

# How to read from a file during execution

---

version	1.1.0
scope	Example. This code is provided as example code for a user to base their code on.
description	How to read from a file during execution
boards	Unless otherwise specified, this example runs on the SliceKIT Core Board, but can easily be run on any XMOS device by using a different XN file.

The xTIMEcomposer tools can handle system calls on behalf of the target application. This handling is enabled by default.

A file can be opened for reading as follows:

```
int fd = _open("test.txt", O_RDONLY, 0);
if (fd == -1) {
    printstrln("Error: _open failed");
    exit(1);
}
```

An open file can be read using the `_read` system call

```
_read(fd, readBuffer, BUFFER_SIZE);
```

We can then close an open file using the `_close` system call

```
if (_close(fd) != 0) {
    printstrln("Error: _close failed.");
    exit(1);
}
return 0;
}
```

Compile the above code, then create a file in the current directory named 'test.txt'. When the resulting executable is run, the first 16 bytes of the file will be read and the contents displayed on the console.

Note: In this case the raw system call functions directory is used as you are working in XC. However, if working in C then it might be advisable to the the C std library equivalents: `fopen`, `fread` and `fclose`.