# Identity Paging

From OSDev Wiki

Identity Paging, Identity Mapped Paging and 1:1 Paging are terms often used on the forum to describe a design choice where a portion of virtual addresses are mapped to physical addresses that have the same value. This means that if paging is enabled with identity paging, `0xb8000` **is** `0xb8000`, as long as that area is identity mapped.

## Advantages

When switching to paged protected mode, your 1:1 mapping region doesn't care of whether paging is enabled or disabled. Placing your switching code and important data such as the core page directory and a few system page tables in this region gives you an easier way to set up paging without headaches.

## Example

Let's say you decide to use Identity Paging in the lowest 1MB. In this case vaddr `00000000..00000fff` are mapped to frame #00000, vaddr `00001000..00001fff` are mapped to frame #00001, and so on. (vaddr `000ff000..000fffff` are mapped to frame #000ff)

You can easily do this with a loop filling the page table:

```
void idpaging(uint32_t *first_pte, vaddr from, int size){
   from = from & 0xfffff000; // discard bits we don't want
   for(;size>0;from+=4096,size-=4096,first_pte++){
      *first_pte=from|1;     // mark page present.
   }
}
```

Retrieved from "http://wiki.osdev.org/index.php?title=Identity_Paging&oldid=17420"
Category: Memory management

- This page was last modified on 1 January 2015, at 13:28.
- This page has been accessed 18,247 times.