

Winston Technical Documentation

Version 1.3.17

Documentation last updated (11/6/2020)

TABLE OF CONTENTS

1	Overview	3
2	System Requirements & Installation	3
3	WWS Configuration	3
4	Starting Winston Wave Server	6
5	Data Import	6
5.1	Import From Earthworm (ImportEW)	6
5.1.1	ImportEW Configuration	6
5.1.2	Running ImportEW	11
5.2	Import From Another Winston (ImportWS)	11
5.2.1	ImportWS Configuration	12
5.3	Import From Data Files (SAC, SEED, SEISAN)	13
5.4	Importing Station Locations	14
5.4.1	ImportDataless	14
5.4.2	ImportHypoinverse	15
6	Database	15
6.1	ROOT Database	15
6.1.1	Channels	15
6.1.2	Channelmetadata	15
6.1.3	Groupnodes	16
6.1.4	Grouplinks	16
6.1.5	Instruments	16
6.1.6	Instrumentmetadata	16
7	Winston Tools	17
7.1	Admin	17
7.2	Merge	18
7.3	PlotHelicorder	18
7.4	Upgrade	18

8	WWS Commands	18
9	Client Applications	21
9.1	Winston Web Interface	21
9.2	Swarm	21
10	Source Code	21
Appendix A	SQL Nuggets	21
Appendix B	Manual Update of Station Locations	22
Appendix C	Detailed Install Instructions for CentOS 7	24
Appendix D	Example Winston.service for CentOS7	25
Appendix E	Example ImportEW.service for CentOS7	26
Appendix F	Hyponinverse Station File Format.....	27

1 OVERVIEW

Winston is a Java-based suite of tools for storing, serving and plotting seismic waveform data. Winston's data storage system, or Winston Wave Server (WWS), is based on [Earthworm](#)'s Wave Server protocol but uses MySQL compatible database for storage. Scripts are provided for data import from Earthworm, other Winston's, and files. The plots are available through a web interface.

2 SYSTEM REQUIREMENTS & INSTALLATION

Winston requires Java 8 or higher and a MySQL compatible database (e.g. MySQL, MariaDB). Winston is platform independent but is typically run on Windows or Linux systems.

To install Winston:

1. Download and unzip latest Winston from <https://volcanoes.usgs.gov/software/winston/index.shtml>.
2. Install MySQL or MariaDB and set database root password
3. Create Winston user in the database. Example is below but be sure to change 'password' to something else.

```
mysql> CREATE USER 'winstonuser'@'localhost' IDENTIFIED BY 'password';
mysql> GRANT ALL PRIVILEGES ON . TO 'winstonuser'@'localhost' WITH GRANT OPTION;
mysql> CREATE USER 'winstonuser'@'%' IDENTIFIED BY 'password';
mysql> GRANT ALL PRIVILEGES ON . TO 'winstonuser'@'%' WITH GRANT OPTION;
mysql> flush privileges;
```

3 WWS CONFIGURATION

The configuration files are in the root directory of the unzipped Winston folder. All Winston configuration files are unordered list of [key]=[value] pairs. When WWS starts it will look for a configuration file called *WWS.config* in the current working directory.

Table 1 *WWS.config*

Key	Required/Optional	Description
wws.port	required	The port WWS will bind to
wws.keepalive	optional	If true, <code>SO_KEEPALIVE</code> will be set on accepted connections. Helpful when transiting firewalls.
wws.handlers	required	The number of server handlers. Server handlers receive requests from clients and fill them by requesting data from the database. Most servers run well with 4 handlers. Excessive handlers will make inefficient use of system resources, reducing the overall number of requests the server can fill. Wave

		Servers which serve many HTTP plots or clients across slow networks may benefit from additional handlers.
<code>wws.maxConnections</code>	required	The maximum number of connections that the Wave Server will maintain. Unlike handlers this number can frequently be safely increased. (This option may no longer be supported as there is no longer a need to limit number of connections.)
<code>wws.idleTime</code>	optional	The length of time, in seconds, that a connection can remain quiet before WWS will consider it idle. Used when dropping idle connections to free resources.
<code>wws.allowHttp</code>	required	If true, the Wave Server will respond to requests from web browsers. If false, only the wave server protocol will be supported.
<code>wws.maxDays</code>	required	The maximum age of data, in days, that will be returned to a client. If 0, all data will be available to fill client requests. Used to permit multiple WWS instances to feed from a single database while presenting different apparent retention policies.
<code>wws.slowCommandTime</code>	optional	The length of time, in milliseconds, a command can run before being logged as slow command.

Example WWS.config

```

wws.addr=192.178.2.120
wws.port=16024
wws.keepalive=true
wws.handlers=4
wws.maxConnections=50
wws.idleTime=7200
wws.allowHttp=true
wws.httpMaxSize=10000000
wws.httpRefreshInterval=300
wws.maxDays=0
wws.slowCommandTime=15000
@include winston.config

```

WWS.config also includes *Winston.config* (`@include Winston.config`) at the very end. *Winston.config* contain settings that may be used in multiple Winston applications, so they are separated here to avoid having to configure it in multiple files.

Table 2 *Winston.config*

Key	Required/Optional	Description
-----	-------------------	-------------

winston.driver	required	The fully qualified class name for the database driver to use to connect to Winston. Most likely you'll never have to change this.
winston.url	required	The JDBC URL used to connect to the Winston database.
winston.prefix	required	The prefix on all of the Winston databases. Multiple Winstons may share the same MySQL instance provided each has a unique prefix.
winston.statementCacheCap	Optional	Maximum size of the statement cache. Defaults to 100
winston.tableEngine	Optional	The number of real-time stations Winston can handle is commonly limited by database write times. Winston avoids use of transactional features available in some MySQL storage engines to help streamline writes. Because of this, MyISAM is the preferred storage engine for non-trivial Winston installations.

