-11-08 20:42:00.880 INFO [Timer-0]: [Secondary Thread] No data found for site NV11 channel SHN (13)
2009-11-08 20:42:00.880 INFO [Timer-0]: [Secondary Thread] No data found for site NV11 channel SHE (14)
2009-11-08 20:42:00.881 INFO [Timer-0]: [Secondary Thread] No data found for site NV05 channel SHZ (15)
2009-11-08 20:42:00.881 INFO [Timer-0]: [Secondary Thread] No data found for site NV06 channel SHZ (16)
2009-11-08 20:42:00.881 INFO [Timer-0]: [Secondary Thread] No data found for site NV07 channel SHZ (17)
2009-11-08 20:42:00.881 INFO [Timer-0]: [Secondary Thread] No data found for site NV08 channel SHZ (18)
2009-11-08 20:42:00.882 INFO [Timer-0]: [Secondary Thread] No data found for site NV09 channel SHZ (19)
2009-11-08 20:42:00.882 INFO [Timer-0]: [Secondary Thread] No data found for site NV10 channel SHZ (20)
2009-11-08 20:42:00.884 INFO [Timer-0]: [Secondary Thread] All data found for site NVMX channel LWS (21)
2009-11-08 20:42:00.887 INFO [Timer-0]: [Secondary Thread] All data found for site NVMX channel LWD (22)
2009-11-08 20:42:00.889 INFO [Timer-0]: [Secondary Thread] All data found for site NVMX channel LKO (23)
2009-11-08 20:42:00.891 INFO [Timer-0]: [Secondary Thread] All data found for site NVMX channel LIO (24)
2009-11-08 20:42:00.894 INFO [Timer-0]: [Secondary Thread] All data found for site NVMX channel LDA (25)
2009-11-08 20:42:00.894 INFO [Timer-0]: Writing CSS data...
2009-11-08 20:42:00.896 INFO [Timer-0]: [Secondary Thread] Data file write finished, will run again after: 59.958 sec

```

Figure 4. SMARTGen window, output in SUDS (top) or CSS 3.0 (bottom) format.

For SUDS files, the file naming convention is:

YYYYMMDD_hhmmss_network.ext

Where: YYYY = year, MM = month, DD = day of month, hh = hour, mm = minute, ss = second of the first sample in the data file, network = network name, ext = file extension (e.g.'sud').

The same information as displayed in the console window is also output to log files in the 'logs' folder (e.g. c:\SMARTGeoHub\SMARTGen\logs).

The maximum number of log files to keep, and their maximum size is set in the configuration file c:\SMARTGeoHub\SMARTGen\conf\log4j.properties; the parameters to set

are `log4j.appender.drfa.MaxBackupIndex` (e.g. to 100) and `log4j.appender.drfa.MaxFileSize` (e.g. to 1000KB), respectively.

The SMARTGen configuration parameters can be changed by editing the configuration file `viewer-opts.conf`, located e.g. at:

`c:\SmartGeoHub\SMARTGen\conf\viewer-opts.conf`

The parameter file settings of interest are:

`suds.FileFormat=css;` Defines the file format parameter: `suds`, `css` or `simple` for SUDS, CSS3.0 or simple format (as described in §3.2.4.2).

`suds.interval=60;` Defines the time interval, in seconds, for querying the database and writing the raw data to files. For SUDS and simple format, this represents the duration of the data files, while for the CSS 3.0 format it represents the interval at which the CSS 3.0 database is updated with new data. Example implies 60 seconds (minimum 1 second). For CSS 3.0 recommended value is 60 seconds.

`suds.delay=10;` Defines the time delay versus real time, in seconds, to query the local database and write the raw data to files. Set this parameter to account for communication delays, data frame size in the SMART-24 digitizer, and differences between local PC clock time and SMART-24 time. Example implies 10 seconds.

`suds.FileStart=minute;` Defines whether the first data file starts on the even minute or immediately when SMARTGen is started. Allowed values are 'minute' and 'immediate'. Example implies file write starts on the even minute.

`suds.SudRealFilePath=C:\rawdata;`

`suds.SudNetworkFilePath=C:\rawdata;`

Defines the location of the folder to store the output data files on the local host computer. Example implies `c:\rawdata`.

`suds.ConfigFilePath=C:\SmartMessages;`

Defines the location of the folder to store configuration files for use of SMARTQuake® (optional).

For CSS format only:

`suds.cssDaysToKeep=3;` Defines number of days to retain in the CSS database. Example implies 3 days, data older than 3 days being automatically removed.

`suds.cssSegmentLength=14400;`

Defines length of CSS segments in seconds. Example implies 14,400 seconds, or 4 hours. Valid range is 3,600 to 86,400 seconds (1 hour to 1 day).

`suds.DatabaseFilenameFormat=%A%y_%d_%h;`

Defines the format of the database filename. Example implies the filename consists of network (array) name followed by year, “_” (as

separator), day of year, “_”, and hour (by default). Keeping the default setting is recommended.

suds.SegmentFilenameFormat=/.w/%y/%d/%A%y_%d_%h%m_%S_%C.w;

Defines the format of the segment filename. Example implies the segment files are located in a subfolder of the data folder called “.w”, in a separate subfolder of “.w” named by the year, and in a separate subfolder of the year folder named by the day of the year. The filename consists of network (array) name followed by year, “_”, day of year, “_”, hour, minute, “_”, site name, “_”, component name, and the extension “.w”. (by default). Keeping the default setting is recommended.

When used together with the automated earthquake processing software SMARTQuake®, SMARTGen relays configuration messages between SMARTGeoHub® and SMARTQuake®. This enables SMARTQuake® to reconfigure itself automatically based on configuration changes made by the operator in the SMARTGeoHub® database.

3.1.6 SMARTCal Calibration Module

The SMARTCal program can be used to automatically process calibrations and store calibration results. The results can be manually reviewed by the operator and validated using the interactive SMARTCalUpdate program.

SMARTCal program should only be started after SMARTServer is already running. Typically both SMARTServer and SMARTCal applications (and optionally other programs as well, such as SMARTGen) are started automatically by running the StartAll shortcut. Otherwise SMARTCal can be started by double-clicking on the SMARTCal shortcut that points to the following program file, e.g. on a Linux system:

```
~/SMARTGeoHub/SMARTCal/bin/SMARTCal.bsh
```

When started, SMARTCal is continuously searching for active calibrations in progress on any of the network's digitizers; if no calibrations are found the program is idle, not performing any operations. In such case, SMARTCal program's console window is as shown in Figure 5.

SMART-24 calibrations can be controlled, set up and initiated using SMART24Config software (see SMART24Config manual for complete details). For example, Figure 6 shows SMART24Config configuration window for both sine wave and random binary calibrations on a short-period (Geotech Model GS-13) sensor. Note that in parallel with the calibration sensor output recorded on the selected digitizer channel (relay state set to 'sensor'), the calibrator output signal, input to the sensor in calibration mode, is required to be recorded on another digitizer channel (relay state set to 'loopback'). If available, an unused digitizer channel should be set for loopback. It is required that both sensor and loopback channels are enabled for sending real-time data to the SMARTServer (see SMART24Config Real Time Data Properties window, Figure 15).

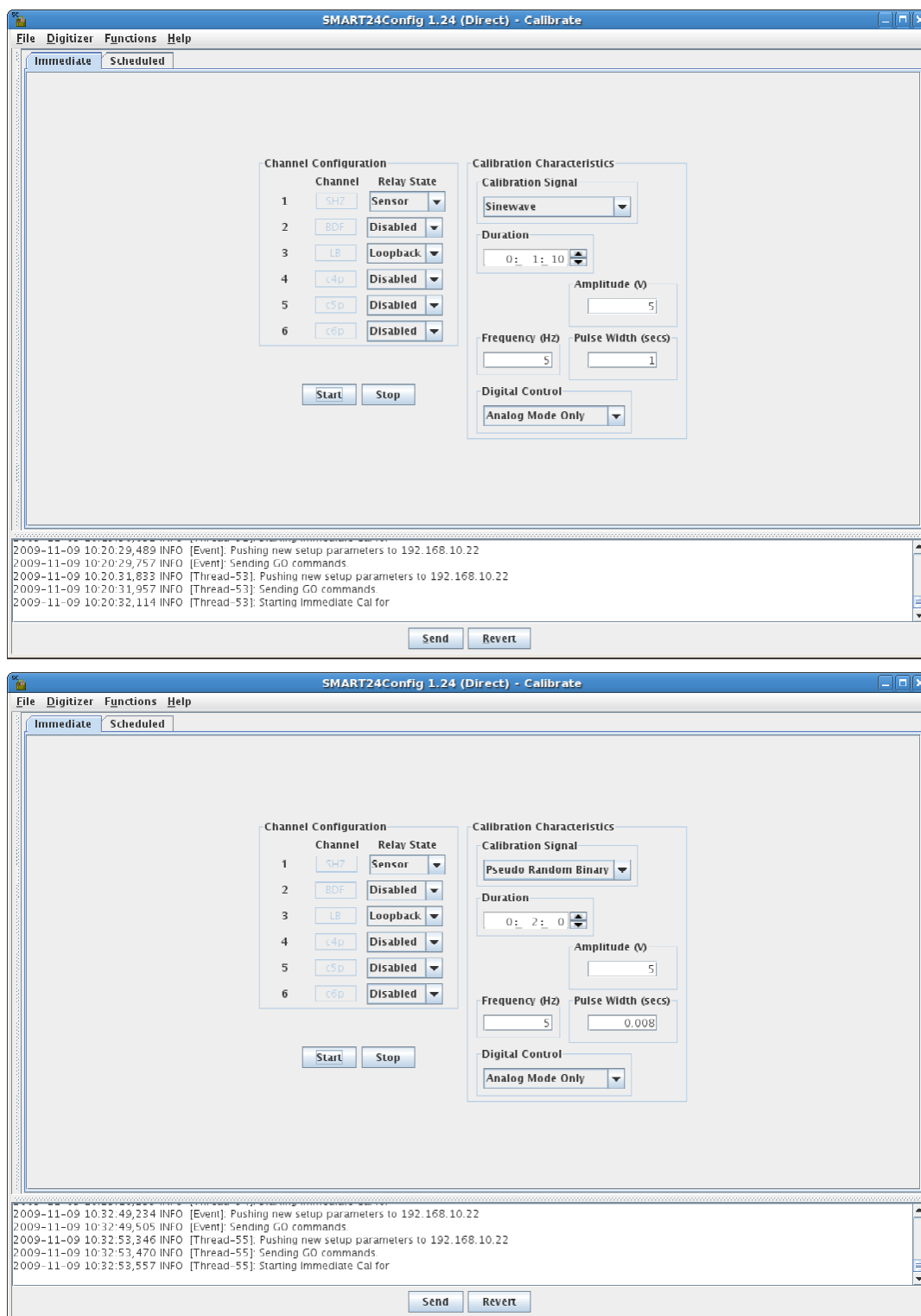


Figure 6. SMART24Config window for Sine Wave (top) and Pseudo Random Binary (bottom) calibrations

- ~/SMARTGeoHub/SMARTCal/bin/rbcal.def, a text file that contains the maximum and minimum frequencies for the frequency response, for each spectral band, where the designator in the first column (“band code”) should match the first character in the channel

name; otherwise the default values of 100 Hz and 0.005 Hz are used. The program limits the maximum frequency by 99% of Nyquist frequency. The number of frequency values between the minimum and maximum frequency (100 to 10,000, by default 100) and the indication whether to use equal frequency spacing between values in linear (1) or logarithmic (0) scale (by default logarithmic) is also set in this control file.

- ~/SMARTGeoHub/SMARTCal/bin/digitizer.fap, optionally, a text file containing the digitizer instrument response function as the number of frequency values followed by three columns for frequency (in Hz), normalized amplitude, and phase (in degrees). If this file is present, SMARTCal software outputs the full system (sensor+digitizer) calibration response function; otherwise only the sensor calibration response function if output.

Figure 7 and Figure 8 show the SMARTCal program window during processing a sine wave calibration, and random binary calibration, respectively.

```

Terminal
File Edit View Terminal Tabs Help
Searching for calibrations...
Calibration detected: NV01.cal
SITE=NV01
START=20091109 16:20:20.000
STOP=20091109 16:22:00.000
CHAN=SHZ CAL=00
CHAN=LB CAL=01
Site: NV01
Start: 20091109162020
Duration: 100
LB Channel: LB
Channel: SHZ Type0
Channels: NV01:SHZ,NV01:LB
Running: sh ./SMARTGenReq.bsh 20091109162020 100 NV01:SHZ,NV01:LB 20091109_16202
0_NV01
PC-SUD5 Generator started at: 2009-11-09 10:23:17.401
2009-11-09 16:23:17,562 INFO [main]: Database UTC time at start: 2009-11-09 16:23:17.0
2009-11-09 16:23:17,568 INFO [main]: Time of first file start: 2009-11-09 16:20:20.0
Starting reminder...
2009-11-09 16:23:17,591 INFO [main]: Initializing CSS Producer
2009-11-09 16:23:17,595 INFO [main]: CSS database directory is /home/realtime/caldata
2009-11-09 16:23:17,595 INFO [main]: CSS segment length is 3600 seconds
2009-11-09 16:23:17,595 INFO [main]: CSS time tear is +/- sampling period * 0.5
2009-11-09 16:23:17,596 INFO [main]: CSS waveforms and tables will be retained for 0 days
2009-11-09 16:23:17,596 INFO [main]: CSS database name format is '20091109_162020_NV01'
2009-11-09 16:23:17,596 INFO [main]: CSS waveform segment filename format is './w/%y/%d/%A%y %d %h%m %S %C.w'
2009-11-09 16:23:17,596 INFO [main]: CSS selected sites:channels to output are NV01:SHZ NV01:LB
2009-11-09 16:23:17,693 INFO [Timer-0]: [Primary Thread] Starting data timestamp = 2009-11-09 16:20:20.0

Terminal
File Edit View Terminal Tabs Help
2009-11-09 16:23:17,848 DEBUG [Timer-0]: [Primary Thread] Accrued execution time: 0.154 sec
2009-11-09 16:23:17,848 INFO [Timer-0]: [Primary Thread] Data file write finished

Processing Calibration: ./sincal /home/realtime/caldata/20091109_162020_NV01.wfdisc NV01:SHZ NV01:LB -MA -D50000
Reading /home/realtime/caldata/20091109_162020_NV01.wfdisc
Reading /home/realtime/caldata/./w/2009/313/NVAR2009_313_1600_NV01_LB.w
Reading /home/realtime/caldata/20091109_162020_NV01.wfdisc
Reading /home/realtime/caldata/./w/2009/313/NVAR2009_313_1600_NV01_SHZ.w
Reading last cal file./cal/20091016_0136.cal
Reading last dat file./cal/20091016_0136_NV01__SHZ.dat
Writing calibration results to ./tmp/20091109_1620.cal
Writing calibration data to ./tmp/20091109_1620_NV01__SHZ.dat
Normalization factor applied = 1.0044783
Writing calibration results to ./tmp/20091109_1620_NV01__SHZ.txt
Calibration Constant : 0.000002 N/A

Velocity Channel Sensitivity : 1986. Volts/(meter/sec)

System Sensitivity : 0.001640 nm/count

Calibration Period : 0.200 sec

Searching for calibrations...

```

Figure 7. SMARTCal window processing a sine wave calibration

