

OpenCore Sammelthread (Hilfe und Diskussion)

Beitrag von „cobanramo“ vom 12. Oktober 2024, 00:40

[Zitat von fabiosun](#)

How to fill Unload tab here i mean.

<https://github.com/mikebeaton/...OpenNetworkBoot/README.md>

Identifying missing network boot drivers

The `ip` command in LEFI Shell (e.g. `ipnetboot`) provided with OpenCore) is useful for working out which drivers are missing for network boot.

`ip -q Load(){}` shows available network boot entries. Handles with a device path ending in `(IPv4 or IPv6 address)` should indicate available PXE boot options. Handles with a device path ending in `(URL...)` should indicate available HTTP boot options.

Note 1: On some systems, there may be additional `Load(){}` handles with vendor-specific device paths. These may correspond, for instance, to GUI network boot options. These will not produce boot entries when using OpenNetworkBoot.

After identifying the handles for network boot entries, the other handles just before and after these, in the full list of handles displayed by `ip`, should correspond to the currently loaded network boot drivers. If there are no LoadFile options, then search in the full handle list for strings such as `tcp`, `udp`, `dhcp` (normally the native network boot drivers will appear grouped together). Examining the names printed by `ip` for these handles and comparing them to the available network boot drivers (see Network Boot Stack section) can be used to identify missing drivers.

Note 2: On systems with reasonably fast console text output, the `-H` option can be used with `ip` (as with most LEFI Shell commands) to display results one page at a time.

Note 3: For systems with slow console output, it may be more convenient to pipe the results of `ip` to a file on a convenient file system to examine later, within a booted OS. For example `ip > /tmp/ib_list` or:

```
ip > /tmp/ib_list
ip > /tmp/ib_list
```

HTTP Boot

On most recent firmware either no or only a few additional drivers are needed for HTTP boot, as most of the required drivers are already present in firmware.

After adding `ipnetboot`, if no HTTP boot entries are seen, try adding just the driver `ipnetboot`. If this does not produce network boot entries, try also adding `ipnetboot` and `ipnetboot`. If `http://` URLs can be booted but not `https://` try adding `TlsLib.efi`.

If the above steps do not work, proceed to the next section to identify which drivers are required.

Note 1: When using `https://` as opposed to `http://` URLs, one or more certificates, as required to validate the connection, must be configured on the network boot client. This can be done using OpenNetworkBoot's certificate configuration options, as documented in the [OpenCore documentation](#).

Note 2: In some firmware the existing `ipnetboot` driver may produce options which do not work correctly (e.g. blank screen when selected), because they are designed to work with a firmware UEFI which is not present when OpenCore is running). If so, in order to get working HTTP Boot options it may be necessary to use the `ipnetboot` config setting to `unload` the existing `ipnetboot` driver before loading the `ipnetboot` driver provided with OpenCore.

Note 3: In some firmware the existing `ipnetboot` and `ipnetboot` drivers may be locked down to `http://` URLs (even if the machine has no BIOS UEFI for HTTP Boot; e.g. Dell OptiPlex 3070). This means that while the `ipnetboot` from OpenCore can work with the native `ipnetboot`, it will only boot from `http://` URLs, giving a failure message otherwise. If `https://` URLs are required, this limitation can be worked around by using the `ipnetboot` config setting to `unload` the existing `ipnetboot` driver before loading the `ipnetboot` driver provided with OpenCore.

Note 4: During HTTP Boot 'Error: Could not retrieve MDP file size from HTTP server' is a very generic error message for 'something went wrong'. It could be that `https://` URLs are not allowed by `ipnetboot` or `ipnetboot`, or that a file does not exist at the specified URL on the server, or that the certificates (if any) stored in NVRAM could not be used to validate an `https://` URL, or any one of a number of other similar problems.





Mit dem aktuellen OpenCore Nightly Build wirst du kein Erfolg haben mit NetworkBoot, es ist noch nicht merged in die aktuelle Build.

Müsstest vorübergehend bis es publiziert ist auch mit diesem Artifact vom Link arbeiten..

<https://github.com/acidanthera.../actions/runs/11230753718>

Gruss Coban

EDIT: Bei mir sieht das ungefähr so aus, start über OpenCore zu Windows WDS Server...

Konnte zwar den Pxe ip6 nicht starten aber das liegt vermutlich an meiner Server konfiguration.

Http Boot funktioniert bei mir nicht aber das ist auch eine Konfigurationssache den ich auch noch nicht weitergekommen bin.



hier die Ablauf...

<https://youtu.be/53RrS1KZqd0>