Example Winston.config

```
winston.driver=com.mysql.jdbc.Driver
winston.url=jdbc:mysql://localhost/?user=winstonuser&password=winstonpass
winston.prefix=W
winston.statementCacheCap=200
winston.tableEngine=MyISAM
```

Note 1: The winston.tableEngine may not work with older Winston installations. The best way to ensure your Winston tables are getting created as MyISAM is to set default-storage-engine=MyISAM in the /etc/my.cnf:

```
mysql>
default-storage-engine=MyISAM
```

Note 2: *If you are on MySQL 8* or are getting a warning such as below, you may need to add the useSSL=false option to the winston.url or provide truststore for server certificate verification.

```
Fri Nov 06 10:39:01 PST 2020 WARN: Establishing SSL connection without server's
identity verification is not recommended. According to MySQL 5.5.45+, 5.6.26+ and
5.7.6+ requirements SSL connection must be established by default if explicit option
isn't set. For compliance with existing applications not using SSL the
verifyServerCertificate property is set to 'false'. You need either to explicitly
disable SSL by setting useSSL=false, or set useSSL=true and provide truststore for
server certificate verification.
```

```
winston.url=jdbc:mysql://localhost/?user=winstonuser&password=winstonpass&useSSL=false
```

4 STARTING WINSTON WAVE SERVER

WWS can be started by running `bin/WWS.bat` on Windows, or `bin/WWS.sh` on Linux and Mac. If WWS successfully started, you should see output like this:

```
2019-11-26 21:54:22 INFO - Version: 1.3.17 Built: 20200820-1859
2019-11-26 21:54:22 INFO - config: wws.addr=winston.
2019-11-26 21:54:22 INFO - config: wws.port=16024.
2019-11-26 21:54:22 INFO - config: wws.dbConnections=5.
2019-11-26 21:54:22 INFO - Launching WWS. Version: 1.3.17 Built: 20200820-1859
2019-11-26 21:54:23 INFO - WWS started and listen on /192.178.2.120:16024
```

If WWS is started as a service on boot, then you will see this in the `WWS.log` file. Sample `Winston.service` for CentOS 7 is available in Appendix D. If Winston service does not start, the best way to troubleshoot is to run it from command line as described in previous paragraph.

The Winston root database is created automatically when the first waveforms are ingested. The wave server will not function until there is data to server. Most Winston users feed data from Earthworm export. Data can also be imported from another Winston or loaded using SEED or SAC files. See section 5 for more info on data import.

5 DATA IMPORT

5.1 IMPORT FROM EARTHWORM (IMPORTEW)

A common configuration is to use Earthworm as the acquisition software and then export the wave data to Winston for wave server storage. This is accomplished by 1) configuring the Earthworm instance to export `TRACEBUF` or `TRACEBUF2` messages using an export module (e.g. `export_generic`), and 2) configuring and running `ImportEW` in Winston to import the exported data.

5.1.1 ImportEW Configuration

`ImportEW.config`, found with the Winston distribution, is the configuration file for `ImportEW`.

5.1.1.1 Import Configurations

Key	Required/Optional	Description
<code>Import.host</code>	required, unique	Host name or IP address of the Earthworm <code>export_generic</code> module that is sending the data.
<code>import.port</code>	required, unique	Port number of the Earthworm <code>export_generic</code> module that is sending us the data.

import.receiveID	required, unique	The identification string that is sent from the Earthworm export_generic module.
import.sendID	required, unique	The identification string that is sent to the Earthworm export_generic module.
import.heartbeatInterval	required, unique	Number of milliseconds between sending heartbeats to the export.
import.expectedHeartbeatInterval	required, unique	Number of milliseconds between expected heartbeats from the export.
import.timeout	required, unique	Number of milliseconds before socket timeouts
import.dropTableDelay	required, unique	Number of seconds to delay between successive table drops after the GMT day changes. For users with a large number of channels this can eliminate some slowdown during the GMT day change.
import.log.name	required, unique	Name of the ImportEW log. Each log file will have a number appended to it.
import.log.numFiles	required, unique	Number of log files to rotate through.
import.log.maxSize	required, unique	Maximum number of bytes for a single log file.
import.enableValarmView	required, unique	Controls creation of views intended to be used with the VAlarm package.

Example configuration

```
import.host=192.168.0.101
import.port=18000
import.receiveID=alive
import.sendID=alive
import.heartbeatInterval=30000
import.expectedHeartbeatInterval=30000
import.timeout=60000
import.dropTableDelay=10
import.log.name=ImportEW.log
import.log.numFiles=10
import.log.maxSize=1000000
import.createValarmView=false
# include the database connection parameters
@include winston.config
```

Note that Winston.config is also imported in this configuration file.

5.1.1.2 Filter Configurations

Filters are used to accept or reject tracebufs. *All TraceBufs are rejected unless accepted by one of the filters.*

Table 3 Options available to all filters

Key	Required/Optional	Description
filter	At least one required, multiple allowed.	Specifies a filter for incoming TraceBufs. Each filter needs further information about how to run. This information is provided on subsequent lines of the configuration file.
filter.class	Required	Specifies the Java class for the filter.
filter.order	Optional	The order in which this filter should be applied if there are multiple filters.
filter.action	Optional	Whether to 'reject' or to 'accept' the TraceBuf if it meets the filter's criteria. Defaults to 'accept'.
filter.log	Optional	Specifies Log Level. 0=Finest, 1=Fine, 2=Warning.

The following filters are available in ImportEW:

5.1.1.2.1 gov.usgs.volcanoes.winston.in.ew.TimeFilter

TimeFilter filters TraceBufs by their start time.

