

Blacknest Bds Test Suite

User Manual – 1.2.1 - 2010-03-08

Project	Blacknest
Date	08/03/10
Reference	blacknest/bdsTestSuite
Author	Dr Terry Barnaby

Table of Contents

1. Introduction.....	1
2. Operation.....	1
3. Usage.....	2
4. Modifying The Tests.....	2
5. The Water Mark.....	2
6. Test Meta Data.....	2
7. Test Sensor Data.....	3

1. Introduction

The BDS Test suite is a system designed to test the basic operation and the data integrity of the BDS system. The system consists of a number of test programs and script files that allows an automated test of the system for regression tests when a new software release is produced and also to assist in implementing new features and to aid debugging of the system. The system generates test data with embedded MetaData information water marked into the data.

2. Operation

In order to function without affecting any of the data on a BdsServer, the system uses a test network, array, stations and channels. The system uses the following information for MetaData and Sensor data:

Network	TT
Array	TSA
Stations	TSB01 – TSB10, TSR01 – TSR10, TSM01 – TSM10, TSN01 - TSN10
Channels	BHZ_01
Source	Main

The system consists of the following programs:

- **bdsTestCreateMetaData:** This program deletes the test Meta data from the system, if present, and creates a new set based on the above parameters. It creates the network, array and stations as well as the channels and a set of default sensors and digitisers for the system.
- **bdsTestDeleteData:** This deletes the test seismic sensor data stored on the system.

- **bdsTestCreateData:** This program creates “water marked” test data in a number of different data formats including BDRS.
- **bdsTestImportDirectData:** This program creates data by directly writing it into the BDS system. It can create channel multiplexed and sample multiplexed data.
- **bdsTestValidateData:** This program reads a set of data from the BDS system and validates it based on the required data and the water marks contained in the data. It also checks that the MetaData is valid.
- **bdsTestImport:** This is a Python program to perform a test import of a set of test data. It makes use of the **bdsTestCreateData** and the **bdsImportData** programs to create and import a set of example test data.
- **bdsTestExport:** This is a Python program to export a set of test data from the BDS system and validate it with the **bdsTestValidateData** command.

3. Usage

In order to perform a test of the BDS system the following commands need to be run:

1. **bdsTestCreateMetaData:** Deletes and re-creates the test Meta Data
2. **bdsTestDeleteData:** Deletes the current test sensor data files.
3. **bdsTestImport:** This program creates a set of test data files and imports them into the BDS system.
4. **bdsTestExport:** This program reads a set of the test data and validates it.

If there are any errors the program will emit an error message and return an error status.

4. Modifying The Tests

It is relatively easy to implement new tests by modifying the **bdsTestImport** and **bdsTestExport** Python scripts or creating new versions of these. For more detailed test changes the C++ code in the other programs will need to be modified.

5. The Water Mark

In order to assist in the validation of data a “water mark” is inserted into each channels seismic sensor data block. The water mark consists of an ASCII string of comma separated parameters defining the data in the block. The current items described are: The network, the array, the station, the channel, the source the start and end times, the block size and a checksum of the data ignoring the water mark.

The water mark is inserted in the least significant byte of each data channel's sensor values. Thus the sensor data is still seen with some increased noise due to the “water mark”.

6. Test Meta Data

The test MetaData is designed to be relatively representative but with values that aid checking. The MetaData covers all time but there are changes in period 2008-02-01T00:00:00 to 2008-02-01T00:10:00 in order to test MetaData changes. The changes are:

<i>Time</i>	<i>Change</i>
2008-02-01T00:01:00	Station location change
2008-02-01T00:02:00	Calibration change
2008-02-01T00:03:00	Response change
2008-02-01T00:04:00	Instrument change
2008-02-01T00:05:00	Sensor Location change

7. Test Sensor Data

<i>Time</i>	<i>Description</i>
2008-01-01T00:00:00 - 2008-01-02T00:00:00	BDRS data 1 days worth
2008-01-02T00:00:00 - 2008-01-03T00:00:00	BDRS data 1 days worth
2008-01-03T00:00:00 - 2008-01-04T00:00:00	BDRS data 1 days worth
2008-01-04T00:00:00 - 2008-01-05T00:00:00	BDRS data 1 days worth
2008-01-05T00:00:00 - 2008-01-06T00:00:00	BDRS data 1 days worth
2008-02-01T00:00:00 - 2008-02-02T00:00:00	BDRS data 1 days worth
2008-02-02T00:00:00 - 2008-02-03T00:00:00	BDRS data 1 days worth
2008-02-03T00:00:00 - 2008-02-04T00:00:00	BDRS data 1 days worth
2008-02-04T00:00:00 - 2008-02-05T00:00:00	BDRS data 1 days worth
2008-02-05T00:00:00 - 2008-02-06T00:00:00	BDRS data 1 days worth
2008-03-01T00:00:00 - 2008-03-01T00:10:00	BDRS data 10 minutes worth
2008-03-01T00:10:00 - 2008-03-01T00:20:00	BDRS data 10 minutes worth
2008-03-01T00:20:00 - 2008-03-01T00:30:00	BDRS data 10 minutes worth
2008-03-01T00:30:00 - 2008-03-01T00:40:00	BDRS data 10 minutes worth
2008-03-01T00:40:00 - 2008-03-01T00:50:00	BDRS data 10 minutes worth
2008-03-01T00:50:00 - 2008-03-01T01:00:00	BDRS data 10 minutes worth
2008-04-01T00:00:00 - 2008-04-01T00:10:00	BDRS data 10 minutes worth with 1 missing block in every 10
2008-05-01T00:00:00 - 2008-05-01T00:10:00	BDRS-MM data 10 minutes worth (38 channels)

BEAM

<i>Time</i>	<i>Description</i>
2008-06-01T00:00:00 – 2008-06-01T00:20:00	Variable block size data like GCF. Two 10 minute segments with data block overlaps in some channels.
2008-07-01T00:00:00 – 2008-06-01T00:10:00	Sample multiplexed data like TapeDigitiser. Overlaps with segment below
2008-07-01T00:09:00 – 2008-06-01T00:19:00	Sample multiplexed data like TapeDigitiser. Overlaps with segment above