Page Frame Database

So far, we've talked about tracking virtual memory. The next piece of the puzzle is physical memory; specifically, how does the memory manager know which physical memory pages are free/used/corrupt, and so on?

The answer is the **Page Frame Database (PFD)**, which contains a representation of each page in physical memory. A page can be in one of a number of states, including:

- `Valid` –in use
- `Free` –available for use, but still contains data and needs zeroing before use
- `Zeroed` –ready for allocation.

As you can probably guess, the PFD is heavily used by the VMM.

So far we know that the memory manager uses:

- the VAD to keep track of virtual memory
- the PFD to keep track of physical memory.

But, as yet there is no way of translating between virtual addresses and physical memory! To do that, another mapping structure is required, called the **page table**, which maps virtual pages to their actual locations in memory and on disk.