Table 4 TimeFilter options

Key	Required/Optional	Description
past	Optional	Matches if start time is older that this number of seconds. Leave undefined to not perform this check.
future	Optional	Matches if start time is in the future more than this number of seconds. Leave undefined to not perform this check.

Example TimeFilter

```
# Filters apply to incoming TraceBufs
# rejects packets from more than 10 seconds in the future
filter=TimeFilter
TimeFilter.class=gov.usgs.volcanoes.winston.in.ew.TimeFilter
TimeFilter.order=1
TimeFilter.past=-3600
TimeFilter.future=120
TimeFilter.action=reject
TimeFilter.log=1
```

5.1.1.2.2 gov.usgs.volcanoes.winston.in.ew.MaxDaysFilter

MaxDaysFilter filters TraceBufs by whether they are older than maxDays value in their options set. See section 5.1.1.3 for more information on options.

Example MaxDays

```
filter=MaxDays
MaxDays.class=gov.usgs.volcanoes.winston.in.ew.MaxDaysFilter
MaxDays.order=0
MaxDays.action=reject
MaxDays.log=1
```

Note: There is no performance-based reason to limit the amount of data stored in Winston. Any delays in response to Winton is likely a MySQL issue. Some effort will have to be made to track down and fix bottlenecks. This is something a system administrator might be able to help with. If there is a performance drop while querying older data, that points to the database caches or physical disk as the bottle neck. Giving the caches more memory to work with while keeping a close eye on I/O performance and watching for any unexpected loads on the disk may help.

5.1.1.2.3 gov.usgs.volcanoes.winston.in.ew.SCNLFilter

SCNLFilter filters TraceBufs by their SCNLs.

Table 5 SCNLFilter options

Key	Required/Optional	Description
scnl	Optional	A regular expression for the SCNLs to match. Four space-separated fields.

Example SCNLFilters

Full example for SCNLFilter that accepts all TraceBufs from network AV:

```
filter=All
All.class=gov.usgs.winston.in.ew.SCNLFilter
All.order=3
All.scnl=* * AV *
All.action=accept
```

Full example for SCNLFilter that rejects all north component TraceBufs:

```
filter=RejectNorth
RejectNorth.class=gov.usgs.winston.in.ew.SCNLFilter
RejectNorth.order=2
RejectNorth.scnl=* .N. * *
RejectNorth.action=reject
```

5.1.1.2.4 gov.usgs.winston.in.ew.McCalPulseFilter

McCalPulseFilter identifies McVCO calibration pulses.

Table 6 McCalPulseFilter options

Key	Required/Optional	Description
preambleFreq	Optional	McVCO preamble frequency. Default is 21.25.
Threshold	Optional	Preamble detection limit. Default is 500.
terminal	Optional	If true, matching tracebufs will not be passed to further filters. Default is false.

Example McCalPulseFilter

```
filter=McCalPulse
McCalPulse.class=gov.usgs.volcanoes.winston.in.ew.McCalPulseFilter
McCalPulse.order=0
McCalPulse.threshold=500
McCalPulse.terminal=false
```

Note: This filter may be removed in the future.

5.1.1.3 Option Configurations

Table 7 Options available to each options set

Key	Required/Optional	Description
options	required, multiple allowed	Specify options specific to a class of SCNLs.
rsam.delta	Optional	Number of seconds behind current time to calculate the average sample value. Default is 10 seconds.
rsam.duration	Optional	Number of seconds used to calculate the RSAM value. Default is 60 seconds.
timeThreshold	Optional	Time threshold for whether to flush buffered packets to the database. If a currently buffered packet is older than this number of seconds, then all packets for this channel are written to the database. Default is 1 second.
maxBacklog	Optional	The maximum number of TraceBufs to hold before dropping them. The backlog is used when the database slows down or otherwise malfunctions. Default is 100.
maxDays	Optional	Maximum number of days to store in the database. Whenever a new GMT day occurs a new table is created that stores the tracebufs for that day. At that time, ImportEW checks to see if more than maxDays tables exist for that channel. If so, it drops as many tables as necessary to get down to maxDays tables. An unspecified value or 0 value means to never drop tables.
applyTo	Required	Comma-separated list of four space-separated SCNL regular expressions that specifies which channels this option set should apply to.

Example options

```
# Default options
options=Default
Default.rsam.enable=true
Default.rsam.delta=10
Default.rsam.duration=60
Default.timeThreshold=60
Default.traceBufThreshold=60
Default.maxBacklog=200
Default.maxDays=0

# Fast options (to ensure that Tracebufs from vertical components are written to the
database as soon as they come in)
options=Fast
Fast.timeThreshold=1
Fast.traceBufThreshold=1
Fast.maxBacklog=200
Fast.rsam.delta=10
Fast.rsam.duration=60
Fast.applyTo=* ..Z * *
```

5.1.2 Running ImportEW

ImportEW can be run from command line but typically it is run as a service to ensure continuous import. Example of ImportEW.service for CentOS 7 is found in Appendix E. However, it is best to first run on command line to ensure proper configuration and troubleshoot any issues.

Command line usage:

```
java gov.usgs.volcanoes.winston.in.ew.ImportEW [-options] [config file]
```

or:

```
java gov.usgs.volcanoes.winston.in.ew.ImportEW --help
```

5.2 IMPORT FROM ANOTHER WINSTON (IMPORTWS)

Data into a WWS can be imported from another WWS instance from command line:

```
java -cp lib/winston.jar gov.usgs.volcanoes.winston.in.ew.ImportWS --help
```

Usage:

```
java gov.usgs.volcanoes.winston.in.ew.ImportWS [--help] [(-t|--timerange)
<timeRange>]
[(-w|--waveserver) <host:port>] [-i|--noinput] [-l|--SCNL] <configFilename>
```

Winston ImportWS

This program gets data from a Winston wave server and imports it into a Winston database. See 'ImportWS.config' for more options.

