NAME
    fdt - Flattened Device Tree support

SYNOPSIS
    options FDT
    makeoptions FDT_DTS_FILE=<board name>.dts
    options FDT_DTB_STATIC

DESCRIPTION
    Flattened Device Tree is a mechanism for describing computer hardware resources, which cannot be probed or self enumerated, in a uniform and portable way. The primary consumers of this technology are *embedded systems,* where a lot of designs are based on similar chips, but have different assignment of pins, memory layout, addresses bindings, interrupts routing and other resources.

    Configuration data, which cannot be self discovered in run-time, has to be supplied from external source. The concept of a flattened device tree is a platform and architecture independent approach for resolving such problems. The idea is inherited from Open Firmware IEEE 1275 device-tree notion, and has been successfully adopted by the embedded industry. The scheme works in the following way:

    ⊕   Hardware platform resources are *manually* described in a human readable text source format, where all non self-enumerating information is gathered.

    ⊕   This source description is converted *(compiled)* into a binary object i.e. a flattened device tree *blob* which is passed to the kernel at boot time.

    ⊕   The kernel (driver) learns about hardware resources details and dependencies from this [externally supplied] blob, which eliminates the need for embedding any information about the underlying platform hardware resources in the kernel.

    ⊕   The flattened device tree mechanism in principle does not depend on any particular first-stage bootloader or firmware features. The only overall requirement for the environment is to provide a complete device tree description to the kernel.

    The **fdt** layer allows any platform code in the kernel to retrieve information about hardware resources from a unified origin, which brings advantages to the embedded applications (eliminates hard-coded configuration approach, enforces code to be data driven and extensible) leading to easier porting and maintenance.

DEFINITIONS
    Device tree source (DTS)

The device tree source is a text file which describes hardware resources of a computer system in a human-readable form, with certain hierarchical structure (a tree). The default location for DTS files in the FreeBSD source repository is *sys/dts* directory.

Device tree blob (DTB)

The textual device tree description (DTS file) is first converted (compiled) into a binary object (the device tree blob) i.e. the DTB, which is handed over to the final consumer (typically kernel) for parsing and processing of its contents.

Device tree compiler (DTC)

A utility program executed on the host, which transforms (compiles) a textual description of a device tree (DTS) into a binary object (DTB).

Device tree bindings

While the device tree textual description and the binary object are media to convey the hardware configuration information, an actual meaning and interpretation of the contents are defined by the device tree *bindings*. They are certain conventions describing definitions (encoding) of particular nodes in a device tree and their properties, allowed values, ranges and so on. Such reference conventions were provided by the legacy Open Firmware bindings, further supplemented by the ePAPR specification.

BUILDING THE WORLD

In order for the system to support **fdt** it is required that FreeBSD world be built with the *WITH_FDT* build knob supplied either via src.conf(5) or command line defined with -D.

This creates the user space *dtc* compiler and enables **fdt** support in loader(8).

BUILDING KERNEL

There is a couple of options for managing **fdt** support at the FreeBSD kernel level.

makeoptions DTS+=<board name>.dts

Specifies device tree source (DTS) files for a given kernel. The indicated DTS files will be converted (compiled) into a binary form along with building the kernel itself. Any DTS file names not written as an absolute path must be specified relative to the default location of DTS sources i.e., *sys/dts*.

makeoptions DTSO+=<overlay name>.dtso

Specifies device tree source overlay (DTSO) files for a given kernel. Overlay files will be built with the kernel as with the makeoption *DTS* described above. Overlay files

specified as relative paths will be relative to the default location of DTS overlays for the platform being built i.e., *sys/dts/arm/overlays*.

options FDT   The primary option for enabling **fdt** support in the kernel. It covers all low-level and infrastructure parts of **fdt** kernel support, which primarily are the fdtbus(4) and simplebus(4) drivers, as well as helper routines and libraries.

makeoptions FDT_DTS_FILE=<board name>.dts

Specifies a preferred (default) device tree source (DTS) file for a given kernel. It will be built along with the kernel as if it were supplied via the makeoption *DTS* described above. This makeoption is not mandatory unless FDT_DTB_STATIC is also defined (see below).

options FDT_DTB_STATIC

Typically, the device tree blob (DTB) is a stand-alone file, physically separate from the kernel, but this option lets statically embed a DTB file into a kernel image. Note that when this is specified the FDT_DTS_FILE makeoption becomes mandatory (as there needs to be a DTS file specified in order to embed it into the kernel image).

SEE ALSO

fdtbus(4), openfirm(4), simplebus(4)

STANDARDS

IEEE Std 1275: IEEE Standard for Boot (Initialization Configuration) Firmware: Core Requirements and Practices (*Open Firmware*).

Power.org Standard for Embedded Power Architecture Platform Requirements (*ePAPR*).

HISTORY

The **fdt** support first appeared in FreeBSD 9.0.

AUTHORS

The **fdt** support was developed by Semihalf under sponsorship from the FreeBSD Foundation. This manual page was written by Rafal Jaworowski.