In this directory you will find examples of ISAPI filters and extensions.

The filter loading mechanism works like this:

\* IIS loads the special Python "loader" DLL. This DLL will generally have a

leading underscore as part of its name.

\* This loader DLL looks for a Python module, by removing the first letter of

the DLL base name.

This means that an ISAPI extension module consists of 2 key files - the loader

DLL (eg, "\_MyIISModule.dll", and a Python module (which for this example

would be "MyIISModule.py")

When you install an ISAPI extension, the installation code checks to see if

there is a loader DLL for your implementation file - if one does not exist,

or the standard loader is different, it is copied and renamed accordingly.

We use this mechanism to provide the maximum separation between different

Python extensions installed on the same server - otherwise filter order and

other tricky IIS semantics would need to be replicated. Also, each filter

gets its own thread-pool, etc.