All output goes to standard error.

The command line takes precedence over the config file.

```
[--help]
    Prints this help message.

[(-t|--timerange) <timeRange>]
    The time range. Relative times are assumed to be in the past.

[(-w|--waveserver) <host:port>]
    The Winston wave server to poll.

[-i|--noinput]
    Do not poll keyboard for input.

[-l|--SCNL]
    Always request SCNL

<configFilename>
    The config file name. (default: ImportWS.config)
```

%

A convenience script, ImportWS.bat and ImportWS.sh is provided in the bin directory. The script can be run also from cron or task scheduler for repeated imports using relative time in the timeRange.

5.2.1 ImportWS Configuration

Table 8 ImportWS Configuration

Key	Required/Optional	Description
waveServer	required	Hostname or IP and port of the Winston instance to pull from in <host>:<port> format.
createChannels	required	Create channel in database if it does not exist. Set to true. Otherwise it will not import the channel if it doesn't already exist.
createDatabase	required	Create root database if it does not exist. Set to true. Otherwise it will not create root database on import start.
findGaps		Not implemented.
timeRange	required	Time range of the data to pull. It can be specified as [startTime],{,endTime}} or by specifying the relative time in days/hours/minutes before present. Examples: # one day explicit YYYYMMDDHHMMSS timeRange=20051201000000,20051202000000 # one day with time range timeRange=-1d,20051202000000

		# or timeRange=-24h,2005120200000 # last 10 minutes timeRange=-10i
chunkSize	required	Maximum size in seconds to ask for from the wave server.
chunkDelay	required	Time to wait in milliseconds between asking for chunks.
rsam.delta	required	Number of seconds behind current time to calculate the average sample value.
rsam.duration	required	Number of seconds used to calculate the RSAM value
channel	Required, multiple allowed	Comma-separated list of four space-separated SCNL regular expressions to specify which channels to import.

Example ImportWS.config

```
@include winston.config
# Server to pull from
waveServer=192.168.0.39:16022
createChannels=true
createDatabase=true
findGaps=true # unimplemented
timeRange=-24h
chunkSize=3600
chunkDelay=0
rsam.delta=10
rsam.duration=60
channel=* * * *
```

5.3 IMPORT FROM DATA FILES (SAC, SEED, SEISAN)

Winston provides ImportSAC, ImportSEED, and ImportSeisan scripts (.bat for Windows, .sh for Linux and Mac OS) to import data files into the WWS. To import simply pass in the files names as an argument to the appropriate script, e.g.:

```
./bin/ImportSEED.sh file1 file2 file3
```

ImportSEED and ImportSeisan scripts also allow the channel info and RSAM delta and duration to be set during the import. To see full import options, you can run the command with --help option:

```
[winston@winston bin]$ ./ImportSEED.sh --help
```

Usage:

```
java gov.usgs.volcanoes.winston.in.ImportSeed [--help] [(-s|--station)
<station>] [(-c|--channel) <channel>] [(-n|--network) <network>]
[(-l|--location) <location>] [(-r|--rsamDelta) <rsamDelta>] [(-d|--rsamDuration)
<rsamDuration>] [file1 file2 ... fileN]
```

Import miniSEED

This program imports data from miniSEED volumes into a winston database

```
[--help]
    Prints this help message.

[(-s|--station) <station>]
    Override station identifier code

[(-c|--channel) <channel>]
    Override channel identifier

[(-n|--network) <network>]
    Override network identifier

[(-l|--location) <location>]
    Override location identifier

[(-r|--rsamDelta) <rsamDelta>]
    Override location identifier
    (default: 10)

[(-d|--rsamDuration) <rsamDuration>]
    Override location identifier
    (default: 60)

[file1 file2 ... fileN]
    files to import.
```

RSAM parameters: delta=10, duration=60.

2019-12-31 10:22:29 INFO - Connected to database.

No files to import.

5.4 IMPORTING STATION LOCATIONS

5.4.1 ImportDataless

Populate Winston station locations from SEED dataless volume.

```
% java -cp lib/winston.jar gov.usgs.volcanoes.winston.in.metadata.ImportDataless
Usage: ImportDataless [-c <winston.config>] <dataless>
%
```

5.4.2 ImportHypoinverse

Populate Winston station locations from Hypoinverse station file. See Appendix F for Hypoinverse file format.

```
% java -cp lib/winston.jar gov.usgs.volcanoes.winston.in.metadata.ImportHypoinverse
Usage: ImportDataless [-c <winston.config>] <dataless>
%
```

6 DATABASE

6.1 ROOT DATABASE

The Winston root (e.g. W_ROOT) database tables to be aware of:

6.1.1 Channels

Contains information about the channels available in Winston and some key metadata associated with the channel.

Table 9 Root channels table columns

sid	Station id (primary key)
iid	Instrument id, the primary key in instruments table
code	Channel name in SCNL in format: <station>\$<channel>\$<network>\$<location>
st	Start time in J2K seconds
et	End time in J2K seconds
alias	Used in place of channel name in helicorder images
unit	Wave amplitude unit, e.g. nm/s
linearA	Multiplier. Y-value is divided by this number after offset is applied.
linearB	Offset, subtracted from y-value.

6.1.2 Channelmetadata

Stores additional metadata not found in the channels table.

Table 10 Root channelmetadata table columns

sid	Station id from channels table (primary key)
name	Metadata name (primary key)
value	Metadata value

6.1.3 Groupnodes

Applications that use this information, such as Swarm, will organize channels in a tree data structure under these nodes. Groupnodes table contain information about the group or network nodes. The grouplinks table associates a station with a node.

Table 11 Root groupnodes table columns

nid	Node id (primary key)
parent	Parent node id
name	Name of the node
open	If true (1), this branch of the tree will be open in Swarm Data Source to display the subnodes or stations under this node.