```

2009-11-09 16:37:13,563 INFO [Timer-0]: [Primary Thread] No data found for site NV05 channel LB (30)
2009-11-09 16:37:13,564 INFO [Timer-0]: [Primary Thread] No data found for site NV06 channel LB (31)
2009-11-09 16:37:13,568 INFO [Timer-0]: [Primary Thread] No data found for site NV07 channel LB (32)
2009-11-09 16:37:13,569 INFO [Timer-0]: [Primary Thread] No data found for site NV08 channel LB (33)
2009-11-09 16:37:13,570 INFO [Timer-0]: [Primary Thread] No data found for site NV09 channel LB (34)
2009-11-09 16:37:13,571 INFO [Timer-0]: [Primary Thread] No data found for site NV10 channel LB (35)
2009-11-09 16:37:13,571 INFO [Timer-0]: Writing CSS data...
2009-11-09 16:37:13,571 DEBUG [Timer-0]: New CSS segment
2009-11-09 16:37:13,577 DEBUG [Timer-0]: Open waveform file /home/realtime/caldata/./w/2009/313/NVAR2009_313_1600_NV01.SH
2009-11-09 16:37:13,590 DEBUG [Timer-0]: Open waveform file /home/realtime/caldata/./w/2009/313/NVAR2009_313_1600_NV01_LB
2009-11-09 16:37:13,616 DEBUG [Timer-0]: [Primary Thread] Execution time: 0.168 sec
2009-11-09 16:37:13,616 DEBUG [Timer-0]: [Primary Thread] Accrued execution time: 0.168 sec
2009-11-09 16:37:13,616 INFO [Timer-0]: [Primary Thread] Data file write finished

Processing Calibration: ./rbcal /home/realtime/caldata/20091109_163240_NV01.wfdisc NV01:SHZ NV01:LB -MA
Reading /home/realtime/caldata/20091109_163240_NV01.wfdisc
Reading /home/realtime/caldata/./w/2009/313/NVAR2009_313_1600_NV01.SHZ.w
Reading /home/realtime/caldata/20091109_163240_NV01.wfdisc
Reading /home/realtime/caldata/./w/2009/313/NVAR2009_313_1600_NV01_LB.w
Reading last cal file../cal/20091109_1620.cal
Reading last dat file../cal/20091109_1620_NV01_SHZ.dat
Writing calibration results to ../tmp/20091109_1632.cal
Writing calibration data to ../tmp/20091109_1632_NV01_SHZ.dat

```

Figure 8. SMARTCal window processing a random binary calibration

After calibration processing is finished, SMARTCal goes back to waiting for calibrations.

The calibrations and their results are stored to a list file (`~/SMARTGeoHub/SMARTCal/bin/smartcal_update.list`) and a temporary calibration results folder (`~/SMARTGeoHub/SMARTCal/tmp`) where for each calibration following data files are output (see Figure 11):

- calibration text (.TXT) file with Sensor Sensitivity, System Sensitivity (`calib`) and Calibration Period (`calper`) as resulted from a sine wave calibration (see Figure 7) of a particular site/channel. The name of the file contains the calibration date and time, site and channel name, as: `YYYYMMDD_hhmm_SITEN_CHN.txt`, where: `YYYY` = year, `MM` = month, `DD` = day of month, `hh` = hour, `mm` = minute, `SITEN` = site name (5 characters, if shorter padded with ‘_’ to 5 characters), `CHN` = channel name (3 characters), and ‘`txt`’ = file extension.

- calibration data (.DAT) file with the instrument’s transfer function as resulted from a random binary calibration of a particular site/channel. The name of the file is the same as for the text file, with the extension ‘`dat`’. This file contains the normalized station calibration as three columns for frequency (in Hz), normalized amplitude, and phase (in degrees). The calibration function contains values at a number of frequencies within a frequency interval set in file `rbcal.def`. If the digitizer reponse is provided in file “`digitizer.fap`” then the full system (sensor+digitizer) response function is stored in the .DAT file, otherwise only the sensor response.

- calibration (.CAL) report file with sensitivity values for all network sites and channels at a given calibration date and time. The name of the file contains the calibration date and time, as: `YYYYMMDD_hhmm.cal`, where: `YYYY` = year, `MM` = month, `DD` = day of month, `hh` = hour, `mm` = minute, and ‘`cal`’ = file extension.

Calibration processing is performed automatically by SMARTCal. To update the calibration values used on the system in real-time operation and validate a particular set of automatic calibration results, the operator must manually run the SMARTCalUpdate program.

Start the SMARTCalUpdate by double-clicking on the SMARTCalUpdate shortcut that points to the following program file, e.g. on a Linux system:

```
~/SMARTGeoHub/SMARTCal/bin/CalUpdate.bsh
```

SMARTCalUpdate GUI is started, prompting the operator to select the calibration to save from the list of calibrations performed since the last system update, as shown in Figure 9.

For each calibration, displayed information includes date and time of the calibration, calibrated site and channel name, and calibration values (calib, calper and sensor sensitivity) resulted from a sine wave calibration, or the label 'Random Binary' to designate a random binary calibration.

Once the operator selects a calibration from the list and presses the Save Calibration button, a confirmation window is displayed as in Figure 10.



Figure 9 SMARTCalUpdate Select Window with a sine wave calibration results (top) and random binary calibration results (bottom)

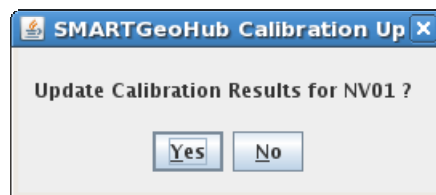


Figure 10. SMARTCalUpdate confirmation window

After operator confirmation (by selecting 'Yes' in the confirmation window) the following operations are performed automatically:

- calibration results are updated on the local computer, by moving the three calibration results files (.TXT, .DAT and .CAL) from the temporary calibration results folder (~\SMARTGeoHub\SMARTCal\tmp) to the archive calibration results folder (~\SMARTGeoHub\SMARTCal\cal) shown in Figure 11. An information message like in Figure 12 is displayed. At the same time, all older pending (unsaved) calibrations, performed at an earlier date and time than the saved one, are deleted from the temporary calibration results folder, and an information message as shown in Figure 13 is displayed.

- when saving a sine wave calibration, calib, calper and sensor sensitivity parameters are updated on the remote SMART-24 configuration, and the SMART-24 will start using the new calibration values when sending out CD1.1 data in real-time. An information message as shown in Figure 14 is displayed. The updated SMART-24 parameters can be inspected by connecting to the remote SMART-24 site using SMART24Config software, and checking the Real Time Data Properties window as shown in Figure 15.

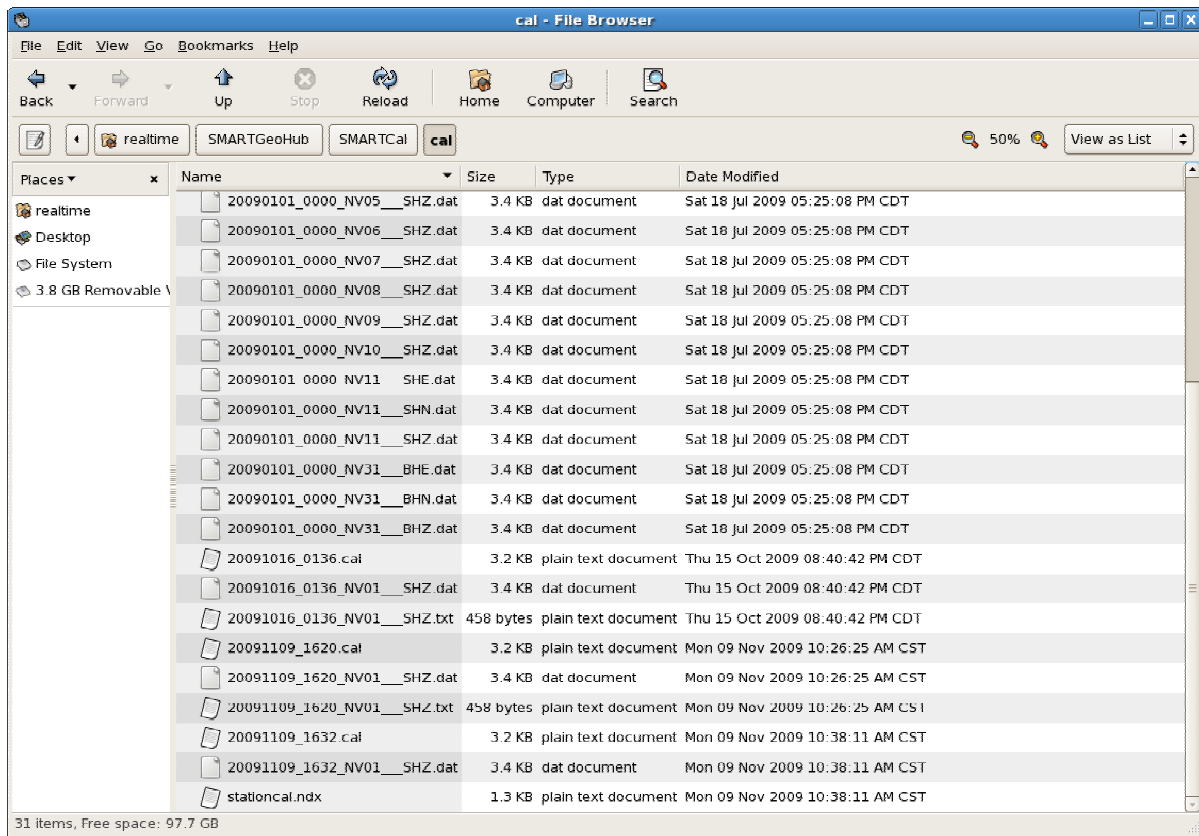


Figure 11. SMARTCal and SMARTCalUpdate calibration files



Figure 12. SMARTCalUpdate local results updated window

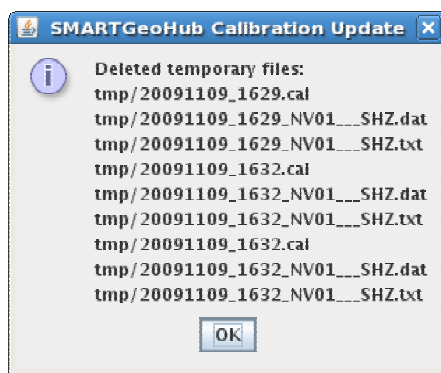


Figure 13. SMARTCalUpdate cleanup window

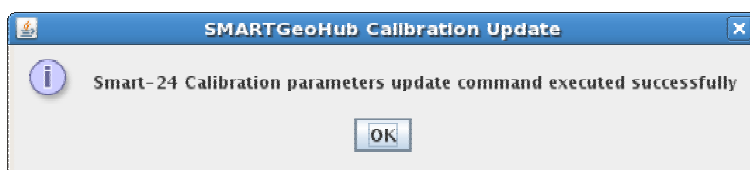


Figure 14. SMARTCalUpdate SMART-24 updated window

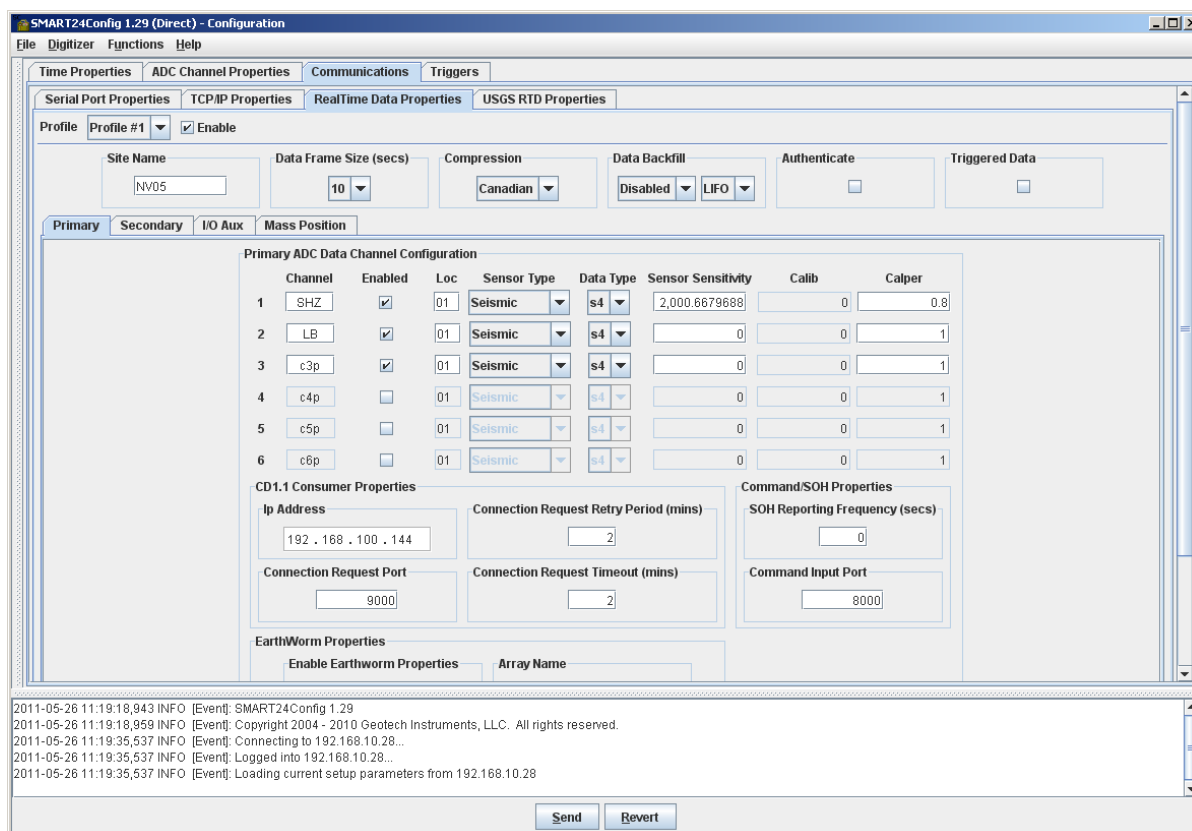


Figure 15. SMART-24 Real Time Data Properties window with calibration parameters

3.2 SMARTGeoHub® for Large Networks

The large database ('flat file') model is used on large SMARTGeoHub® systems involving more than 100 channels. In such applications, SMARTServer writes raw data using a structure of data files (instead of MySQL data tables) to optimize efficiency in accessing large quantities of data.

3.2.1 SMARTGeoHub® 'Flat File' Data Server

Specific to this 'flat file' model, SMARTServer and other SMARTGeoHub® clients read system settings from a file named 'geotech.properties', typically located in the user's home folder (e.g. c:\Documents and Settings\RTSmart on a Windows XP system, or c:\Users\RTSmart on a Windows Vista or Window 7 system).

The 'geotech.properties' parameter file settings of interest are:

primary.registry=localhost:1099

Defines the IP address and port number (separated by ':') of the computer where the SMARTServer is located. Example implies localhost computer, port number 1099.

data.file.directory=c://tw//data.files

Defines the location where the 'flat files' are stored. Example implies folder c:\tw\data.files. Defaults to geotech\data.files subfolder of the user's home folder (e.g. c:\Documents and Settings\RealTimeUser\geotech\data.files)

data.file.offset.mode=false

Defines whether the data are sampled on the even second or with a non-zero time offset relative to the second. For SMART-24 instruments data are always sampled on the even second and this parameter should be set to false (as in this example).

station.name=CWB-BH

Defines the station (or network) name used by the SMARTGeoHub®. Example implies CWB-BH.

The 'flat file' SMARTServer can be started by double-clicking on the SMARTServer shortcut that points to the following program file:

c:\SMARTGeoHub\SmartServerFF\bin\runServer.bat

The 'Earthworm direct' SMARTServer (no writing to the local files, only sending data directly to Earthworm) can be started by double-clicking on the SMARTServer shortcut that points to the following program file:

c:\SMARTGeoHub\SmartServerFF\bin\runServerEwDirect.bat

Figure 16 shows examples of the 'flat file' SMARTServer running window, at start-up and during normal operation.

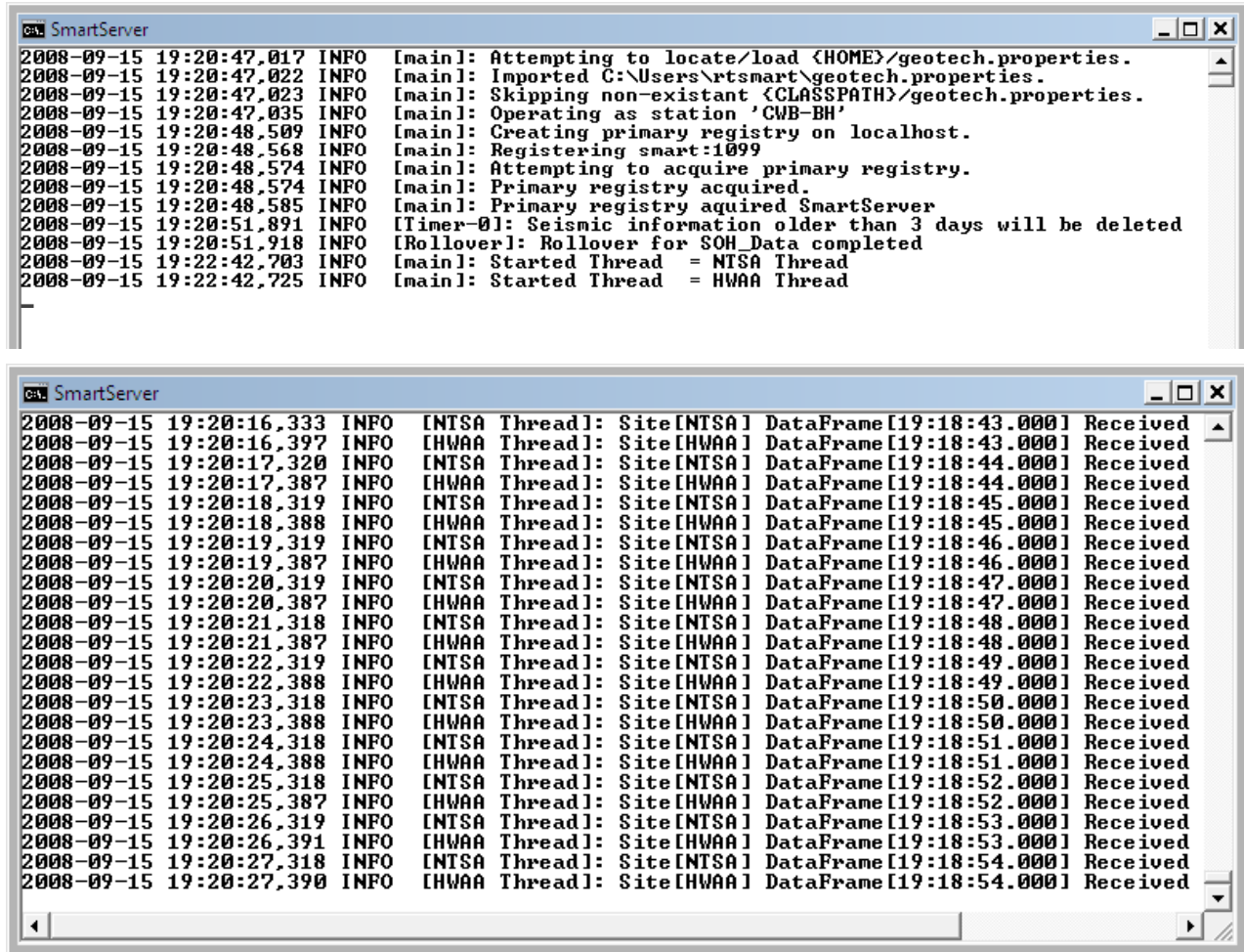


Figure 16. SMARTServer 'flat file' at start-up (top) and in normal operation (bottom)

3.2.2 Database Configuration File

When SMARTGeoHub® is using the large database model, the system database structure is created from a configuration file written in XML format, as described below. Typically, this file (named e.g. DB.xml) is located under the c:\SMARTGeoHub\GeoDB folder.

The XML configuration file contains all the parametric information to be entered in the database: the names of the station (network), sites, and channels (components) with their respective IP addresses for real-time communications, and other site specific parameters, e.g. geographical coordinates and instrument parameters.

The 'DB.xml' parameter file settings of interest are:

- | | |
|----------------------|---|
| clear.tables="false" | Defines whether to clear or not database tables when updating the database structure from the configuration file. Default is false, do not clear database structure. |
| drop.tables="false" | Defines whether to delete and then recreate database tables when updating the database structure from the configuration file. Default is false, do not drop database structure. |

duration="259200000"	Defines the duration of each data channel 'flat file', given in milliseconds. Data older than this time interval will be overwritten. Example implies 3 days.
station name="CWB-BH"	Defines the station (or network) name used by the SMARTGeoHub®. Example implies CWB-BH.
ip="192.168.100.55"	Defines the IP address of the host computer where SMARTGeoHub® resides. This is the IP address where the SMART-24 digitizer will send data to.

For each SMART-24 digitizer, a section is contained in the configuration file with all its relevant site parameters. The site section starts with `<site` and ends with `</site>` delimiters. Site parameters of interest are:

site name="NTSA"	Defines the name given to the SMART-24 digitizer. Each SMART-24 would have a unique site name.
ip="192.168.100.21"	Defines the IP address of the SMART-24 digitizer. Each SMART-24 would have a unique IP address.
data.port="11656"	Defines the data port address that SMARTServer will assign the SMART-24 digitizer for data reception. Each SMART-24 would have a unique data port assignment. The recommended parameter values are 10000 + SMART-24 serial number. Thus the example implies serial number 1656.
longitude="20.0"	Defines the longitude of the respective site geographical location.
latitude="40.0"	Defines the latitude of the respective site geographical location.
elevation="1000.0"	Defines the elevation of the respective site geographical location.

For each SMART-24 data channel, a section is contained in the configuration file with all its relevant channel parameters. Every site section includes one or more channel sections. The channel section starts with `<channel` and ends with `</>` delimiters. Channel parameters of interest are:

channel name="ABZ"	Defines the name given to the SMART-24 digitizer data channel.
location.name="02"	Defines the location name given to the SMART-24 digitizer data channel. Each SMART-24 data channel would have a unique combination of channel and location name.
motion.type="A"	Defines the motion type for that particular data channel. 'A' is acceleration, 'V' is velocity.
lsb="1.6"	Defines the SMART-24 digitizer bit weight, given in units of microvolts/digital count.
sensitivity="0.25"	Defines the sensor sensitivity for the respective data channel, given in units of volts/m/s or volts/m/s ² for velocity or acceleration sensors, respectively.
period="100."	Defines the sensor natural period in seconds.
damping="0.707"	Defines the sensor damping in fractions of critical damping.

broadband="true" Defines the type of sensor, broadband or not (e.g. short period).

3.2.3 Database Configuration Operation

Database configuration can be done either to create for the first time the database structure, or to update the database.

If running, all SMARTGeoHub® programs must be stopped before configuring the database (e.g. by using the StopAll command, see §3.2).

The XML configuration file can be easily changed by using a text editor (e.g. Notepad), to add, modify, or delete elements (see §3.3.2). The use of the configuration file allows easy backup of the database parametric information.

Next steps are to open a command prompt console window, change folder to c:\SMARTGeoHub\GeoDB, and execute the command:

