

Application Note: AN10041

# How to use overlays to reduce application memory requirements

This application note is a short how-to on programming/using the xTIMEcomposer tools. It shows how to use overlays to reduce application memory requirements.

---

## Required tools and libraries

This application note is based on the following components:

- xTIMEcomposer Tools - Version 14.0.0

## Required hardware

Programming how-tos are generally not specific to any particular hardware and can usually run on all XMOS devices. See the contents of the note for full details.

## 1 How to use overlays to reduce application memory requirements

An overlay is a block of code and data that is loaded on demand at runtime. Each overlay has a predetermined region of memory that it is copied to called an overlay region. Several overlays may be associated with the same overlay region, but only one of these overlay can be loaded at any one time.

Overlays reduce the amount of memory needed to run your application since it is no longer necessary to reserve space for all your code and data - instead the tools only need to reserve space for the largest overlay that can be loaded into each overlay region.

Create an overlay by marking a function definition as an overlay root:

```
[[overlay]]
void print_hello()
{
    printstr("Hello");
}
```

This creates an overlay containing the function `print_hello`. The tools will automatically place code and read only data that is only referenced from `print_hello` in this overlay. The overlay is loaded on demand when `print_hello` is called. Calls to `print_hello` may be slow (since the overlay must be loaded), but once inside the overlay code will execute at full speed.

Create a second overlay by marking a second function as an overlay root:

```
[[overlay]]
void print_world()
{
    printstrln(" World!");
}
```

Call the functions in sequence:

```
int main()
{
    print_hello();
    print_world();
    return 0;
}
```

The two overlays will be automatically placed in the same overlay region since they are never in use at the same time. At the start of `main` no overlay is loaded. When `print_hello` is called the overlay runtime will load the overlay containing `print_hello` into the overlay region. When `print_world` is called the overlay runtime will load the overlay containing `print_world` into the same overlay region, replacing `print_hello` in memory.

Finally specify the option `-foverlay=syscall` you build the application. This links against an overlay runtime that loads overlays from a host machine using a system call. Applications that use this overlay runtime must either be run via JTAG using a debug adapter or be run in XSIM.

its use. Xmos Ltd. makes no representation that the Information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.