6.1.4 Grouplinks

Maps a station to a node.

Table 12 Root grouplinks table columns

gid	Grouplink id (primary key)
sid	Station id from channels table
nid	Node id from groupnodes table
open	If true (1), this branch of the tree will be open in Swarm Data Source to display the subnodes or stations under this node.

6.1.5 Instruments

Contains information about an instrument's location and timezone. The instrument id (iid) here is mapped to the station id in channels table.

Table 13 Root instruments table columns

iid	Instrument id (primary key)
name	Instrument name, usually the 1-5 character station name.
description	Instrument description
lon	Instrument location longitude in decimal degrees
lat	Instrument location latitude in decimal degrees
height	Instrument location height in meters
timezone	Timezone used to stamp time on the data. This is typically UTC and can be left blank. If the instruments are configured to stamp time using localtime, this should be configured. If set, it may be used by Swarm to display the time in this timezone on the left vertical axis of the helicorder. Use timezones listed in Winston's web interface plot tabs. E.g. America/Guatemala

6.1.6 Instrumentmetadata

Table 14 Root instrumentmetadata columns

imid	Instrument metadata id (primary key)
-------------	--------------------------------------

iid	Instrument id from instruments table
name	Metadata name
value	Metadata value

7 WINSTON TOOLS

Convenience scripts (.bat and .sh) for each of below are provided in the bin directory of the Winston download.

7.1 ADMIN

The Admin class provides a collection of commands for administering Winston database.

```
% java -cp lib/winston.jar gov.usgs.volcanoes.winston.db.Admin --help
Winston Admin
```

A collection of commands for administering a Winston database.
Information about connecting to the Winston database must be present
in Winston.config in the current directory.

Usage:

```
java gov.usgs.volcanoes.winston.db.Admin [options] command [command arguments]
```

Valid options:

```
--delay seconds           the delay between each channel for commands
                           for multiple channels
```

Valid commands:

```
--list                   lists all channels
--list times              lists all channels with time span
--delete channel          delete the specified channel
--deletex SSS$CCC$NN[$LL] delete the specified channels where:
                           SSS$ is the station,
                           CCC is the channel which may contain
                           a wild card (%),
                           SSS$ is the station,
                           NN is the network,
                           LL is the optional location which may contain
                           a wild card (%)
--span                    recalculate table spans
--purge channel days      purge the specified channel for the
                           specified number of days
--purgex channel days     purge the specified channel for the
                           specified number of days where the channel
                           may contain a wild card (%) anywhere
--repair YYYY_MM_DD [channel] repair all tables on given day
                           optionally, just repair the specified channel
```

```
%
```

7.2 MERGE

```
% java -cp lib/winston.jar gov.usgs.volcanoes.winston.db.Merge
usage: java gov.usgs.volcanoes.winston.db.Merge [srcURL] [destURL] [table] [date]
[table] is case sensitive; example: CRP_SHZ_AK
[date] is in YYYY_MM_DD form; example: 2005_03_27
localhost [6:19pm] %
```

7.3 PLOTHELICORDER

PlotHelicorder class creates a PNG helicorder plot.

```
% java -cp lib/winston.jar gov.usgs.volcanoes.winston.PlotHelicorder
Server/station/time options
-wws [host]:[port], the WWS, mandatory argument
-s station, mandatory argument
-e end time [now], format: 'YYYYMMDDHHMMSS' (GMT) or 'now'
-m minutes on x-axis [20]
-h hours on y-axis [12]
-tz time zone abbreviation [GMT]
-to time zone offset, hours [0]

Output options
-x total plot x-pixel size [1000]
-y total plot y-pixel size [1000]
-lm left margin pixels [70]
-rm right margin pixels [70]
-o output file name [heli.png]
-c clip value, a number [auto]
-b bar range, a number [auto]
-r show clipped trace as red, 0 or 1 [0]
%
```

7.4 UPGRADE

Upgrade Winston schema.

```
% java -cp lib/winston.jar gov.usgs.volcanoes.winston.db.Upgrade
2015-12-09 06:23:45 INFO - Connected to database.
Current Winston schema version: 1.1.1
```

Available upgrade:
Winston schema up-to-date, no upgrades available.

Run with '--upgrade' option to perform an upgrade.
%

8 WWS COMMANDS

Commands can be sent to WWS by creating a TCP/IP connection and sending below commands over the socket. Winston commands take the form of a single line of text terminated by <CRLF>. The line of text

begins with a command name followed by a request id, separated by a space, and arguments, separated by space.

```
request = cmd SP req-id [ SP args ] CRLF
args = command-arg [ SP command-arg ]
    =/ channel-spec [ SP command-arg ]
req-id = 1*CHAR
channel-spec = scn1 [ SP time-span ]
scn1 = station SP channel SP network [ SP location ]
time-span = time-span-j2ksec / time-span-ew
time-span-j2ksec = j2ksec SP j2ksec
time-span-ew = unix-time SP unix-time
J2kSec = 1*DIGIT [ "." *DIGIT ] ; seconds since 2000-01-01T12:00:00+00:00
unix-time = 1*DIGIT [ "." *DIGIT ] ; seconds since 1970-01-01T00:00:00+00:00
```

The WWS protocol is descended from the protocol used by Earthworm's Wave Server (wave_serverV).