```
Db-import db.xml
```

This command configures the SMARTGeoHub® database with all the parametric information contained in the XML configuration file. If the clear.tables and drop.tables parameters in the XML configuration file are set to 'true', the 'flat files' should be deleted from their location (see §3.2.1) before starting data acquisition tasks.

After the database structure is created, SMARTDBConfig user-friendly interface can be used to inspect the configuration. Then, SMARTGeoHub® data acquisition can be started, e.g. by running the SMARTServer.

3.2.4 FileGen Operation

The FileGen program can be used to write raw data in near real-time as data files of fixed duration, for the use of an automated processing software (e.g. SMARTQuake®). The data files can be formatted using either SUDS format (Banfill, 2003) or a so-called 'simple' format, consisting of a small header followed by the binary data values (see § 3.3.4.2). Other formats such as CSS, SAC or miniSeed are possible on request – consult factory for details.

FileGen program should only be started after SMARTServer is already running. Typically both SMARTServer and FileGen applications are started automatically by running the `StartAll` shortcut. Otherwise FileGen can be started individually by double-clicking on the FileGen shortcut that points to the following program file:

```
c:\SMARTGeoHub\FileGen\bin\FileGen.bat.
```

The time when the FileGen program is started (immediately, or at next even minute) marks the start time of the first file. The data are extracted from the SMARTGeoHub® database periodically, with a user programmable delay versus real-time to allow for communications delays, and written to data files. Each data file contains all data channels recorded by the SMARTGeoHub® for the operator specified recording time period.

The file naming convention is:

```
YYYYMMDD_hhmmss_network.ext
```

Where: `YYYY` = year, `MM` = month, `DD` = day of month, `hh` = hour, `mm` = minute, `ss` = second of the first sample in the data file, `network` = network name, `ext` = file extension (e.g. 'sim').

FileGen relies on the local computer time to determine when to read and write data. Because of this reason, the time of the host computer should be close (within reasonable accuracy) to the SMART-24 digitizers' time. It is strongly recommended for the host computer's clock to be automatically synchronized to a time server (either a Internet time server, or local).

When used together with the automated earthquake processing software SMARTQuake®, FileGen relays configuration messages between SMARTGeoHub® and SMARTQuake®. This enables SMARTQuake® to reconfigure automatically based on configuration changes made by the operator in the SMARTGeoHub® database.

FileGen logs the site and channels names, the data file name and its start date and time in the program's console window (shown as an example in Figure 17), and to log files stored in a log folder (e.g. `c:\SMARTGeoHub\FileGen\logs`).

The maximum number of log files to keep, and their maximum size is set in the configuration file `c:\SMARTGeoHub\FileGen\conf\log4j.properties`; the parameters to set are `log4j.appender.drfa.MaxBackupIndex` (e.g. to 100) and `log4j.appender.drfa.MaxFileSize` (e.g. to 1000KB), respectively.

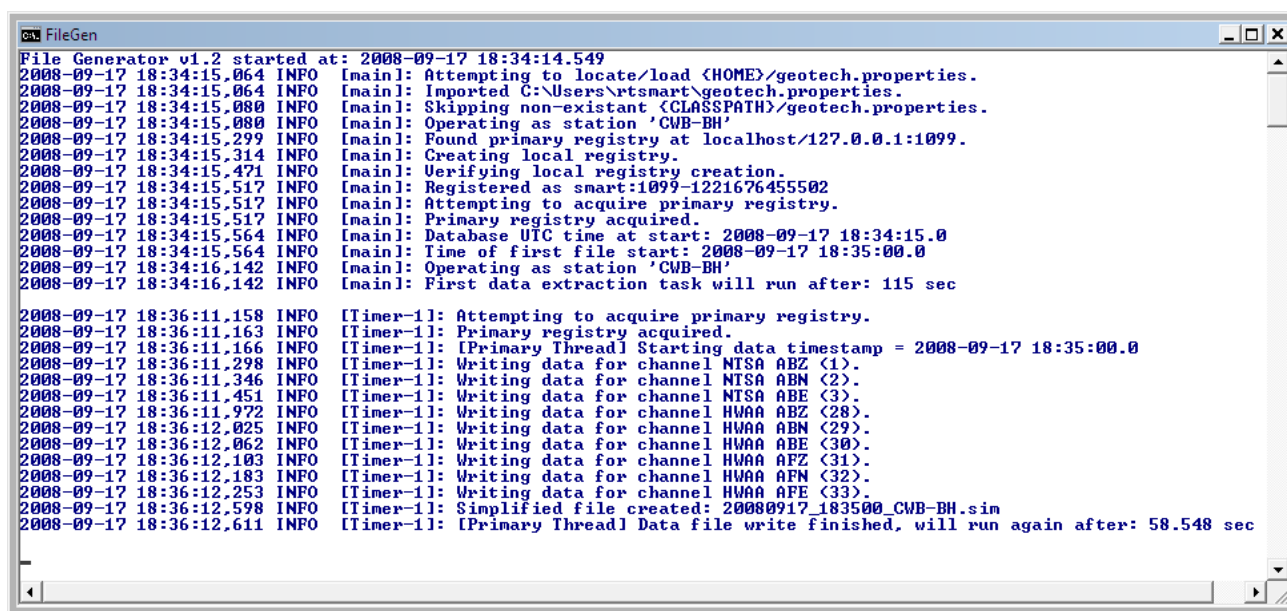


Figure 17. FileGen program window.

3.2.4.1 FileGen Settings

The FileGen configuration parameters should be changed by editing the configuration file `viewer-opts.conf`, located e.g. at:

`c:\SmartGeoHub\FileGen\conf\viewer-opts.conf`

The parameter file settings of interest are:

- `suds.interval=60;` Defines the duration of the output data files, in seconds. Example implies 60 seconds (minimum 1 second).
- `suds.delay=10;` Defines the time delay versus real time, in seconds, to write raw data to files. Example implies 10 seconds.
- `suds.FileStart=minute;` Defines whether the first data file starts on the even minute or immediately when FileGen is started. Allowed values are 'minute' and 'immediate'. Example implies file starts on the even minute.
- `suds.FileFormat=simple;` Defines the file data format: 'suds' for SUDS format, or 'simple' for a simple data format as described below. Example implies simple format.
- `suds.SudRealFilePath=C:\rawdata;` Defines the location of the folder to store the output data files on the local host computer. Example implies `c:\rawdata`.
- `suds.FileExtension=sim;` Defines the data files extension. Example implies 'sim'.
- `suds.ConfigFilePath=C:\SmartMessages;` Defines the location of the folder to store configuration files for use of SMARTQuake® (optional).

3.2.4.2 The Simple Data Format

The data file consists of a succession of channel records, one for each active data channel recorded by the SMARTGeoHub®.

The channel record consists of a header followed by a data block.

The channel header has the following structure, described using the seismological convention (see About This Document):

char network[8];	Network name, same for all sites recorded at the the SMARTGeoHub® (8 characters).
char station[5];	The SMART-24 digitizer site name (5 characters).
char component[3];	The SMART-24 digitizer channel name (3 characters).
double begintime;	Data trace begin time, given as number of seconds elapsed since January 1 st , 1970 at 00:00:00.0, stored as an 8 byte real number.
long length;	Number of samples in trace, a 4 byte integer number.
float rate;	Sample rate, a 4 byte real number.

Data values follow the header, stored as 4 byte signed integers.

3.3 SMARTGeoHub® Data Interface to Earthworm

The SMARTServer module also provides functionality for exporting SMARTGeoHub® real-time data to Earthworm.

The SMARTGeoHub® SMARTServer application can be used to accept incoming real-time data connections from SMART-24 instruments via a TCP/IP network connection, to write the received data to the local database, and/or to send the received data via a TCP/IP network connection to Earthworm. Data of all SMART-24 sites can be sent to the same Earthworm server, or different subsets of sites to different destinations.

Using SMARTGeoHub® together with a subset of Earthworm modules, as described in this chapter, makes the system even more flexible, so that all, or any subset of SMART-24 sites and components, raw data or filtered channels, can be sent to multiple Earthworm systems. A block diagram of the SMARTGeoHub® with Earthworm extensions is shown in Appendix A, Figure 19.

The SMARTServer outputs Earthworm waveform messages for data channels with specified Station, Component, Network and (optional) Location names (SCN/SCNL), and transfers them to Earthworm Import module(s). SMARTServer is compatible with trace data messages of Earthworm releases V7.x and prior (V6.3).

The SMARTServer acts like an Earthworm Export module. Data originated from each SMART-24 instrument are sent to a separate Earthworm Import module, using a multi-threaded, parallel processing mechanism to improve data exchange efficiency.

Data connection is initiated by the SMARTServer, while the Earthworm Import module is passive, waiting for the connection.

3.3.1 SMARTServer Configuration Notes

SMARTServer (Server to SMART-24 digitizer(s) and client to Earthworm) must be configured before attempting to run initially. The SMARTServer configuration parameters specific to Earthworm should be changed by editing the configuration file `viewer-opts.conf`, located e.g. at:

