Futuristic Error Handling

Error handling in C++ today and tomorrow

Dawid Pilarski

dawid.pilarski@panicsofware.com

Introduction

Why am I here?

Why should we bother with error handling?

Recommendable error handling mechanism

Which error mechanism would you choose?

- error codes?
- exceptions?

Error codes nowadays

The error codes.

- Old. C-compatible. Comes from assembly time.

The error codes.

- Old. C-compatible. Comes from assembly time.
- Machine friendly.

The error codes.

- Old. C-compatible. Comes from assembly time.
- Machine friendly.
- Super fast.

The error codes.

- Old. C-compatible. Comes from assembly time.
- Machine friendly.
- Super fast.
- Used till today.

Error code example

```
int sqlite3_open( const char *filename, sqlite3 **ppDb );
```

Error code example

```
int sqlite3_open( const char *filename, sqlite3 **ppDb );
```

```
int open_status = sqlite3_open(/* ... */ );
if(open_status == SQLITE_OK){
  // make use of opened database
} else if( open_status == SQLITE_CANTOPEN_ISDIR ) {
  // handle the error
}
```

Handle the error

How to handle the error correctly?

Handle the error

How to handle the error correctly?

- `std::terminate()`

Handle the error

How to handle the error correctly?

- `std::terminate()`
- take the error callback

Handle the error

How to handle the error correctly?

- `std::terminate()`
- take the error callback
- propagate the error to the caller

Titleformats

Metropolis titleformats

Backup slides

Sometimes, it is useful to add slides at the end of your presentation to refer to during audience questions.

The best way to do this is to include the `appendixnumberbeamer` package in your preamble and call `\appendix` before your backup slides.

metropolis will automatically turn off slide numbering and progress bars for slides in the appendix.

References i