Table 15 Supported Earthworm Wave Server commands

MENU	Request listing of known stations and metadata request = "MENU" *1":" SP req-id [SP "SCNL"] CRLF If the SCNL argument is provided, location codes will always be included.
	response = req-id SP SP channel-record-list CRLF channel-record-list = channel-record *(SP SP channel-record) channel-record = pin SP scn1 SP time-span-ew SP data-type
GETSCNRAW	Alias for GETSCNLRAW. Either command will accept a SCN or SCNL and return data in the form it was received in.
GETSCNLRAW	request = "GETSCNLRAW" SP req-id SP scn1 SP time-span-ew CRLF response = header CRLF *tracebuf header = req-id SP pin SP scn1 SP "F" SP data-tpye ST time-span-ew length CRLF =/ req-id SP pin SP scn1 SP "FG" SP data-tpye CRLF =/ req-id SP pin SP scn1 SP "FR"SP data-tpye unix-time CRLF =/ req-id SP pin SP scn1 SP "FL"SP data-tpye unix-time CRLF If the request header contains the FR flags, then the requested time period is prior to all available data. No request body will be returned. If the request header contains the FL flags, then the requested time period is after available data. No request body will be returned. If the request header contains the FG flags, then the requested time period lies fully within a gap in data. No request body will be returned. If the request head contains only the F flag, then data was found, and a request body will be returned. If data is found, it will be returned following the header as a stream of tracebuf or tracebuf2 structures.
GETSCN	Alias for GETSCNL. Either command will accept a SCN or SCNL.
GETSCNL	request = "GETSCNL" SP req-id SP scn1 SP time-span-ew CRLF response = header CRLF *samples

```

header = req-id SP pin SP scnl SP "F" SP data-tpye ST time-span-ew
length CRLF
      =/ req-id SP pin SP scnl SP "FG" SP data-tpye CRLF
      =/ req-id SP pin SP scnl SP "FR"SP data-tpye unix-time CRLF
      =/ req-id SP pin SP scnl SP "FL"SP data-tpye unix-time CRLF

```

Winston commands consist of a sequence of characters optionally terminated by a colon. Times in Winston requests are specified as J2kSec (seconds from 1/1/2000 00:00:00).

Table 16 Winston commands

VERSION	Request WWS protocol version. This is the only WWS command which does not accept an ID argument. <pre><req> = "VERSION" <cr></pre> No response header. Response body: <pre><response> = "PROTOCOL_VERSION:" <sp> <protocol version><cr></pre>
STATUS	Retrieve server status. <pre><cmd> = "STATUS" <sp> <id> [<sp> <timeout>]</pre> If the timeout value is given, stations which do not have data within seconds will not be considered calculating median date age. The status command optionally takes a single floating-point number as its only argument. If a number is given, channels with no data within that number of seconds will not be used in determining the median data age of operational channels.
GETWAVERAW	<pre><cmd> = "GETWAVERAW" <sp> <id> <sp> <channel spec> <sp> <compress> <compress> = 0 1</pre>
GETCHANNELS	Retrieves a list of channels in Winston. <pre><cmd> = "GETCHANNELS" <sp> <id> [<sp> "METADATA"]</pre> The response will include one header line and a line for each channel in Winston. The Header line consists of two space-separated fields: <ul style="list-style-type: none"> • request id • number of channel lines Each channel line is a colon-separated string with the following values: <ul style="list-style-type: none"> • station id • \$-spearated SCNL • earliest sample as J2kSec • most recent sample as J2kSec • instrument longitude • instrument latitude • alias

- unit
- linear a
- linear b
- groups

The last five fields are only provided if the METADATA argument was provided with the request.

GETMETADATA	Get metadata associated with either instruments or channel. <cmd> = "GETMETADATA" <sp> <id> <sp> ("INSTRUMENT" "CHANNEL")
GETSCNLHELIRAW	Get raw helicorder data. <cmd> = "GETSCNLHELIRAW" <sp> <id> <sp> <scnl> <sp> <time span>
GETSCNLRAMRAW	Get raw RSAM data. <cmd> = "GETSCNLRAMRAW" <sp> <id> <sp> <scnl> <sp> <time span> <downsampling factor>

9 CLIENT APPLICATIONS

9.1 WINSTON WEB INTERFACE

If `www.allowHttp` is set to `true` in `WWS.config`, the web interface can be accessed through the browser using the following url format: `http://<server>:<port>`. The AVO Winston is a publicly available example: <http://pubavo1.wr.usgs.gov:16022/>. The web interface allows you to view server menu, helicorder plots, RSAM plots, server status, data gaps, and information on using FDSN web services specifications. More information on these features are available on the web interface.

9.2 SWARM

Swarm is a light-weight, Java-based application designed to display and analyze seismic waveforms. Swarm works with Winston and other data sources. For more information on Swarm and to download the software, visit <https://volcanoes.usgs.gov/software/swarm/index.shtml>.

10 SOURCE CODE

Winston source code is available through on USGS GitLab: <https://code.usgs.gov/vsc/winston>. Access to site is currently limited to account holders. Request account from a USGS employee.

Winston was previously hosted on GitHub. Outdated source code is available on <https://github.com/usgs/winston>.

Appendix A SQL NUGGETS

Maintenance

Give the Winston user database access:

```
GRANT ALL ON `W\_%\`.* to winstonuser@'localhost' identified by 'winstonpass';
```

Find channels which are missing in their database:

```
SELECT code from W_ROOT.channels WHERE CONCAT('W_', code) not in (SELECT schema_name
FROM information_schema.schemata);
```

Groups

Create a new top-level channel group called 'Networks':

```
INSERT INTO W_ROOT.groupnodes (parent, name, open) VALUES (0, 'Networks', 1);
```

Create a new subgroup of Networks called 'Augustine':

```
INSERT INTO W_ROOT.groupnodes (parent, name, open) SELECT nid, 'Augustine', 0 from
groupnodes WHERE name = 'Networks';
```

Add all broadband seismic channels at AU22 to Augustine group:

```
use W_ROOT;
INSERT INTO grouplinks(sid,nid) SELECT channels.sid, groupnodes.nid FROM channels,
groupnodes WHERE channels.code like 'AU22$BH%' and groupnodes.name='Augustine';
```

Remove 3 channels from the Okmok group:

```
use W_ROOT;
DELETE FROM grouplinks WHERE nid IN (SELECT nid FROM groupnodes WHERE name='Okmok')
AND sid IN (SELECT sid FROM channels WHERE code IN
('OKIF$ADC$AV','OKIF$BDF$AV','OKIF$HDF$AV'));
```

Appendix B MANUAL UPDATE OF STATION LOCATIONS

Some Winston clients, such as Swarm, can pull station locations from Winston if it is configured. Section 5.4 describes how station locations can be imported from dataless SEED volumes or Hypoinverse station files. For those who do not have these files readily available, locations can be manually configured.