```
c:\SmartGeoHub\SmartServerFF\conf\viewer-opts.conf.
```

The parameter file settings of interest are:

[Station_Info]

`number.ofsites=12;` Defines the number of SMART-24 digitizers reporting. Example implies 12 sites. In this case entries for all 12 sites must be present in the [Station_Info] section, similar to the examples given below for the first site.

`station.network=192.168.0.55;` Defines the IP address of the host computer where SMARTGeoHub® resides.

`site.name.1=NTSA;` Defines the name given to the SMART-24 digitizer. Example implies site number one. Each SMART-24 would have a unique name and would be identified numerically sequential, i.e. `site.name.2=...` `site.name.n=...`, where `n= number.ofsites`. This

	name must match the station name in Earthworm Station/ Component/ Network/ Location (SCNL) names.
site.network.1=TW;	Defines the name of the network of SMART-24 digitizers. Each SMART-24 would have a unique name and would be identified numerically sequential, i.e. site.network.2=... site.network.n=..., where n= number.oftsites. This name must match the network name in Earthworm Station/ Component/ Network/ Location (SCNL) names.
site.ip.1=192.168.0.21;	Defines the IP address of the SMART-24 digitizer. The example implies site number one. Each SMART-24 would have a unique IP address, and would be identified numerically sequential, i.e. site.ip.2=... site.ip.n=..., where n=number.oftsites.
site.port.1=11656;	Defines the data port address that SMARTServer will assign the SMART-24 digitizer for data reception. Example implies site number one. Each SMART-24 would have a unique data port assignment and would be identified numerically sequential, i.e. site.port.2=... site.port.n=..., where n= number.oftsites. The recommended parameter values are 10000 + SMART-24 serial number. Thus the example implies serial number 1656.
site.respframeip.1=192.168.0.55;	Defines the destination IP address where the SMART-24 digitizer data frames are sent to, i.e. the IP address of the SMARTGeoHub® computer as seen from the SMART-24 site. Example implies site number one. In most cases this parameter is the same for all sites and same as station.network parameter, except in networks with address translation (NAT) and a mixture of SMART-24 digitizers both local and external to the SMARTGeoHub® computer's network.
site.msgmod.1=61;	Defines the Earthworm Module ID. A value of zero is defined to be a wildcard (MOD_WILDCARD) in file earthworm.d. The module ID is part of the logo of the Earthworm messages output by SMARTServer. In case more than one SMART-24 digitizers are reporting, each SMART-24 would have a unique Earthworm Module ID assignment and would be identified numerically sequential, i.e. site.msgmod.2=... site.msgmod.n=..., where n=number.oftsites. Each site Module ID must be registered in Earthworm in the working copy of the file earthworm.d, e.g. by adding the following entry:

Module MOD_EXPORT_NTSA 61

...

Note! Section [Station_Info] continues on, to define entries for all other sites numbered from 2 to number.oftsites. It is followed by:

[Eathworm_Info]

- earthworm.ena=1; Defines whether the SMARTServer data export for Earthworm is enabled ('1') or disabled ('0'), for all SMART-24 digitizers reporting.
- earthworm.direct=0; Defines whether the SMARTServer Earthworm 'direct' mode is enabled ('1') or disabled ('0'), for all SMART-24 digitizers reporting. In Earthworm 'direct' mode, SMARTServer only sends data to Earthworm without recording the data to SMARTGeoHub® flat files. Default is '0' where the data are both saved to SMARTGeoHub® and sent to Earthworm.
- earthworm.serverip.1=192.168.0.55; Defines the IP address of host computer for the Earthworm Import Server. Each SMART-24 defined in the [Station_Info] section should have an assignment. The actual IP address may be the same for all SMART-24s or it could be different for each, depending on the operational environment. Typically it would be the same. As above, the parameter names are indexed numerically sequential. This IP address must match the ReceiverIpAdr setting from Earthworm's import parameter file.
- earthworm.severport.1=16001; Defines the port number of the Earthworm Import Server that will receive the SMART-24 data stream via the SMARTServer. Each SMART-24 should have a unique port assignment. As above, the parameter names are indexed numerically sequential. This port number must match the ReceiverPort setting from Earthworm's import parameter file.
- earthworm.enasta.1=1; Defines whether the SMARTServer data export for Earthworm is enabled ('1') or disabled ('0') for each SMART-24. As above, the parameter names are indexed numerically sequential.

...

Note! Section [Earthworm_Info] continues on, to define entries for all other sites numbered from 2 to number.ofsites. It is followed by:

- earthworm.msg.instid=52; Defines the Earthworm Installation ID for all sites data. The Earthworm installation ID is part of the logo of the Earthworm messages output by SMARTServer. Any Earthworm installation ID value which is defined in file earthworm_global.d can be used: either the value of 255 defined to be unknown (INST_UNKNOWN), or one corresponding to a specific installation. This example implies 52, INST_CWB.
- earthworm.msg.type=19; Defines the type of Earthworm message. Allowed values are 19 (TYPE_TRACEBUF2) or 20 (TYPE_TRACEBUF), as defined in file earthworm_global.d. The message type is part of the logo of the Earthworm messages output by SMARTServer. SMARTServer formats the Earthworm trace data messages according to the message type: for TYPE_TRACEBUF2 it uses Station/ Component/ Network/ Location (SCNL) names compatible with Earthworm V7.x releases, and for TYPE_TRACEBUF it uses

Station/ Component/ Network (SCN) names compatible with Earthworm V6.x. Example implies 19, TYPE_TRACEBUF2.

earthworm.latency.limit=1000; Defines the time window (in microseconds) set as a latency limit in providing data to Earthworm, for logging purposes only. If this time limit is exceeded for the SmartServer to deliver its data to EarthWorm, SmartServer will log a warning message. Example implies 1000ms (1 second).

3.3.2 Earthworm Configuration Notes

To interface Earthworm with the SMARTGeoHub® Server, the following Earthworm modules must be started and running. A configuration tool named SmartEWConfig is provided for easily setting up the parameter files needed for SMARTGeoHub® and Earthworm to operate together (see §3.3.3).

3.3.2.1 Import_gen_pasv

This module's function is to accept data via the TCP/IP network and output them to Earthworm transport ring buffer. For every SMART-24 instrument data source there should be one instance of the import module configured via its own parameter file (e.g. import_gen_pasv.d) and started using Earthworm StartStop mechanism.

The list of import processes to be started, together with their corresponding parameters files and startup parameters should be entered in Earthworm parameter file startstop_nt.d, e.g. by adding the following entries:

Process	"import_gen_pasv import_gen_pasv.d1"
PriorityClass	Normal
ThreadPriority	Normal
Display	NoNewConsole

Import_gen_pasv recommended parameter file settings are:

MyModuleId	MOD_IMPORT_GEN_PASV
------------	---------------------

Set the name of the Module ID. The Module ID must be registered with Earthworm in the working copy of the file earthworm.d, e.g. by adding the following entry:

Module	MOD_IMPORT_GEN_PASV	50
--------	---------------------	----

RingName	WAVE_RING
----------	-----------

Set transport ring buffer name (RingName) to WAVE_RING.

ReceiverIpAdr	192.168.0.55
---------------	--------------

ReceiverIpAdr must contain the local IP address of the Earthworm server assigned to receive data from the SMART-24 instrument.

ReceiverPort	16001
--------------	-------

ReceiverPort must contain the local port number of the Earthworm server assigned to receive data from the SMART-24 instrument. Default port number is 16005. To receive data in Earthworm from multiple SMART-24 sites, unique port numbers have to be set in

the respective Import_gen_pasv parameter files, and as many instances of the import module as SMART-24s have to be started in Earthworm.

SenderIpAdr 192.168.0.55

SenderIpAdr is the IP address of the computer where the SMARTGeoHub® Server is running.

MyAliveInt 0

SenderHeartRate 0

Disable heartbeat messages between the Sender and Receiver by setting parameters SenderHeartRate and MyAliveInt to 0.

SocketTimeout 200000

SocketTimeout has to be larger than SMART-24 data frame size (e.g. of 10 seconds, or 10,000 miliseconds); recommended value is 200,000 miliseconds.

3.3.2.2 Fir (optional)

This module's function is to filter the raw data using a Finite Impulse Response (FIR) filter. Raw data are read from Earthworm transport ring buffer, and results are output to the same Earthworm transport ring buffer. Depending on the desired filter parameters, there can be one instance of the Fir module, or multiple instances, each Fir module configured via its own parameter file (e.g. fir.d) and started using Earthworm StartStop mechanism.

In the example described below, raw data are filtered by using a band pass filter obtained by applying successively first a high pass filter, and then a low pass filter. One instance of the Fir module is started for the high pass, and another for the low pass filter, identified by the names MOD_FIR_HP, MOD_FIR_LP, and configured from the parameter files fir_hp.d and fir_lp.d, respectively.

The list of Fir processes to be started, together with their corresponding parameters files and startup parameters should be entered in Earthworm parameter file startstop_nt.d, e.g. by adding the following entries:

```
Process      "fir fir_hp.d"
PriorityClass Normal
ThreadPriority Normal
Display      NoNewConsole
```

```
Process      "fir fir_lp.d"
PriorityClass Normal
ThreadPriority Normal
Display      NoNewConsole
```

High pass filter Fir module recommended parameter file settings (in file fir_hp.d) are:

MyModId MOD_FIR_HP

Set the name of the Module ID. The Module ID must be registered with Earthworm in the working copy of the file earthworm.d, e.g. by adding the following entry:

```
Module MOD_FIR_HP 55
```

InRing WAVE_RING

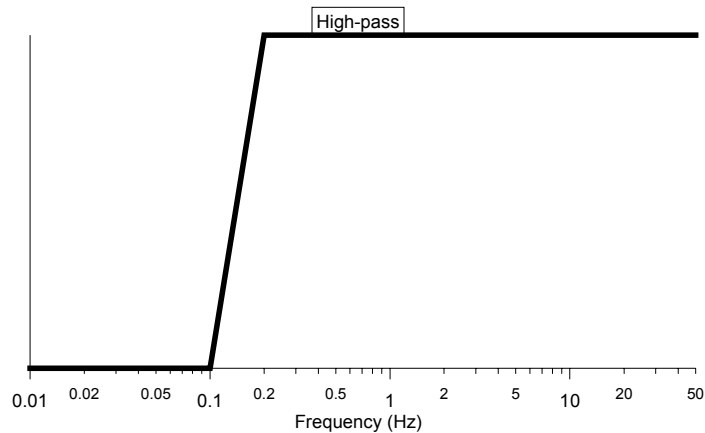
Set the name of the transport ring buffer to read data from.

OutRing WAVE_RING

Set the name of the transport ring buffer to write data to.

Band	0.0	0.0005	0	0.03	(stop band)
Band	0.004	1.0	1	0.03	(pass band)

The above two lines set the parameters of the frequency bands that define the filter: low frequency, high frequency, band level (0 for stop band, 1 for pass band) and amount of ripple allowed from desired level. Frequencies are scaled to the range 0.0 -1.0, where 1.0 corresponds to the Nyquist frequency. Example implies: (first line) a stop band at all frequencies below 0.0005*Nyquist frequency (e.g. 0.025Hz if data are sampled at 100sps), and (second line) a pass band at frequencies above 0.004*Nyquist frequency (e.g. 0.2Hz if data are sampled at 100sps), with the filter frequency response as shown below.



GetWavesFrom INST_WILDCARD MOD_WILDCARD

Specifies the logos of the Earthworm messages to be read from the input transport ring. Example implies messages with any logos (any Installation ID, any Module ID) are read.

GetSCNL NTSV BBZ TW 02 NTSV SFZ TW 03

List Station/ Component/ Network/ Location (SCNL) codes of trace messages to filter and their output SCNL codes. SCNL codes for all traces to be filtered must be specified using GetSCNL lines. Example implies waveforms for Station NTSV, Component BBZ, Network TW, Location 02, are filtered to output traces with names: Station NTSV, Component SFZ, Network TW, Location 03. For the filtered channels both channel and location codes are changed, while station and network are the same. These filtered traces will be sent to the low pass filter described below.

Low pass filter Fir module recommended parameter file settings (in file fir_lp.d) are:

MyModId MOD_FIR_LP

Set the name of the Module ID. The Module ID must be registered with Earthworm in the working copy of the file earthworm.d, e.g. by adding the following entry:

Module MOD_FIR_LP 54

InRing WAVE_RING

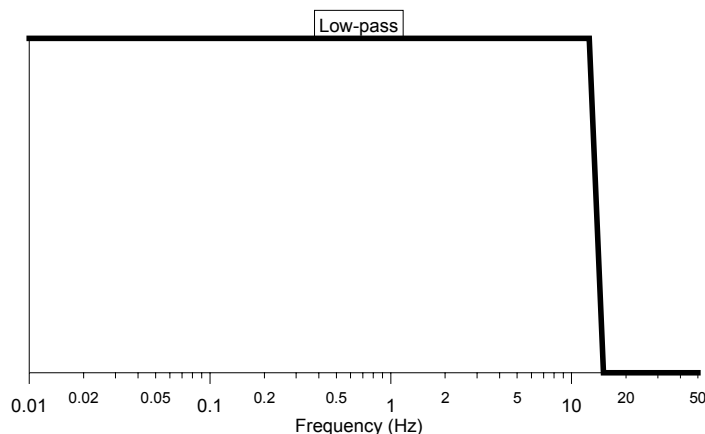
Set the name of the transport ring buffer to read data from.

OutRing WAVE_RING

Set the name of the transport ring buffer to write data to.

Band	0.0	0.25	1	0.005	(pass band)
Band	0.3	1.0	0	0.005	(stop band)

Set the parameters of the frequency bands that define the filter: low frequency, high frequency, band level (0 for stop band, 1 for pass band) and amount of ripple allowed from desired level. Frequencies are scaled to the range 0.0 -1.0, where 1.0 corresponds to the Nyquist frequency. Example implies: (first line) a pass band at all frequencies below 0.25*Nyquist frequency (e.g. 12.5Hz if data are sampled at 100sps), and (second line) a stop band at frequencies above 0.3*Nyquist frequency (e.g. 15 Hz if data are sampled at 100sps)), with the filter frequency response as shown below.



GetWavesFrom INST_WILDCARD MOD_WILDCARD

Specifies the logos of the Earthworm messages to be read from the input transport ring. Example implies messages with any logos (any Installation ID, any Module ID) are read.

GetSCNL NTSV SFZ TW 03 NTSV SFZ TW 02

List Station/ Component/ Network/ Location (SCNL) codes of trace messages to filter and their output SCNL codes. SCNL codes

for all traces to be filtered must be specified using GetSCNL lines. Example implies waveforms for Station NTSV, Component SFZ, Network TW, Location 03 (that are output by the high pass filter described above), are filtered to output traces with same SCNL names but Location name 02. These filtered output traces are the end product of applying a band pass filter, e.g. between 0.2 Hz - 12.5Hz for data sampled at 100sps.

3.3.2.3 Export_scnl_ack (optional)

This module's function is to export data via the TCP/IP network and output them to other Earthworm systems, using the corresponding import_ack modules running on the remote Earthworm systems. Selected data channels, defined by their Station/Component/Network/Location names, can be specified for export. For every destination Earthworm system there should be one instance of the export module configured via its own parameter file (e.g. export_scnl_ack.d) and started using Earthworm StartStop mechanism.

The list of export processes to be started, together with their corresponding parameters files and startup parameters should be entered in Earthworm parameter file startstop_nt.d, e.g. by adding the following entries:

```
Process      "export_scnl_ack export_scnl_ack.d"
PriorityClass Normal
ThreadPriority Normal
Display      NewConsole
```

Export_scnl_ack recommended parameter file settings are:

```
MyModuleId    MOD_EXPORT_SCNL_ACK
```

Set the name of the Module ID. The Module ID must be registered with Earthworm in the working copy of the file earthworm.d, e.g. by adding the following entry:

```
Module  MOD_EXPORT_SCNL_ACK  52
```

```
RingName      WAVE_RING
```

Transport ring buffer name (RingName) used for input. Example implies WAVE_RING.

```
GetMsgLogo    INST_CWB  MOD_WILDCARD  TYPE_TRACEBUF2
```

Specifies the logos of the Earthworm messages to export to client systems. Example implies logo with Installation ID INST_CWB, any Module ID and type TYPE_TRACEBUF2.

```
ServerIPAddr  192.168.0.55
```

ServerIPAddr must contain the local IP address of the Earthworm server assigned to listen for connections from client systems.

