

## Blacknest Data System (BDS)

### Prototype Notes

<b>Project</b>	Blacknest
<b>Date</b>	2008-09-17
<b>Reference</b>	blacknest/bdsPrototypeNotes-1
<b>Author</b>	Dr Terry Barnaby

### Table of Contents

1. References.....	1
2. Introduction.....	1
3. Basic Usage.....	2
3.1. bdsAdminGui.....	2
3.2. BdsDataAccess.....	2
3.3. bdsImportOldData.....	2
3.4. bdsImportData.....	2
3.5. Web Access.....	3
3.6. Autodrm Access.....	3
4. BDS Installed Data.....	3
5. BDS Server Administration.....	3
6. BDS API.....	4
7. BDS SQL.....	4
8. Notes.....	4
8.1. Prototype System Notes.....	4
8.2. AIP Notes.....	5
8.3. SQL Notes.....	5
8.4. General Notes.....	5

### 1. References

- The BEAM Blacknest support website at: <https://portal.beam.ltd.uk/support/blacknest>. This provides detailed information on the BDS system and the current AutoDRM, information on alternative AutoDRM implementations and information on data formats.

### 2. Introduction

This document provides some notes on the Prototype BDS system. The bdsPrototypeOverview document should be read for more detailed information.

The BDS prototype system is a prototype. It is not fully featured and has been basically implemented. There will be a number of faults in the system. The purpose of the Prototype system is for Blacknest to get a more detailed idea on the proposed BDS system and to feedback any fundamental issues with the design.

The purpose of the prototype system is to allow Blacknest's staff to look at the overall design of the BDS

# BEAM

system and report on issues on the high level design. There will be a large number of missing features and bugs within the current system. The notes section at the end of this document lists the main issues.

The main items to look at are:

- The API. Will this give the information required.
- The SQL Schema. Does this look sensible.
- The BDS Data Format. Does this look Ok.

The BDS Server's root password is currently "beam00" and the MySQL's root password is "beam00".

## 3. Basic Usage

You can either ssh login to the BdsServer to test it out, or install the bds-clients package on your Centos5 or Fedora8 workstation. The main commands are listed below:

### 3.1. *bdsAdminGui*

This a simple GUI application that allow basic data and metadata information to be manipulated. It allows all of the available data and metadata to be viewed. The MetaData can be edited and verified. Siesmic data cane be viewed in graphical form. To run use the command "bdsAdminGui -host localhost".

### 3.2. *BdsDataAccess*

This is a simple command line program that allows access to the seismic data and associated meta data. Use the command without arguments to get a basic usage description. To download some data in BKNAS form you can use a command like:

```
bdsDataAccess -host localhost -select BN:EKA -startTime 2008-01-04T03:00:00 -endTime 2008-01-04T03:01:00 -command dataGetFormatted -format BKNAS -o data.bknas
```

### 3.3. *bdsImportOldDatabase*

This program is able to import all of the existing instrument response database into the BDS system. This has already be used so it is not that useful for testing unless a complete MetaData import is required.

### 3.4. *bdsImportData*

This command line program is able to import all of the existing seismic data files into the BDS system.

1. Data can be imported using the bdsImportData command line program. Initially this handles: BDRS, GCF and TapeDigitiser data.
2. The user will need to supply the file name, the data format and the network, station and channel names for each of the files data channels. (We could automate the channel names by either using a simple file database or adding a table to the BDS SQL. This would use the time and station name to decide the orientation of channels with the files).
3. The bdsImportData program will first validate the data. The following validations will be performed:
  - Generally check for data file corruption.

# BEAM

- Check that the blocks time-stamps are continuous and within an expected range.
  - Check that the date and time encoded in the file name matches with the time stamps within the data blocks.
  - Check that the sample rate is correct.
4. If the validation passes the data is uploaded into the BDS system. If it fails an error message is reported to the user. In this case the user can use the “-force” flag to upload the data with an appropriate description message.

There are some example scripts of its usage in /usr/bds/import. This code is quite rudimentary.

## **3.5. Web Access**

There is a webserver running on the BdsServer. You can access the BDS web interface using the /bds path. So a URL of the form <http://localhost/bds> will access the BDS web interface. This is very rudimentary and is there only as an example.

## **3.6. Autodrm Access**

There is a simple Autodrm daemon running. It should have been configured to use an email address of [bdsautodrm@somehost](mailto:bdsautodrm@somehost). It allows the following formats to be used: “BKNAS”, “IMS 1.0”, “ASCII”.

It does not support FTP at this time and is fairly rudimentary.

## **4. BDS Installed Data**

The BdsServer has about 60GBytes of example data from EKA (BDRS) and WOL (GCF) installed. There were quite a lot of time stamp verification errors during import so there will be quite a lot of gaps in the data.

The BDS Meta Data has been imported from the existing autodrm MySql database and should be complete as far as the original data was complete.