The locations are configured in the instruments table (see section 6.1.5):

```
MariaDB [W_ROOT]> show columns from instruments;
```

Field	Type	Null	Key	Default	Extra
iid	int(11)	NO	PRI	NULL	auto_increment

name	varchar(255)	YES	UNI	NULL		
description	varchar(255)	YES		NULL		
lon	double	YES		-999		
lat	double	YES		-999		
height	double	YES		-999		
timezone	varchar(128)	YES		NULL		
+-----+-----+-----+-----+-----+-----+-----+						

Add data to the instruments table:

```
MariaDB [W_ROOT]> insert into instruments values (NULL, 'FG8','Fuego', -90.9359, 14.4325, 3636, NULL);
```

The iid will automatically be generated. Repeat for all other stations and check instruments table:

```
MariaDB [W_ROOT]> select * from instruments;
```

+-----+-----+-----+-----+-----+-----+-----+						
+-----+-----+-----+-----+-----+-----+-----+						
iid	name	description	lon	lat	height	timezone
+-----+-----+-----+-----+-----+-----+-----+						
+-----+-----+-----+-----+-----+-----+-----+						
16	FG8	Fuego, Guatemala	-90.9359	14.4325	3636	NULL
17	STG7	Guatemala	-91.58876	14.77487	3370	NULL
18	MOY	Guatemala	-90.07833	14.0495	1320	NULL
31	STG6	Guatemala	-91.526	14.80484	-999	NULL
+-----+-----+-----+-----+-----+-----+-----+						
+-----+-----+-----+-----+-----+-----+-----+						

Now that we have data in the instruments table, we can map the channels to the stations. The channels table (see section 6.1.1) will contain an iid column that is blank and needs to be populated.

```
MariaDB [W_ROOT]> show columns from channels;
```

+-----+-----+-----+-----+-----+-----+-----+						
Field	Type	Null	Key	Default	Extra	

sid	int(11)	NO	PRI	NULL	auto_increment	
iid	int(11)	YES		NULL		
code	varchar(50)	YES		NULL		
st	double	YES		NULL		
et	double	YES		NULL		
alias	varchar(255)	YES		NULL		
unit	varchar(255)	YES		NULL		
linearA	double	YES		1e300		
linearB	double	YES		1e300		

Update the channels table:

```
MariaDB [w_ROOT]> update channels a join instruments b on a.code like
concat(b.name, '$%$%') set a.iid=b.iid;
```

Then check channels table:

```
MariaDB [w_ROOT]> select a.code, b.iid, b.name, b.description from channels a join
instruments b on a.iid=b.iid;
```

code	iid	name	description	
FG3\$SHZ\$GI\$01	19	FG3	Fuego (analog), Guatemala	
FG3\$SHE\$GI\$01	19	FG3	Fuego (analog), Guatemala	
FG3\$SHN\$GI\$01	19	FG3	Fuego (analog), Guatemala	
FG8\$BHN\$GI\$00	16	FG8	Fuego, Guatemala	
FG8\$BHE\$GI\$00	16	FG8	Fuego, Guatemala	

Appendix C DETAILED INSTALL INSTRUCTIONS FOR CENTOS 7

Install MariaDB if not already installed:


```

yum install mariadb-server
systemctl enable mariadb
systemctl start mariadb
mysqladmin -u root password 'newpassword'
mysql -u root -p
mysql> CREATE USER 'winstonuser'@'localhost' IDENTIFIED BY 'password';
mysql> GRANT ALL PRIVILEGES ON . TO 'winstonuser'@'localhost' WITH GRANT OPTION;
mysql> CREATE USER 'winstonuser'@'%' IDENTIFIED BY 'password';
mysql> GRANT ALL PRIVILEGES ON . TO 'winstonuser'@'%' WITH GRANT OPTION;
mysql> flush privileges;

```

Download and install Winston from <https://volcanoes.usgs.gov/software/winston/index.shtml>

- Install under /opt/winston (e.g. /opt/winston/winston-1.3.14)
- Create install link to version (this makes it easier to point to a new version of Winston later)

```

cd /opt/winston
ln -s /opt/winston/winston-1.3.14 install

```

- Edit configuration files
 - WWS.config - Edit IP address (and port if necessary)
 - Winston.config - Edit database user and password
- Open port for external access:

```

firewall-cmd --permanent --zone=public --add-port=<port>/tcp
firewall-cmd --reload

```

- Access web interface via URL `http://<IP address>:<port>/`
- Configure instruments – see Appendix B
- Set up service for Winston to start on boot:
 - Create winston.service (see Appendix D) in /etc/systemd/system.
 - Run commands:

```

systemctl daemon-reload
systemctl start winston.service
systemctl enable winston.service

```

- Set up data imports (see section 5)

Appendix D EXAMPLE WINSTON.SERVICE FOR CENTOS7

[Unit]

Description = Winston

```

Wants                = network.target

After                = multi-user.target mariadb.service

Requires             = mariadb.service

[Service]

WorkingDirectory     = /opt/winston/install/

ExecStart             = /bin/sh -c '/opt/winston/install/bin/WWS.sh'

ExecStop             = /bin/pkill -9 `ps -ef | grep winston | grep -v grep | awk
'{print$2}`

StandardOutput       = null

StandardError        = null

User                 = winston

Group                 = winston

[Install]

WantedBy              = multi-user.target