```
ServerPort    16050
```

SereverPort must contain the well-known port number of the Earthworm server assigned to export messages on. Example implies port number is 16050. To export data to different

Earthworm clients, unique port numbers have to be set in the respective Export_scnl_ack parameter files, and as many instances of the export module as client systems have to be started in Earthworm.

Send_scnl_remap NTSA * TW 02 NTS * * *

List Station/ Component/ Network/ Location (SCNL) codes of trace messages to export and their output SCNL codes. SCNL codes of all traces to be exported must be specified using Send_scnl_remap lines (wildcards are permitted). SCNL codes are changed on the fly. Example implies NTSA Station names are changed to NTS, while Component, Network and Location names are kept the same.

3.3.2.4 Wave_serverV (optional)

Its function to get the data from the transport ring buffer, write tank structures and send data to a wave viewer (Earthworm WaveViewer or Swarm). The wave_serverV module is configured via its own parameter file (e.g. wave_serverV.d) and started using the Earthworm StartStop mechanism.

The wave_serverV process to be started, together with its corresponding parameter files and startup parameters should be entered in Earthworm parameter file startstop_nt.d, e.g. by adding the following entries:

Process	"wave_serverV wave_serverV.d"
PriorityClass	Normal
ThreadPriority	Normal
Display	NoNewConsole

Wave_ServerV recommended parameter file settings are:

RingName	WAVE_RING
----------	-----------

Transport ring buffer name (RingName) used for input. Example implies WAVE_RING.

ServerIPAdr	192.168.0.55
-------------	--------------

ServerIPAdr must contain the local IP address of the Earthworm server running the wave server.

ServerPort	16022	SereverPort must contain the port number of the Earthworm server assigned to receiving requiests and sending waveforms to the wave viewer. Example implies port number is 16022.
------------	-------	--

Tank NTSA ABZ TW 02 1024 INST_CWB MOD_WILDCARD ...

Earthworm uses Station, Component, Network and Location (SCNL) names to uniquely identify trace data. In the tank file list, configure SMART-24 SCNL names and message logos to match the Earthworm messages output by the SMARTGeoHub® Server (see its configuration settings above). Also, Earthworm SCNL

names have to match the actual SMART-24 instrument channel names configured with SMART24Config.

InputQueueLen 500 Set the size of the message queue (InputQueueLen) to the recommended value of 500 or above, in particular if necessary to accomodate a large number of data channels.

3.3.3 SmartEWConfig Configuration Tool

SmartEWConfig is a configuration tool to facilitate setting up SMARTGeoHub® - large database model - and Earthworm in order to import SMARTGeoHub® data into Earthworm. This tool reads the SMARTGeoHub® database configuration file (see §3.2.2) and generates all other parameter files for SMARTGeoHub® and Earthworm modules needed for the two systems to operate together.

Typically, SmartEWConfig is located under the `c:\SMARTGeoHub\GeoDB` folder. To run SmartEWConfig, simply execute the SmartEWConfig.bat program file from its home folder.

SmartEWConfig reads the information from the configuration XML file already configured by the user with all necessary parametric information as described in §3.2.2. It also uses the following model files for a typical installation, provided by Geotech and stored in the same home folder:

- **Model files:**

`geotech.properties_model` – model SMARTGeoHub® system settings file (see §3.2.1)

`viewer-opts.conf_model` – model SMARTServer configuration file (see §3.3.1)

`earthworm_global.d` – Earthworm global configuration file

`earthworm.d_model` – model Earthworm configuration file

`export_scn1_ack.d_model` – model Earthworm export configuration file

`import_gen_pasv.d_model` – model Earthworm import configuration file

`startstop_nt.d_model` – model Earthworm start and stop configuration file

`wave_serverV.d_model` – model Earthworm wave server configuration file

When running SmartEWConfig, new versions of the above configuration files are generated based on the existing model files and the actual system configuration as read from the SMARTGeoHub® XML file. The following files are written out, and copied to the respective SMARTGeoHub® or Earthworm system folders, overwriting the existing files:

- **SMARTGeoHub®:**

`geotech.properties` – SMARTGeoHub® system settings file, copied to the SMARTGeoHub home folder (typically `c:\SMARTGeoHub`) and user's folder (under `c:\Documents and Settings`)

`viewer-opts.conf` – SMARTServer configuration file, copied to the SMARTServer folder (typically `c:\SMARTGeoHub\SmartServerFF\conf`)

- **Earthworm:**

`earthworm_global.d` – Earthworm global configuration file (same as input)

`earthworm.d` – Earthworm configuration file

`export_scn1_ack.d` – Earthworm export configuration file

`import_gen_pasv.d` – Earthworm import configuration file

`startstop_nt.d` – Earthworm start and stop configuration file

`wave_serverV.d` – Earthworm wave server configuration file

All the above Earthworm parameter files are copied to the Earthworm parameters home folder (typically `c:\Earthworm\run\params`).

The SMARTGeoHub® and Earthworm system folders used by SmartEWConfig for outputting the configuration files are set to the typical values as given above. For changes from the typical setup contact Geotech.

For data communication between the SMART-24 instruments and the SMARTServer, data port addresses of $10000+n$ are used on the SMARTGeoHub® computer, where n is the site number, from 1 to the total number of sites, in the order in which the sites are enumerated in the configuration XML file.

For data import from SMARTGeoHub® into Earthworm, data port addresses of $16000+n$ are used on the SMARTGeoHub®/Earthworm computer, where n is the site number, from 1 to the total number of sites, in the order in which the sites are enumerated in the configuration XML file.

For importing SMARTGeoHub® data into Earthworm, the **Import_gen_pasv** Earthworm module is used, and registered in the `earthworm_global.d` file with the module ID of 60. For each SMART-24 site, the SMARTServer data export to Earthworm is registered in the `earthworm_global.d` file with the module ID of $60+n$, where n is the site number, from 1 to the total number of sites, in the order in which the sites are enumerated in the configuration XML file.

For changes from this typical setup, run SmartEWConfig from a Command Prompt window, in its home folder, with the command line:

```
SmartEWConfig -input InputFile -installation InstallationName -  
smartPort Smart24Port -ewPort EarthwormPort -ewMod ModuleID
```

Where:

`InputFile` – Name of the input configuration file, by default DB.XML

`InstallationName` – Earthworm installation name (registered in `earthworm_global.d` file), by default INST_UNKNOWN

`Smart24Port` – Starting data port address used by SMARTServer to receive SMART-24 data, by default 10000

`EarthwormPort` – Starting data port address used for Earthworm import from SMARTGeoHub®, by default 16000

`ModuleID` – Starting module ID used for Earthworm import from SMARTGeoHub®, by default 60

Note:

SmartEWConfig has a special provision for SMARTGeoHub® working in Earthworm ‘direct’ mode, i.e. SMARTServer only sends data to Earthworm without recording the data to SMARTGeoHub® flat files (see §3.3.1). In this case the SMARTGeoHub® configuration XML file is not required for SMARTGeoHub® to run. SmartEWConfig can then take its input from the SMARTServer configuration file instead, already configured by the user with all necessary parametric information as described in §3.3.1, by specifying in the command line the `InputFile` parameter as the name of the SMARTServer configuration file (e.g. `viewer-opts.conf_model`). The channel and location names, which are not included in the input configuration file, are assumed to be the SMART-24 defaults (e.g. ‘clp’ for channel name, ‘01’ for location name). Same set of parameter files as described above are generated as output and copied to SMARTGeoHub® and Earthworm system folders.

3.4 SMART-24 Data Interface to SeisComP/SeedLink

The SMARTSeedLink plugin is a variant of the SMARTServer module that provides functionality for importing SMART-24 real-time data into SeisComP/SeedLink.

The SMARTGeoHub® SMARTSeedLink application can be used to accept incoming real-time data connections from SMART-24 instruments via a TCP/IP network connection, and to deliver the received data to SeisComP/SeedLink.

The SMARTSeedLink plugin is distributed by Geotech as precompiled binary file. It was compiled and tested with SUSE Linux 10.3 and SeisComP 2.5.

The program is installed by copying the plugin executable file (e.g. SmartSeedLinkPlugin) to the SeisComP binaries directory (e.g. /home/sysop/seiscomp/acquisition/bin).

The SMART-24 instrument is configured using the Smart24Config software, e.g. using the following parameters:

- sample rate: 50 sps
- channel names (for first three channels): BHZ, BHN, BHE
- CD1.1 destination: IP address of the SeisComP computer
- Connection Request Port: Set a unique port number for each Smart-24 site. Recommended is to start with the default value (9000) and increment it by one for each Smart-24 site added to the system.

Depending on the sample rate and channel names used, changes may be required in the SeisComP stream processing scheme, defined in the configuration file `~/seiscomp/acquisition/config/streams.xml`.

When configuring the station in SeisComP, following steps are recommended:

- Run the SeisComP configure utility (`seiscomp config`) to add a new station, with:
 - Station code: SMART-24 site code (e.g. S1140)
 - Station description: any unique description (e.g. S1140_Station)
 - Type of data source: 9 (corresponding to the DR-24 digitizer)
 - Stream processing scheme: e.g. `stream_50`, for 50 sps sample rate

For all others settings choose the default answers.

Add other new SMART-24 stations, as many as needed, using the same procedure.

Exit and save SeisComP configuration.

- Open with a file editor the SeisComP configuration file `~/seiscomp/acquisition/config/seedlink.ini`. It contains a plugin entry line for each new station added.

Change each station entry, e.g.:

from

```
plugin dr24_S1140 cmd="/home/sysop/seiscomp/acquisition/bin/serial_plugin -v -f /home/sysop/config/plugins.ini"
```

to:

```
plugin smart24_S1140 cmd="/home/sysop/seiscomp/acquisition/bin/SmartSeedLinkPlugin  
/home/sysop/config/SmartSeedLinkPlugin_S1140.conf"
```

Keep the default values for the plugin parameters:

```
timeout = 600
```

```
start_retry = 60
```

```
shutdown_wait = 10
```

For each station entry the plugin name must be the same (e.g. `~/seiscomp/acquisition/bin/SmartSeedLinkPlugin`), but calls as argument a different station configuration file. It is recommended that the name of the station configuration file includes station name, e.g.:

```
~/config/SmartSeedLinkPlugin_S1140.conf
```

- Copy the sample station configuration file distributed by Geotech `SmartSeedLinkPlugin.conf` to the configuration directory (e.g. `~/config`).

For each SMART-24 site create a copy of the station configuration file with the name as specified in the plugin command line argument, e.g.:

```
~/config/SmartSeedLinkPlugin_S1140.conf
```

Edit the file and choose the following recommended settings:

Set to the number of SMART-24 sites connected to one instance of the plugin to 1, e.g.:

```
number.ofsites=1;
```

Enter in the appropriate field the IP address of the SeisComp computer, e.g.:

```
seiscomp.server=192.168.100.100;
```

Enter in the appropriate field the port of the `SmartSeedLinkPlugin`. This value must match the Connection Request Port set on the respective SMART-24, e.g.:

```
seiscomp.server.port=9000;
```

Enter in the appropriate fields the SMART-24 site name and the data stream port number. To ensure port number is unique, set its value to 10000 + the SMART-24 serial number, e.g.:

```
site.name.1=S1140;
```

```
site.port.1=11140;
```

Enable or disable SeisComp logging of incoming data frames by setting the `verbose` field to true or false:

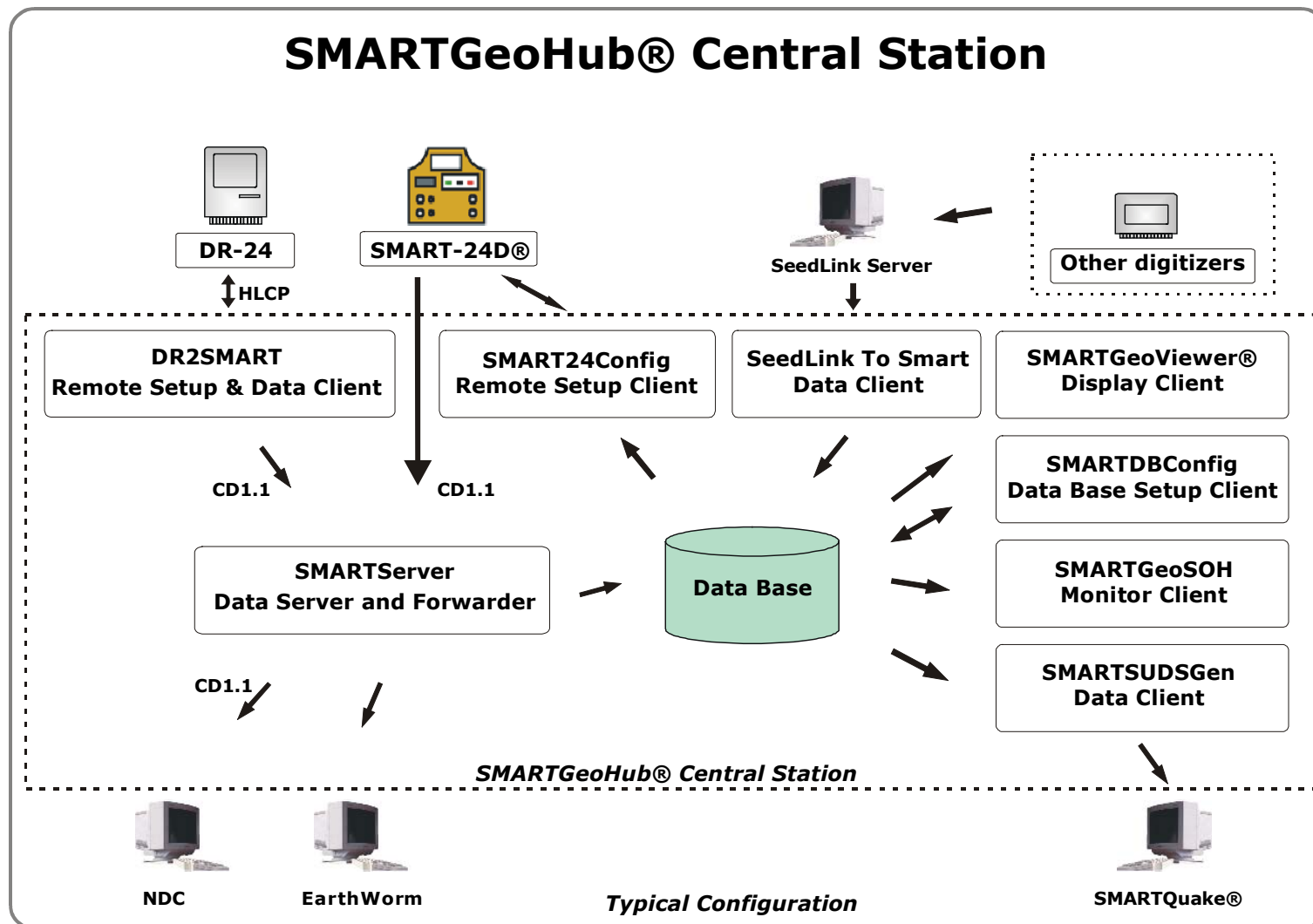
```
verbose.data=true;
```

Check the settings of the Linux firewall (if active) to allow access on the data ports used by the `SmartSeedLinkPlugin`.

- Start SeisComp to start data acquisition for the new SMART-24 station(s). Allow at least two minutes for the digitizer to connect and start sending data to the server. Verify status of incoming data using SeisComp's data viewer and log files.

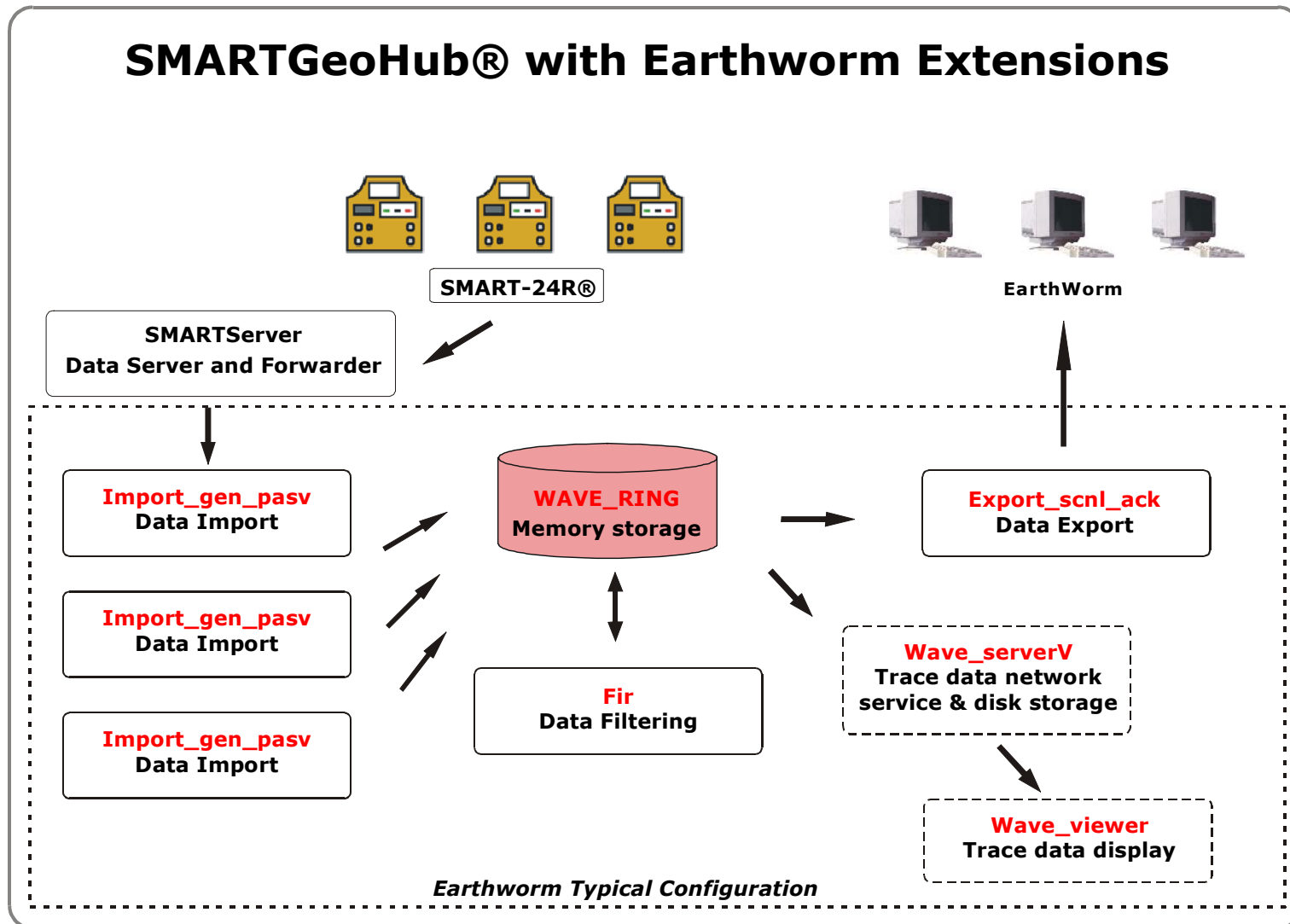
4 Appendix A.

Figure 18. SMARTGeoHub® Central Station data flow diagram.



(Page Intentionally Left Blank)

Figure 19. SMARTGeoHub® with Earthworm extensions data flow diagram



(Page Intentionally Left Blank)

5 Communication and Data Security Management

5.1 Installation of the Communication and Data Security Management Software

The Communication and Data Security Management Software is installed as part of the SMART Central Station software. The install procedure is simple and fully automated, and consists in running the Install program from the Geotech's SMARTGeoHub® CD, that copies all files necessary to run the SMART Central Station software on your computer.

If the workstation is running Windows operating system, the Pragma Systems' Console Telnet client, part of Pragma Fortress Client Suite is required for the data security management application SMARTKeyCommand to run. Please refer to Pragma Systems website (www.pragmasys.com) and documentation for instructions on installing the Pragma Fortress Client Suite software, and during installation use the default options.

Depending on the workstation operating system environment, Windows or Linux, and on the type of cryptology card used to secure data communications, Mykotronx or Spyrys Fortezza card or Safenet Luna PCM token, different programs are employed for key management control at the station (workstation) and sites (SMART-24 digitizers), as indicated in the following table. Please refer to the corresponding section of this manual which is applicable for your specific hardware and software environment according to Table

Table 3 Key management programs used and corresponding section in the manual, in function of the software and hardware environment.

If workstation operating system is...	If the workstation cryptocard is...	And the SMART-24 cryptocard is...	Use this program for station (workstation) key management	Use this program for site (SMART-24) key management
Windows	Fortezza	Fortezza	SMARTKeyCommand (see §5.6)	SMARTKeyCommand (see §5.6)
Windows	Safenet Luna PCM	Fortezza	StationKeyCommand (see §5.5)	SMARTKeyCommand (see §5.6)
Linux	Safenet Luna PCM	Fortezza	StationKeyCommand (see §5.5)	StationKeyCommand (see §5.5)

5.2 Initialization of the Fortezza Card(s) for the Workstation

Part of the setup required for the Infrasound or Seismic system is to properly initialize the Fortezza Cryptology card(s) used to secure the communications between users and the system.

Typically, two Fortezza cards are used on the Workstation: First one, installed on the Workstation in the Fortezza card reader socket 1 (top position on the card reader), is used to secure communications for the CD1.1 data transmitted from the station to the NDC/IDC. The second one, installed in the Workstation in the Fortezza card reader socket 2 (bottom position on the card reader), is used to secure communications for the AutoDRM message exchange between the station and the NDC/IDC.

In each case, setup of the Fortezza card requires that information be entered into a data file and then running the appropriate initialization program.

Ensure that all active windows (programs) are shutdown and closed before continuing with this procedure.

To setup the AutoDRM messaging Fortezza card, open the file `c:\station\authd.par` using a text editor, e.g. Notepad. A representation of this file is shown below.

```
#LIST

#####
# authd server parameters
#
org=????????????????
unit=????????????????
name=autodrm
email=autodrm@????????????????????????????????????
cert-db-file=station.db
port=29000

<portions removed for brevity>

#####
# DSA parameters
#
DSA-P=95796dc5bdabb8afaf5b7bb99aceae4b12c8ecc82fb6237c5be26
DSA-Q=8aa48abd2805d4ed5f2437671d46a6d717cb1091
DSA-G=59505e9150e45bb7b96b3125e085bf9ff2a3c3439e3d
```

Fill in the fields for `org=`, `unit=` and `email=` using the appropriate information for the installation. If the field for `name=` needs to be changed (not recommended) the new name must be entered using a text editor to replace `autodrm` in the file `c:\station\station.db`. The `email=` field **MUST** be `autodrm@` the appropriate domain, corresponding to the station AutoDRM email address. Fill in the DSA parameters `P`, `Q` and `G` with values that are suitable for this installation. It may be easiest to cut and paste these values from another source.

IMPORTANT

Do **NOT** add spaces between the equal sign (=) and the value for that parameter.

Once these fields have been filled in, save the file and close notepad. It is wise to make a backup of this file for use in the event that the original becomes damaged.

Now that all the information has been entered, run the card initialization program, `RunCardInitForMsgFortezza.bat`. When this program has completed, the Fortezza card is ready for use and the output of the initialization is a self-signed x509 certificate located in `c:\station\station.pem` in PEM format, and in `c:\station\msgkey.txt` in text format. The certificate in `station.pem` will be used to sign secure communications - AutoDRM messages - from the Station. The information in the `station.pem` or

msgkey.txt file should be used to create an x509 certificate signed by a certificate authority (instead of the self-signed certificate) that will be used to sign secure communications from the Station. This certificate, in PEM format, should be copied to c:\station\station.pem.

To setup the CD1.1 data Fortezza card, open the file c:\station\authd1.par using a text editor, e.g. Notepad. A representation of this file is shown below.

```
#LIST

#####
# authd server parameters
#
org=????????????????
unit=????????????????
name=cd11
email=cd11@????????????????????????????????????
cert-db-file=station.db
port=29001

<portions removed for brevity>

#####
# DSA parameters
#
DSA-P=95796dc5bdabb8afaf5b7bb99aceae4b12c8ecc82fb6...
DSA-Q=8aa48abd2805d4ed5f2437671d46a6d717cb1091
DSA-G=59505e9150e45bb7b96b3125e085bf9ff2a3c3439e3d...
```

Fill in the fields for org=, unit= and email= using the appropriate information for the installation. If the field for name= needs to be changed (not recommended) the new name must be entered using a text editor to replace cd11 in the file c:\station\station.db. Fill in the DSA parameters P, Q and G with values that are suitable for this installation. It may be easiest to cut and paste these values from another source.

IMPORTANT

Do **NOT** add spaces between the equal sign (=) and the value for that parameter.

Once these fields have been filled in, save the file and close notepad. It is wise to make a backup of this file for use in the event that the original becomes damaged.

Now that all the information has been entered, run the card initialization program, RunCardInitForCD11Fortezza.bat. When this program has completed, the Fortezza card is ready for use and the output of the initialization is a self-signed x509 certificate located in c:\station\station1.pem in PEM format, and in c:\station\cd11key.txt in text format. The key pair generated during the initialization will be used to sign secure communications - CD1.1 data packets - from the Station.

Once all these steps are completed, double-click on the StartAll shortcut on the desktop will automatically restart all data acquisition and communication functions.

5.3 Initialization of the Fortezza Card for the SMART-24

Part of the setup required for Infrasound or Seismic system is to properly initialize the Fortezza Cryptology card used in each SMART-24. This card is used to secure the communications between users and the system. SMART-24 Fortezza cards need to be initialized **ONLY** if it is necessary to replace the original Fortezza card with a replacement.

Ensure that all active windows (programs) are shutdown and closed before continuing with this procedure.

If the workstation Fortezza initialization steps have been completed, there is no need to edit the data in `authd.par` so the first step may be skipped.

Setup of the Fortezza card requires that information be entered into a data file and then running the appropriate initialization program. Open the file `c:\station\authd.par` using a text editor, e.g. Notepad. A representation of this file is shown below.

```
#LIST

#####
# authd server parameters
#
org=????????????????????
unit=????????????????????
name=autodrm
email=autodrm@????????????????????????????????????????
cert-db-file=station.db
port=29000

<portions removed for brevity>

#####
# DSA parameters
#
DSA-P=95796dc5bdabb8afaf5b7bb99aceae4b12c8ecc82fb6...
DSA-Q=8aa48abd2805d4ed5f2437671d46a6d717cb1091
DSA-G=59505e9150e45bb7b96b3125e085bf9ff2a3c3439e3d...
```

Fill in the fields for `org=`, `unit=` and `email=` using the appropriate information for the installation. If the field for `name=` needs to be changed (not recommended) the new name must be entered using a text editor to replace `autodrm` in the file `c:\station\station.db`. The `email=` field **MUST** be `autodrm@` the appropriate domain, corresponding to the station AutoDRM email address. Fill in the DSA parameters `P`, `Q` and `G` with values that are suitable for this installation. It may be easiest to cut and paste these values from another source.

IMPORTANT

Do **NOT** add spaces between the equal sign (=) and the value for that parameter.

Once these fields have been filled in, save the file and close notepad. It is wise to make a backup of this file for use in the event that the original becomes damaged.

The Fortezza card that is to be initialized must be removed from the SMART-24 and brought to the Workstation. It is necessary to remove power from the SMART-24 prior to the card's removal.

If a Safely Remove Hardware icon is shown in the system tray on the workstation computer for the Fortezza card, double-click on this icon. This will stop the currently operating Fortezza card in the workstation. Open the workstation's PC cover and locate the Fortezza reader. Remove the AutoDRM Messaging Fortezza card from the card reader second slot (located on the bottom), taking special care to remember which Fortezza card belongs in the workstation. You will hear beeps and see a visual indication on the screen indicating that the card has been removed. Insert the Fortezza card from the SMART-24 into the same slot. You should hear beeps and see a visual indication that the card has been recognized and started. You are now ready to initialize this card

Run the card initialization program, `RunCardInitForMsgFortezza.bat`. When this program has completed, the Fortezza card is ready for use by the SMART-24. Once again, stop the card. You may now remove the card and replace it in the SMART-24. The SMART-24 will finalize the card setup when it is re-powered. Replace the Fortezza card that was removed from the workstation into its appropriate slot and double-click on the `StartAll` shortcut on the desktop to automatically restart all data acquisition and communication functions.

5.4 Configuration of Incoming Message Verifier

Upon receipt of an email, AutoDRM verifies that it has come from a trusted source by use of the sender's certificate. This certificate is stored in PEM format in the `c:\Station` directory. The certificate name indicates the user to whom the certificate belongs. As an example, if a trusted user were `george@somedomain.com` there should be a `c:\Station\george.pem` file. This certificate must be generated with information from the user's public key by a trusted signing authority. Once this PEM file has been generated, it must be copied into the `c:\Station` directory. Ensure that all active windows (programs) are shutdown and closed before continuing with this procedure. After ensuring that the programs have terminated, copy the file to the `c:\Station` directory. Once this PEM file has been copied, its presence must be noted in the verification database. To do this, open the `c:\Station\station.db` file using notepad. The contents of this file are represented below.

```
operator 1000 0 operator.pem
autodrm 2000 687 station.pem
```

Using the example above, `george@somedomain.com`, `station.db` would be edited to look like:

```
operator 1000 0 operator.pem
autodrm 2000 687 station.pem
george    3000 0 george.pem
```

Once this file has been edited and saved, double-click on the `StartAll` shortcut on the desktop to automatically restart all data acquisition and communication functions.

5.5 StationKeyCommand

StationKeyCommand is a java application that enables the operator to execute key management commands for the Safenet Luna PCM token installed at the local Workstation, and, if the Workstation's operating system is Linux, also for the Fortezza Cryptology Cards located at the SMART-24 remote sites.

StationKeyCommand can be started by double-clicking on the StatioKeyCommand shortcut that points to the following program file, e.g. on a Linux system:

```
~/StationKeyCommand/StationKeyCommand.bsh
```

StationKeyCommand program's GUI window is as shown in Figure 20.

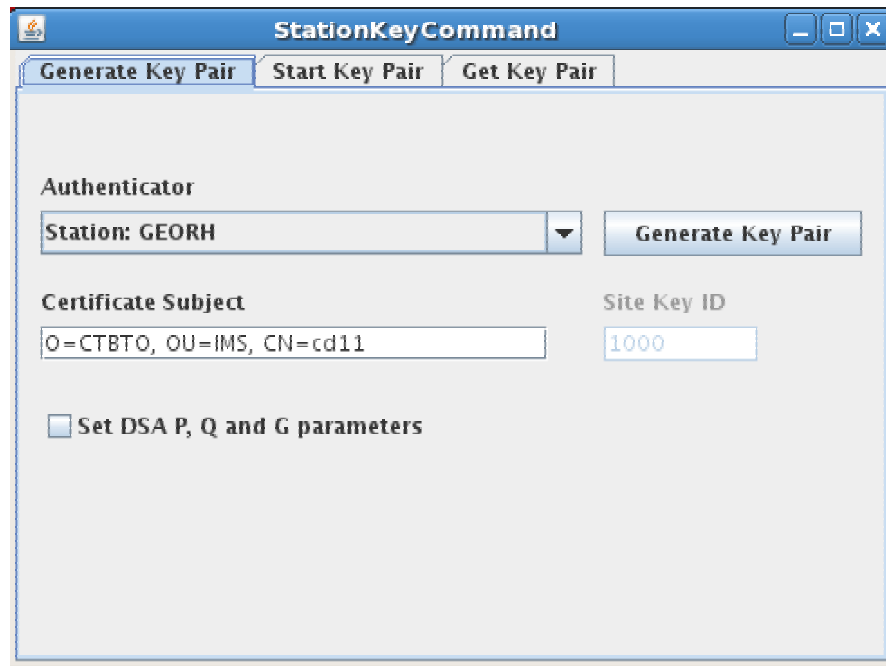


Figure 20. StationKeyCommand GUI

At startup, SMARTKeyCommand reads the station name and the list of site names with their IP addresses from the text file `~/StationKeyCommand/StationKeyCommand.ini`. The content of this file is as follows:

