BEAM

Blacknest BDS Development

Programming Manual – 2.0.5

Project	Blacknest
Date	2013-02-04
Reference	blacknest/bdsDevelopment
Author	Dr Terry Barnaby

Table of Contents

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 information on developing programs for the BDS system.

For more information read the BDS Software Development presentation BdsDevInfo.odp and look at the BDS API overview BdsApiOverview.html and API documentation: bdsApi.

3. C++ API

TBD.

4. Python API

The BDS Python API is built on top of the standard BDS 'C++' API. Thus all of the standard BDS API documentation applies.

The Python language is interpreted rather than compiled and does not require strict types like 'C++'. This can help speed up the development of simple tools and programs, but it can result in less robust and less maintainable code.

To use the BDS API library import the “bdslib” or “bdslibe” modules. The “bdslibe” provides an exception based API as noted below.

The Python API is the same as the C++ API apart from a few coding style differences.

1. Reference returns: 'C++' allows references and/or pointers to be passed as function arguments which allows functions to return values. Python does not support this. Instead Python provides the

Directors: Dr T.K. Barnaby, M.J. Thomas
Registered Number: 2415065
Registered office: Beam Ltd, Northavon Business Centre, Dean Road, Yate, Bristol BS37 5NH, UK.
VAT Number: 520-2521-05
Web: www.beam.ltd.uk
Email: info@beam.ltd.uk

ability for functions to return multiple items. When using a BDS API call that, in 'C++', returns items by reference the Python equivalent will have these returned on the left hand side:

```
err = bds.channelGetList(selection, channels);        // C++

(err, channels) = bds.channelGetList(selection);      # Python
```

2. Testing for error returns. All BDS API calls return a BError object. This provides information as to if the function completed successfully or if there was an error. The BError object contains both an error number and an error string. 'C++' allows an “if” statement to have an asignment operator. This makes returning and checking errors quite concise. Python does not allow this and requires a separate assignment and iff statement. For example:

```
if(err = bds.channelGetList(selection, channels)){
        return err;
}

(err, channels) =  bds.channelGetList(selection);
if(err):
        return err;
```

3. Exceptions: The BDS API does not use 'C++' exceptions. As the BDS functions all return a BError object we have provided an alternative Python API library that uses exceptions for all BDS API calls instead of returning a BError object. With this calls can be written thus:

```
try:
        channels = bds.channelgetList(selection);
except ExceptionBError as e:
                print "Execption", e.number, e.string;
```

There are a number of BDS Python API examples in the /usr/bds/bdsExamples directory.

Directors: Dr T.K. Barnaby, M.J. Thomas
Registered Number: 2415065
Registered office: Beam Ltd, Northavon Business Centre, Dean Road, Yate, Bristol BS37 5NH, UK.
VAT Number: 520-2521-05
Web: www.beam.ltd.uk
Email: info@beam.ltd.uk