```

Appendix E EXAMPLE IMPORTEW.SERVICE FOR CENTOS7

```

[Unit]

Description          = Winston Earthworm Importer

Wants                = network.target

After                = multi-user.target mariadb.service

Requires             = mariadb.service

[Service]

WorkingDirectory     = /opt/winston/install

EnvironmentFile       =

ExecStart             = /bin/sh -c '/bin/java -Xmx64M -cp
/opt/winston/install/lib/winston.jar gov.usgs.winston.in.ew.ImportEW
/opt/winston/install/ImportEW.config > /opt/winston/install/log/ImportEW.log 2>&1'

ExecStop             = /bin/pkill -f gov.usgs.winston.in.ew.ImportEW

```

```

ExecReload          =
StandardOutput      = null
StandardError       = null
User                = winston
Group               = winston

[Install]

WantedBy            = multi-user.target

```

Appendix F HYPONINVERSE STATION FILE FORMAT

FORMAT OF THE HYPONINVERSE STATION FILE FOR FULL 12-LETTER (S-C-N-L) CHANNEL NAMES

The channel identifier consists of a 5-letter site code, a 2-letter network code unique to the owner or operator of the network, a 3-letter component or channel code, and a 2-letter location code. The net, component, and location fields are optional to Hypoinverse, but space for these fields will be reserved in all input and output files. The user of Hypoinverse may specify (with the LET command) how many characters (counted from the left side) in each field to use when matching names in station and phase files. Full use of the fields will require 5, 2, 3, and 2 letters; minimum use will require 3, 0, 0 and 0 letters. Hypoinverse can use both a 1-letter (old USGS style) and 3-letter (SEED style) component field. Space for both fields will be reserved in all input and output files. Select whether to use the 1- or 3-letter field for matching station and phase names with the LES command. It is sometimes useful to keep the 1-letter component field for easier recognition by analysts than the newer, more general 3-letter field.

Cols.	Format	Data
1-5	A5,1X	Station site code. The first character may not be a number or the \$ character.
7-8	A2,1X	Station network code.
10	A1	Optional 1-letter component code for this channel.
11-13	A3,1X	3-letter component code for this channel.
15	A1	Station weight code (in units of 0.1) by which the weights assigned each P & S phase are to be multiplied. Use the digits 0-9 for the weight in tenths; "*" or "0" for no weight; or any other character (including blank) for full weight.
16-17	I2, 1X	Latitude, degrees.
19-25	F7.4	Latitude, minutes.
26	A1	N or blank for north latitude, S for south.
27-29	I3, 1X	Longitude, degrees.
31-37	F7.4	Longitude, minutes.

38	A1	W or blank for west longitude, E for east.
39-42	4X	Reserved for elevation in m. Not used by HYPOINVERSE.
43-45	F3.1, 2X	Default period (in sec) at which the maximum amplitude will be read for this station. Must be greater than 0.1. If period is given on phase card, it overrides this value.
48	A1	Put a "2" or "A" here to designate this as an alternate crustal model station. Both alternate and primary models must be in use. Stations may also be tagged for use with an alternate model in the delay file.
49	A1	Optional station remark field to copy to print output.
50-54	F5.2, 1X	P delay (sec) for delay set 1.
56-60	F5.2, 1X	P delay (sec) for delay set 2.
62-66	F5.2	Amplitude magnitude correction. If in the range +/-2.4, the correction is included (by addition) in the magnitude. If you don't want a station's magnitude used in the event magnitude, use a correction of 5.0 plus the actual correction. You can also assign a zero weight (see next).
67	A1	Amplitude magnitude weight code. Codes 0-9, "*" and blank are the same as the P & S weight codes (col 15). The actual magnitude weight used is the product of those on the station and phase cards. See also col 62.
68-72	F5.2	Duration magnitude correction (works the same as the amplitude magnitude correction).
73	A1	Duration magnitude weight code (works the same as the amplitude weight code).
74	I1	Instrument type code for this station used to select the appropriate response curve to derive an equivalent Wood Anderson amplitude. Must be either 0, 1 or 2: 0: Standard Wood-Anderson torsion seismograph. 1: USGS standard (1 HZ geophone, .7 critical damping.) 2: Hawaii-type Sprengnether seismometer.

Calibration factor for amplitude magnitudes, equal to peak-to-peak amplitude of a 10 microvolt RMS signal at 5 hz applied to the VCO and measured in mm on the Develocorder film viewer. For instrument types 0 and 2 this should generally be 1.0. A cal factor of 0.0 signifies an unknown response for which no amplitude magnitudes will be computed. If a cal factor is given on a phase card it overrides this value. The VCO attenuation may be given in place of the cal factor (see the ATN command). An entire history of station attenuations with the dates of attenuation changes may be read from a separate file with the ATE command.

```

81-82      A2      2-letter location code.
                  Earthworm follows the SEED convention of allowed
characters
                  (A-Z, 0-9, space).  In apparent contradiction to the
previous
                  statement, Earthworm uses the character '-' to represent
'space' in all Earthworm I/O. Just remember to think
                  when you see '-'. Lower case alphanumerics are not
permitted.
                  The location code should always contain 2 characters,
either
                  two alpha-numeric characters or two spaces ('--' or '
').
                  The use of a single space (or '-') adjacent to an alpha-
numeric
                  is not permitted.

```

Other details of hypoinverse configuration can be found in the complete Hypoinverse documentation maintained by Fred Klein at:

<ftp://ehzftp.wr.usgs.gov/klein/hyp2000/docs/hyp2000-1.0.pdf>

<ftp://ehzftp.wr.usgs.gov/klein/docs/shadow2000.pdf>