```
Station=NVAR
Site=NV01   Ip=192.168.10.22
Site=NV02   Ip=192.168.10.23
Site=NV03   Ip=192.168.10.24
Site=NV04   Ip=192.168.10.25
...
Site=NVMX   Ip=192.168.10.34
```

To re-configure the system, first close the StationKeyCommand application, open this file with a text editor, change the fields to reflect the correct array configuration, and then restart StationKeyCommand.

The three tabbed windows of the program provide simple access to the key management commands: Generate Key Pair, Start Key Pair and Get Key Pair

Unless otherwise noted, the output file naming convention used is as follows:

```
S(Station or Site Name)_T(hhmmss)_M(mo)_D(dom)_Y(year).xxx
```

The first character of the file name is 'C' for station CD1.1, and 'R' for the site authenticators, respectively. The file name includes the station or site name, and the date (year, month and day of month) and time (hour, minute and second) when the command was executed. The file extension is used to indicate the type of information contained in the file. For example, 'gen' stands for generated key information, 'get' for retrieved station certificate, 'pem' denotes the PEM format of the certificate.

The files generated following the three key management commands are stored in three folders, respectively:

~/StationKeyCommand/GeneratedKeys for Generate Key Pair command

~/StationKeyCommand/StationCertificates for Start Key Pair command

~/StationKeyCommand/RetrievedKeys for Get Key Pair command

The key management commands are described in detail the following sections.

5.5.1 Generate Key Pair

Select this tab to execute the **Generate Key Pair** command that generates new key pairs at the station or site authenticators. From the "Authenticator" drop-down list the user must select the Station code to denote the authentication card installed at the workstation, or, on a Linux system only, the Site code for which a new key will be generated (see Figure 20).

The certificate subject may be entered by the user in the edit text box provided, or the default setting can be used. Optional, after checking the check box "Set DSA P, Q and G Parameters", the user can enter the DSA P, Q and G parameters to be used when generating the key (typically by copying the ASCII string from the source and paste it into the corresponding edit box in the dialog window), as shown in Figure 21.

If no new DSA parameters values are supplied, the defaults are used. They are values internally stored in the card for the workstation Safenet Luna PCM token, or the current values (those of the current key pair stored on the Fortezza card) for the SMART-24, respectively.

IMPORTANT

*For the SMART-24 (Site) key pair generation only, the key id, an integer value, must be entered by the user in the "Site Key ID" edit box (see Figure 22). The key id parameter is **unique** for each key pair; once assigned to a generated key pair by the SMART-24 same value cannot be reused for another key pair. Recommended is to set a convention for key id assignment, for example based on the date and time when the key pair is generated.*

The image shows a software window titled "StationKeyCommand" with three tabs: "Generate Key Pair", "Start Key Pair", and "Get Key Pair". The "Generate Key Pair" tab is selected. Inside the window, there is a section for "Authenticator" with a dropdown menu showing "Station: GEORH" and a "Generate Key Pair" button. Below this is a "Certificate Subject" field containing "O=CTBTO, OU=IMS, CN=cd11" and a "Site Key ID" field containing "1000". A checkbox labeled "Set DSA P, Q and G parameters" is checked. Below the checkbox are three text boxes containing long hexadecimal strings: "84bb7ca9cba765a9e1c8ae6244828dd9367c0e481d1b1a9d11ed4082268", "c51366487aaabab8001ada4dbd63fff1e726cfb3", and "34727388b9e35495a83afc2274ded7d2f56ba2c66d67cfbbd69123415e2".

Figure 21. Generate Key Pair window for Station with DSA parameters change.

The image shows a software window titled "StationKeyCommand" with three tabs: "Generate Key Pair", "Start Key Pair", and "Get Key Pair". The "Generate Key Pair" tab is selected. Inside the window, there is a section for "Authenticator" with a dropdown menu showing "Site: S1140 IP Address: 192.168.0.231" and a "Generate Key Pair" button. Below this is a "Certificate Subject" field containing "O=CTBTO, OU=IMS, CN=cd11" and a "Site Key ID" field containing "1005". A checkbox labeled "Set DSA P, Q and G parameters" is unchecked.

Figure 22. Generate Key Pair window for Site

Press the Generate Key Pair button to execute the command, and confirm command execution when prompted from the message window. Once the command execution is completed a new message is displayed as shown in Figure 23.

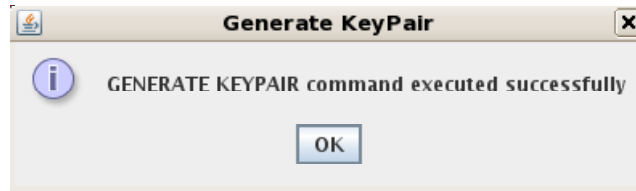


Figure 23. KeyPair Generated message

Following the command execution, the newly generated key pair is in a pending state, and the next **Start Key Pair** command execution will cause it to become effective. If more than one **Generate Key Pair** command is executed, only the last one is saved. The success or any errors that may occur during the execution of this command are displayed on the screen (see Figure 23).

The output generated by this command is written to a file located at `~/StationKeyCommand/GenereratedKeys` with the name as given by the above convention and the extension of “gen”. The content of this file includes the signature of the new public key signed with the old key in hex representation, the organization, unit, name and email, the key id, the public key and DSA P, Q and G parameters given as hexadecimal ASCII values. It is represented as follows:

```
Signature=9bc8a0458a6d5d15a5e88910a360cc7f30f7c3bf983b9c1e62ea29ca39
d692ee957b8bc3b63f0187
```

```
dnname=O=CTBTO, OU=IMS, CN=cd11
```

```
Key_ID=1
```

```
Y=43efbb07dedf97fd18aad7d906670b23d51fbb3e5c47b2da826880131d785c405
33a1370d637d8d3d1ee3d14610d14a6a84515ba11577c452cc4d1ea28f42ec558ee2
d007800ef44a1a2dcf73d2be9d95f3d37bc08273311561bdb9635f356f8ca6664abb
cfd461bd989c6dd8c398957ff2c1c28acfb688d8bc43cb644748f7
```

```
P=e3ec034dc16c91847ca69e1b32f764f23d02fa8aec0bdb16ffb2857f3f88fa12eb
b8abd3934f2b30d8b300baeaf555377f6f9add48cc5732e4d366ac50f6e2da73db0
86df2c8e4759bc1436a53dd7b9b8280edea523552908592cc7f289c11a1d5c9ca312
cc52d347bfdc61a5c951e3534bec8b144984d72660087f3187e803
```

```
Q=c81a4a494978fe86b992ecf73fb0cb25ecc4857f
```

```
G=4a3ed1c803e0e38ff2a8919503a80b347b9bbf550831335655c1fa33b4dd6b50dd
f9bbbbb37638f442276d7c55386decf5750d6574ca719ee6b498a269583a60ba9f8da
a2d7c9cde0841ce320cff3bbbcef0eb8d5ce8eef19cfecf3f33bed5bfd494421af3
2386cd66e7ef27fccc8d22b41d5abec8e04d94190516b48c890fa
```

For the Station only, same information written as a self-signed certificate in PEM format is stored in a file with the same name and with the extension “pem”. Also a certificate request in PEM format is generated and saved in a file with the same name and with the extension “req”.

This information must be provided to the appropriate users so that a new X509 certificate may be generated for the new key pair. This certificate must be included in the **Start Key Pair**

command request that will put these newly generated keys into effect. Failure to provide this information will result in un-verifiable communications.

5.5.2 Start Key Pair

Select this tab to execute an immediate **Start Key Pair** command that starts the pending key pair at the station or site authenticators. From the “Authenticator” drop-down list the user must select the Station code to denote the authentication card installed at the workstation, or, on a Linux system only, the Site code whose previously generated new key pair is to be put in effect (see Figure 24).

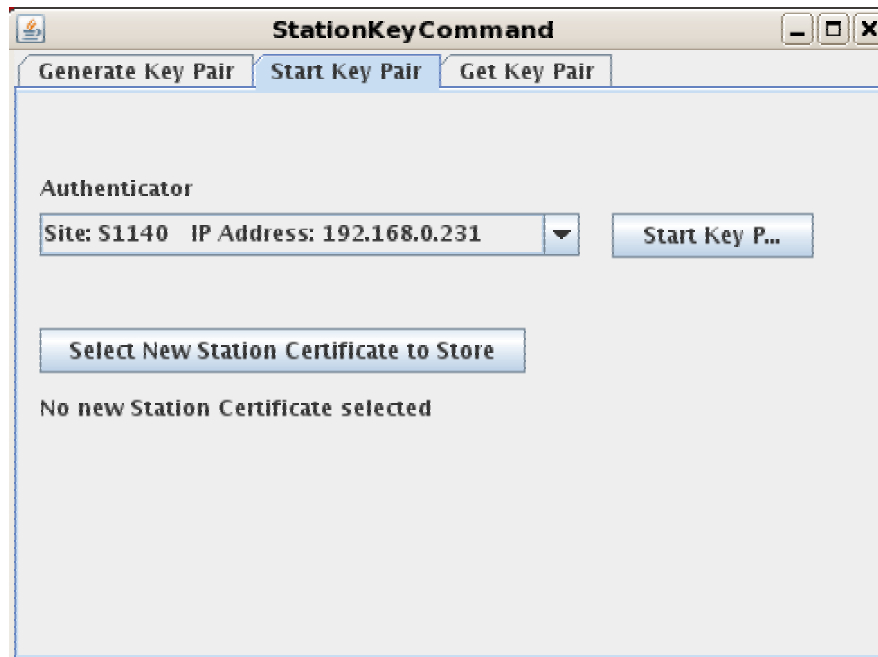


Figure 24. Start Key Pair window

Press the Start Key Pair button to execute the command, and confirm command execution when prompted from the message window. The progress and results of this command are displayed on the screen (see Figure 25). Note that if a new key pair has not been previously generated, this command will fail.

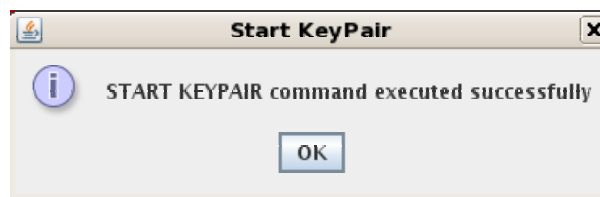


Figure 25. Key Pair Started message

The **Start Key Pair** command requires that a certificate be provided that will be used in the signing of messages that are passed from the Station, the sites and the users. To provide a certificate issued by a signing authority, press the Select New Station Certificate to Store button in Figure 30 and enter the file with the provided certificate when asked for its selection. If no certificate has been provided for the Station, the default (self-signed) certificate will be used.

The certificate used will be saved to a file located at `~/StationKeyCommand/StationCertificates` with an extension of “pem”, together with a text file with the the same information and format as **Generate Key Pair** response file (see §5.5.1), having the same name and the extension “sta”.

In the case of the **Start Key Pair** command for the SMART-24 remote sites, the certificate is not required for the command to be executed.

!!! WARNING !!!

Execution of this command will change both the public and private keys of the Station or Site selected. This may render pending communications un-verifiable by the recipient unless the information from the **Generate Key Pair** is provided to the appropriate users.

These local commands should be executed only by authorized personnel and only after carefully evaluating their effect on data and command communications

5.5.3 Get Key Pair

Select this tab to execute the **Get Key Pair** command that retrieves the current effective public key and DSA parameters in use at the Station or SMART-24 remote sites. From the “Authenticator” drop-down list the user must select the Station code to denote the authentication card installed at the workstation, or, on a Linux system only, the SMART-24 Site code from which the currently active key will be retrieved (see Figure 26).

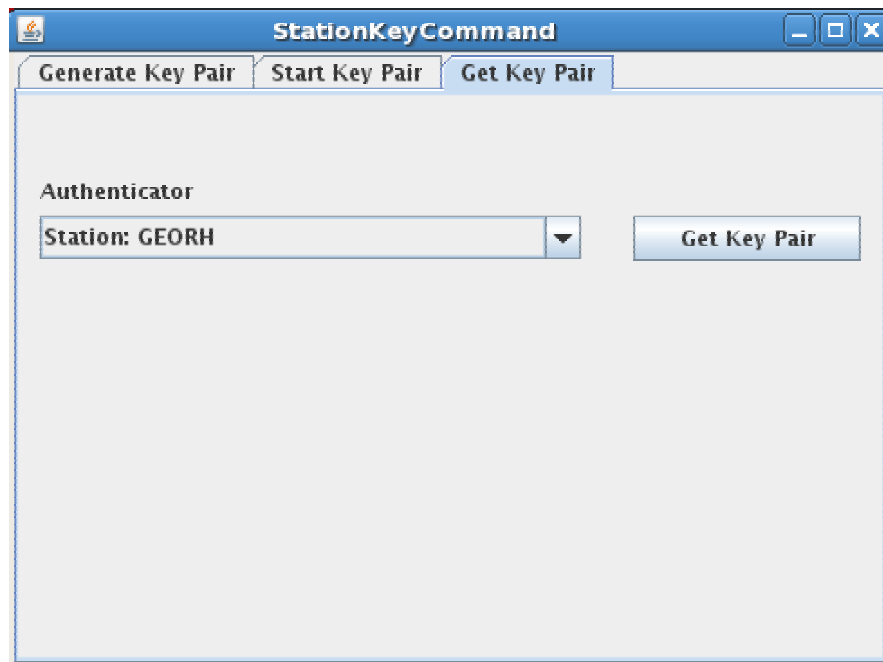


Figure 26. Get Key Pair window

Press the Get Key Pair button to execute the command, and confirm command execution when prompted from the message window. The progress and results of this command are displayed on the screen (see Figure 27). The output generated by this command consists of a text file with the same information and format as **Generate Key Pair** response file (see §5.5.1), located at

~/StationKeyCommand/RetrievedKeys and named using the previously described file naming convention with an extension of “get”. For the Station only, the self-signed certificate in PEM format is also provided in a file with the extension “pem”.



Figure 27. Key Pair Retrieved message

5.6 SMARTKeyCommand

SMARTKeyCommand is an application that enables the user to execute key management commands for the system Fortezza Cryptology Cards located at the SMART-24 remote sites and at the Workstation. It provides two purposes. First, it monitors key management commands sent to the AutoDRM, processes them, routes them to their correct destination, and provides a response to that command. Second, it provides the local operator a method of executing local key management commands. As a convenience to local operators, commands and their responses are logged to the SMARTKeyCommand window. Figure 28 shows a representative screen shot.

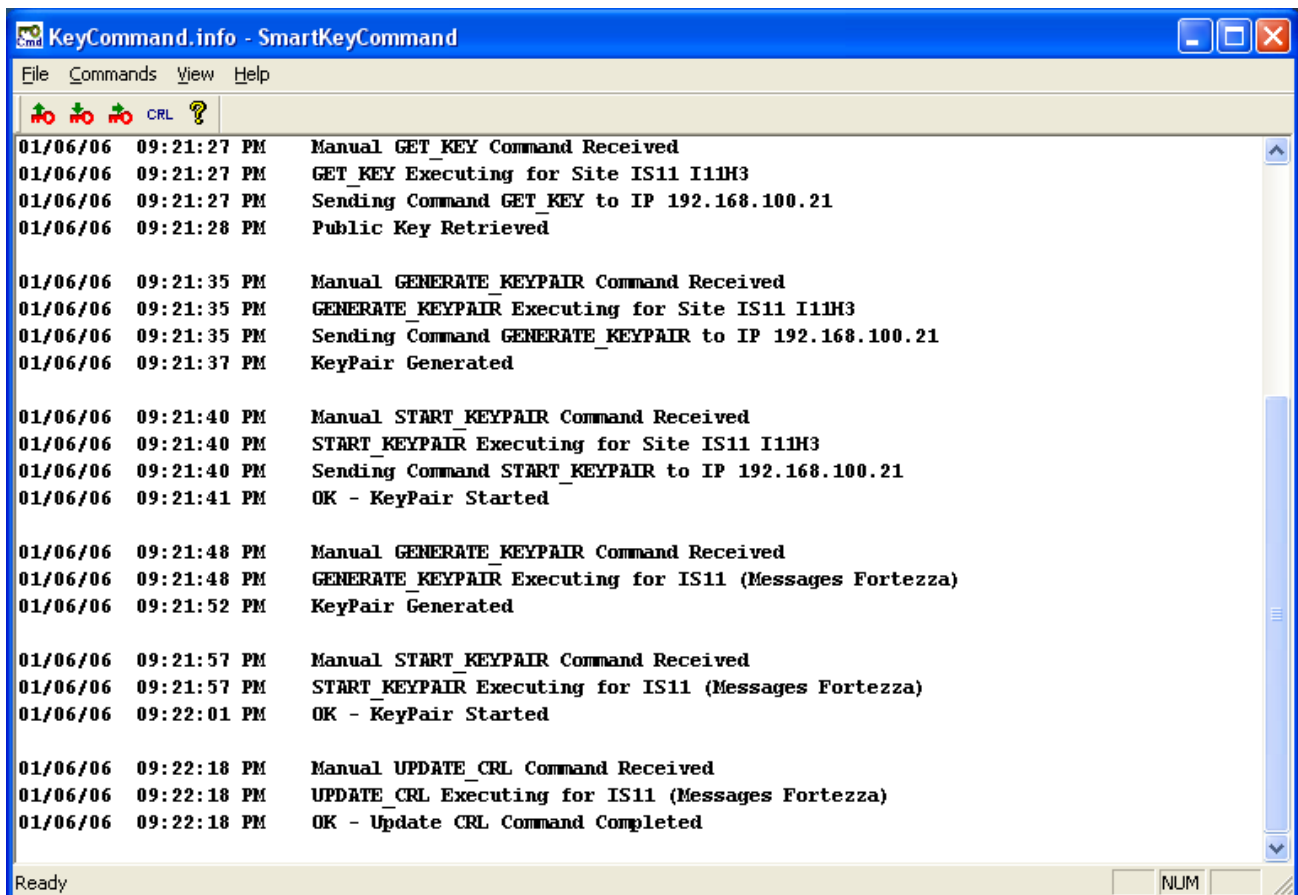


Figure 28. SMARTKeyCommand Window

At startup, SMARTKeyCommand reads the station name and the list of site names with their IP addresses from the text file `\station\SMARTKeyCommand.ini`. The content of this file is as follows:

```
Station=IS11
Site=I11H1    Ip=192.168.100.23
Site=I11H2    Ip=192.168.100.20
Site=I11H3    Ip=192.168.100.21
Site=I11H4    Ip=192.168.100.22
```

To re-configure the system, first close the SMARTKeyCommand application, open this file with an editor like Notepad and change the fields to reflect the correct array configuration, and then restart SMARTKeyCommand.

The four toolbar buttons provide simple access to the key management commands. Unless otherwise noted, the output file naming convention used is as follows:

`S(Station or Site Name)_T(hhmmss)_M(mo)_D(dom)_Y(year).xxx`

The first character of the file name is 'S' for station messaging, 'C' for station CD1.1, and 'R' for the site authenticators, respectively. The file name includes the station or site name, and the date (year, month and day of month) and time (hour, minute and second) when the command was executed. The file extension, denoted in this example as `xxx` is changed to indicate the type of information contained in the file.

IMPORTANT

The key management operations for the CD1.1 data Fortezza Cryptology card that is installed at the workstation are available by local commands only.

The key management commands are described in detail the following sections. A flow chart of the local key management commands is shown in Figure 31. For more information on the automatic key management commands, please refer to the SMARTAutoDRM documentation.



Pressing this button will cause SMARTKeyCommand to execute the **GENERATE_KEYPAIR** command that generates new key pairs at the station or site authenticators. A dialog box will appear from which the user must select the Station code with one of the two Fortezza cards installed at the workstation (for CD1.1 data or AutoDRM Messages), or the Site code for which a new key will be generated (see Figure 29).

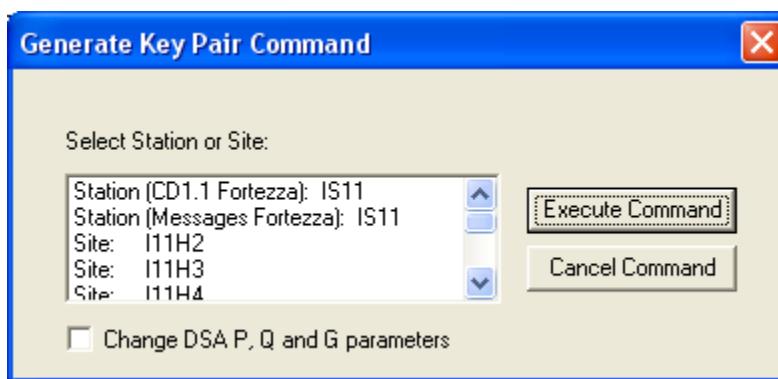


Figure 29. SMARTKeyCommand Generate Key Pair dialog box

Optional, the user can provide the DSA P, Q and G parameters to be used when generating the key in the following format:

```
DSA_P a3e47d7aec749a12950cfa002174b9c7217975f673f1cdbcf2111444b...
DSA_Q 9b0f6cfd374a9e11768eab9be0158f9c8ce2c2bd
DSA_G 23dc0a98c9f2fbe42ae642bcdbe66b5e602255ee08f17d056b78cdfaf8...
```

This information is included in the email request when using remote commands via AutoDRM messages, or can be input in a text file (after checking the checkbox in Figure 29) when using local commands.

If no new DSA parameters values are supplied, the defaults are used. They are the current values (those of the current key pair stored on the Fortezza card) for the SMART-24, and the values stored in `Station\authd.par` or `\Station\authd1.par` files for the workstation AutoDRM Messages or CD1.1 data Fortezza card, respectively.

Following the command execution, the newly generated key pair is in a pending state, and the next START-KEYPAIR command execution will cause it to become effective. If more than one GENERATE_KEYPAIR command is executed, only the last one is saved. The success or any errors that may occur during the execution of this command will be displayed on the screen (see Figure 28), and are logged in the file `\Station\GenerateKeypair.log`. The output generated by this command is written to a file located at `\Station\GeneratedKeys` with an extension of “gen”. This file is overwritten each time a key is generated for that station or site. The content of this file includes the signature of the new public key signed with the old key in hex representation, the organization, unit, name and email, the public key and DSA P, Q and G parameters given as hexadecimal ASCII values. It is represented as follows:

```
SIGNATURE
302C0214068FE6DEC420906C2B956473697819940527D1BF02143BFD696E7529DDCA4C
24FAB0424C

  dname=o=ORG ou=UNIT l=autodrm cn=autodrm@icpa.geoinstr.com
y=228DF38A669B2A834DC90B6F0C02BDCA07F2A98838946E3AC468874B9FBA07036C
2A4AECCC489F56431EECB6384A5FFCCEA07F66DE49DB977F26075C05777539B597A7
6C4A1AB83061FB0067ED56E16A568BA8CA5D59A6CD9E2C047944116AE07ADD6CF408
67D29482DEAF7C0F53846C1D4B94222836B4561F61F43DE38414B5
```

p=95796DC5BDABB8AFAF5B7BB99ACEAE4B12C8ECC82FB6237C5BE26F5105BF9F1E8B
8DFC68647646EEA87C4F42CD2783A99717914A1FAA9DB6627BB137179C9BA7443DE4
5FA348AB8E343B12230A304B3C6658CFF1023BEC8F6A8D2AC5E1568C44D9B7EB35C3
2BBAE5D33EDD80CECFE7D88416FDD81A2675A72BEE435BED6F868B

q=8AA48ABD2805D4ED5F2437671D46A6D717CB1091

g=59505E9150E45BB7B96B3125E085BF9FF2A3C3439E3D5FF9BFA12B377DF9A45E06
91A5F4560058914AA88AA6A65E52B0EB0802FFCD77D73A4B8DEAF82D6A75DFE02B74
465CC2F1877680D4D35DF52A82943C2EFD35F5939346833C38A68051502106E734EB
8D6FFD1CE7E18541BEA5A95381589C4C905272D72F73539485FFEF

Same information written as a self-signed certificate in PEM format is stored in \Station\dsacert.tmp or \Station\dsacert1.tmp files for the workstation AutoDRM Messages or CD1.1 data Fortezza card, respectively.

This information must be provided to the appropriate users so that a new x509 certificate may be generated for the new key pair. This certificate must be included in the START_KEYPAIR command request that will put these newly generated keys into effect. Failure to provide this information will result in un-verifiable communications.



Pressing this button will cause SMARTKeyCommand to execute an immediate **START_KEYPAIR** command that starts the pending key pair at the station or site authenticators. A dialog box will appear from which the user must select the Station code with one of the two Fortezza cards installed at the workstation (for CD1.1 data or AutoDRM Messages), or the Site code whose previously generated new key pair is to be put in effect (see Figure 30).

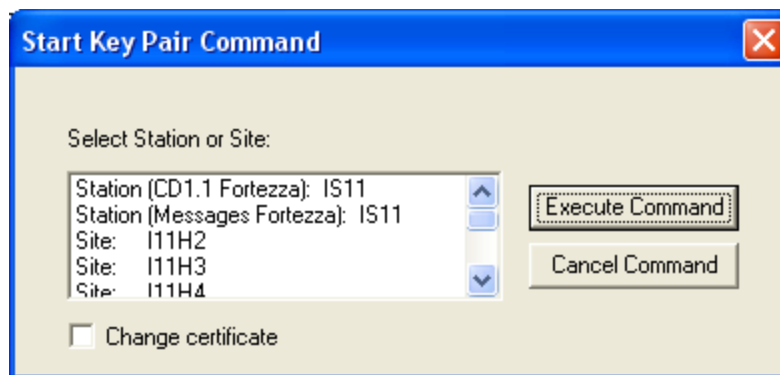


Figure 30. SMARTKeyCommand Start Key Pair dialog box

If a new key pair has not been generated, this command will fail. The progress and results of this command are displayed on the screen (see Figure 28), and are logged in the file \Station\StartKeypair.log.

The START_KEYPAIR command requires that a certificate be provided that will be used in the signing of messages that are passed from the Station, the sites and the users. This information is included in the email request when using remote commands via AutoDRM messages. When using the manual command there are two ways to provide this certificate information. To manually provide a certificate issued by a signing authority, check the checkbox in Figure 30 and enter the file with the provided certificate when asked for its selection. If no certificate has been

provided for the workstation AutoDRM Messages Fortezza card, the default (self-signed) certificate from `\Station\dsacert.tmp` will be used. In either case, the certificate used will be written to a file located at `\Station\StationCertificates` with an extension of “key”. Also, the workstation Messages Fortezza card certificate will be stored in the file `\Station\station.pem` and will be used to sign AutoDRM originated email messages.

Only in the case of the manual `START_KEYPAIR` command for the SMART-24 remote sites, the certificate is not required for the command to be executed.

!!! WARNING !!!

Execution of this command will change both the public and private keys of the Station or Site selected. This may render pending communications un-verifiable by the recipient unless the information from the `GENERATE_KEYPAIR` is provided to the appropriate users.

These local commands should be executed only by authorized personnel and only after carefully evaluating their effect on data and command communications.

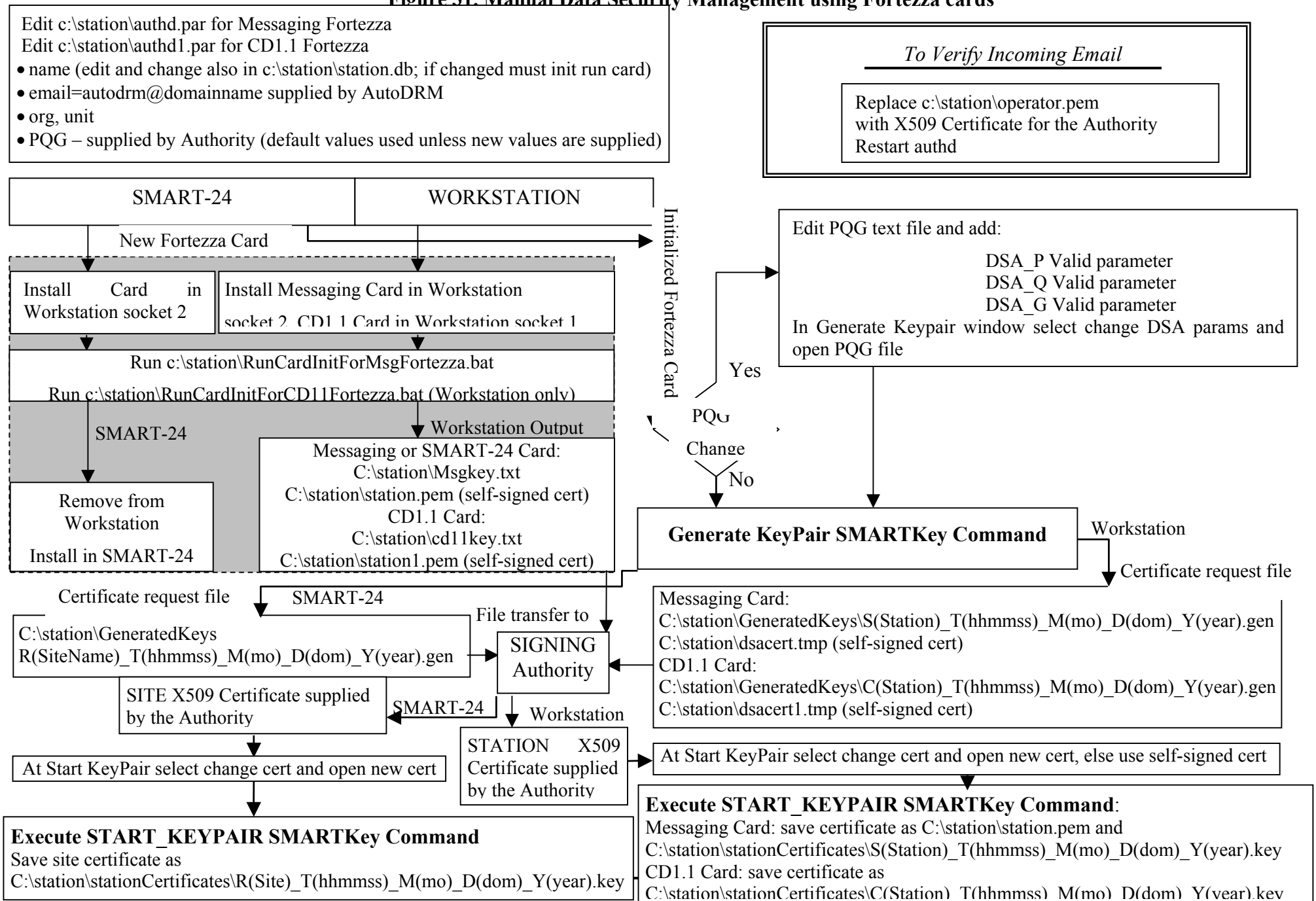


Pressing this button will cause SMARTKeyCommand to execute the **GET_KEYPAIR** command that retrieves the current effective public key and DSA parameters in use at the SMART-24 remote sites. A dialog box will appear from which the user must select the Site code from which the currently active key will be retrieved. This command is not available for the Station. Also, this command is available only locally and not by using AutoDRM messages. The progress and results of this command are displayed on the screen (see Figure 28). The output generated by this command consists of the same information as `GENERATE_KEYPAIR` response file, written to a file located at `\Station\RetrievedKeys`. This file is named using the previously described file naming convention with an extension of “get”. It is overwritten each time a key is retrieved for that site.



Pressing this button will cause SMARTKeyCommand to execute the **UPDATE_CRL** command that updates the Certificate Revocation List (CRL) in use at the station or site authenticators. A dialog box will appear from which the user must select the Station code with one of the two Fortezza cards installed at the workstation (for AutoDRM Messages or CD1.1 data), or the Site code for which the CRL will be updated. The `UPDATE_CRL` command requires that a CRL certificate be provided that will be used to update the station or site copy. This information is included in the email request when using remote commands via AutoDRM messages, or must be entered by the user as a file when asked for its selection in manual mode. The progress and results of this command are displayed on the screen (see Figure 28). The CRL certificate used will be stored in a file located at `\Station\StationCRL`. This file is named using the previously described file naming convention with an extension of “crl”. It is overwritten each time a CRL is updated for the same station or site authenticator.

Figure 31. Manual Data Security Management using Fortezza cards



(Page Intentionally Left Blank)