## **5. BDS Server Administration**

The BdsServer is running the following, BDS specific, daemons:

- Boapns: This is the BOAP name server daemon. It provides a name to IPAddress, port and service number lookup. It runs using port 12000 by default. It can be restarted using the command “service boapns restart”.
- BdsServer: This is the main BdsServer program. It runs using port 12001 by default. It can be restarted using the command “service bdsServer restart”. There is a configuration file /etc/bdsServer.conf to configure the server.
- BdsAutoDrm: This is the DS Autodrm program. It communicates with the BdsServer and currently must be started after the BdsServer. can be restarted using the command “service bdsAutodrm restart”. There is a configuration file /etc/bdsAutodrm.conf to configure the server. This includes the external POP and SNMP servers to use.

It is likely that the bdsServer and bdsAutodrm programs will need restarting occasionally due to crashes

# BEAM

and memory leaks in the prototype system.

## 6. BDS API

The BDS API can be viewed at: <https://portal.beam.ltd.uk/support/blacknest/files/bds/doc/bdsApi/html/index.html>

There are example programs in /usr/bds/bdsExamples.

## 7. BDS SQL

The BDS data is in the database BDS on the BDS Server. The BDS database can be accessed using the userid “bds” and password “bds”. There is a GUI database viewing tool called “kmysqladmin” available to view the data.

There is a separate database named BDS\_WEB which is solely for use by the PHP BdsWeb interface.

## 8. Notes

### 8.1. Prototype System Notes

1. There is no user or group security implemented as yet.
2. The system cannot yet handle accessing data across file boundaries. All requests should be within a single data file. Trying to access a large time period of data will thus fail.
3. Accessing a large time period of data in the Web interface will cause an out of memory exception in the PHP web interface.
4. Wild cards in the data selections are not fully implemented. They will use regular expression syntax.
5. Only supports ASCII, IMS 1.0 and BKNAS 1.0 external file formats as well as direct data access at the moment. Only limited tests have been done of the data formatters and there will be issues. There are quite a lot of issues with the BKNAS format in general, mainly due to it only really being able to handle data from a single array properly. At the moment the BDS has hard coded the Array name to be EKA.
6. Data import programs do not check for valid meta data.
7. There is no high level commands to allow user orientated Meta Data updates. At the moment the bdsAdminGui program only allows for manual database field updates. Higher level functionality such as “Add this set of channels with settings for this time range ...” need to be added. Blacknest will need to make a list of the main higher level functions that would be initially implemented. It is envisage that a simple scripting language would be used to implement these changes.
8. There is no support for Leap seconds.
9. The Autodrm is basic and does not support FTP downloads as yet.
10. The BdsWeb interface is basic and only an example at the moment.
11. The BDS Data format needs some more work. This is mainly concerned with the ability to add meta data on a data block basis for the TapeDigitiser data source.

# BEAM

12. There is no changes logged at the moment.

## **8.2. AIP Notes**

The API should be fairly complete although it is missing and user and group management. There does need to be some changes to the DataBlock format to support the TapeDigitisers extra meta data information on data blocks.

## **8.3. SQL Notes**

1. Station locations have array offsets in them. I think we need to add a special object for an array than contains the list of stations and their individual offsets. ie we want to make an Array a separate entity in the BDS system.
2. The Instrument/Digitiser/Sensor tables could possibly be merged.
3. Channels, Instruments, Digitisers and Sensors. At the moment multiple channels can share an Instrument, Digitiser and/or Sensor (Handled by ID) but there can only be one Instrument, Digitiser and Sensor per channel. If an Instrument change is made a separate channel needs to be created, with appropriate time period, with links to the different instruments. We could have a list of Instruments ID's in the channel or a separate channel/instrument linking table.

## **8.4. General Notes**

4. How to select when there is data overlap ? If data from the TapeDigitiser it would be best to select the later data file as the change may have happened due to head cleaning.
5. System requires Sample rate to be the same on all channels if data is asked for in a synchronously sampled block form. We could re-sample to the highest sample rate if required, with appropriate warning.
6. What if there are data holes within a time period ? Should we only allow data sets to be complete or would it be best to output 0's in the holes with appropriate warning messages ?
7. What happens when MetaData changes with the time period ? At the moment this is ignored. We could add certain meta data to the output data blocks and output blocks synchronised to the meta data change.
8. Need to add support for leap seconds. Probably best to have a flag in each data block which is set if a leap second change occurred within the block. The number of samples would not match the time stamp period by 1.
9. Import check for Metadata for time range on import.

## **8.5. Data Notes**

During import of existing data and checking of BKNAS and IMS data output formats we have noticed the following possible data issues with the current data:

1. Some of the Autodrm old database's floating point values seem to be truncated. This includes Latitude, Longitude, elevation etc.
2. Some of the Autodrm old database's time stamps look a little odd.
3. Quite a lot of the EKA and WOL data fails to import with time stamp validation errors. I have not looked into this as yet.

# BEAM

4. We have seen EKA BDRS files with a first block having timestamps that appear to be incorrect.