

Configuration Keys

This section describes keys that can be used in **.table** files in driver and system configuration bundles (in **/NextLibrary/Devices**). The configuration system and Configure application are described in Chapter 4.

Some keys can have several values, expressed as a space-delimited list. Space-delimited lists have one space between elements, with nothing before the first or after the last element.

Key values that specify addresses are expressed as ranges. Ranges include both the start and end address. If a range consists of a single byte, it's indicated by specifying the same start and end address—for example, `^0x0-0x0^`.

Driver Configuration Keys

The keys described in this section can be used in **.table** files in a driver's bundle. You can also specify your own keys. User and kernel modules alike can get the value of any key using the `IOConfigTable` class. Configure inspectors, which set some key values, use `NXStringTable` to do so; the `NXStringTable` corresponding to the instance configuration is available through the **table** instance variable of `IODeviceInspector`.

Here's an example of a default configuration file:

```
"Class Names" = "myTestDriver";
"Family" = "Example";
"Instance" = "0";
"Version" = "1.1";
"Driver Version" = "myTestDriver, 3.2 version, built by kw 8/20/93";
"DMA Channels" = "1";
"I/O Ports" = "0x0-0x0";
"IRQ Levels" = "2";
"Valid IRQ Levels" = "1 2 3 4";
"Memory Maps" = "0x20000-0x200ff";
"Server Name" = "myTestDriver";
```

See the driver bundles under **/NextLibrary/Devices** for more examples.

The following table shows the keys and explains when they must be used. Each key is explained in detail later in this section.

Key	Used For
^a Auto Detect IDs ^o	Drivers that support device auto detection
^a Auto Recovery ^o	IOTokenRing drivers
^a Block Major ^o	Drivers with UNIX block entry points; optional
^a Boot Driver ^o	Drivers that must be loaded at boot time
^a Bus Type ^o	Drivers that aren't EISA- or ISA-based
^a Character Major ^o	Drivers with UNIX character entry points; optional
^a Class Names ^o	All drivers that don't specify ^a Driver Name ^o
^a Default Table ^o	Instance tables only (inserted by Configure)
^a Display Mode ^o	Display drivers
^a DMA Channels ^o	Drivers that support DMA
^a Driver Name ^o	Alternative to the preferred ^a Class Names ^o
^a Driver Version ^o	All drivers

^a Family ^o	All drivers
^a Instance ^o	All drivers
^a I/O Ports ^o	Drivers that need access to I/O ports
^a IRQ Levels ^o	Drivers that support interrupts
^a Location ^o	All drivers; optional
^a Memory Maps ^o	Drivers that need access to mapped device memory
^a Post-Load ^o	Drivers that require user-level help after loading
^a PostScript Driver ^o	IOSVGADisplay drivers
^a Pre-Load ^o	Drivers that require user-level help before loading
^a Ring Speed ^o	IOTokenRing drivers
^a Server Name ^o	All drivers (inserted by Driver Kit makefiles)
^a Share IRQ Levels ^o	Drivers that use shared interrupts
^a SVGA PostScript Driver Extension ^o	Display drivers that require a special PostScript driver
^a Valid DMA Channels ^o	Drivers that support DMA; optional but recommended
^a Valid IRQ Levels ^o	Drivers that support interrupts; optional but recommended
^a Version ^o	All drivers
^a VGA Memory Maps ^o	Display drivers
^a 16Mb Early Token ^o	IOTokenRing drivers

Keys

Auto Detect IDs

Example: ^aAuto Detect IDs^o = ^aCPQ1234^o;
or
^aAuto Detect IDs^o = ^a0x71789004 0x0e111234^o;

This is a string used by Configure and installation software to identify hardware that can be controlled by the device driver. The string is a space separated list of *auto detect IDs*, each of which is an identifier that can be used to match a device connected to an I/O bus.

The auto detect ID contains both a vendor ID and a 16-bit device ID. An ANSI committee assigns vendor IDs; the vendor assigns device IDs. The auto detect ID takes the form of a 7 character string described in the EISA specification. It consists of two fields: *VVVdddd*, where *V* is an upper-case letter, and *d* is a hexadecimal digit. The three letters *VVV* represent the vendor code; the four digit hexadecimal number *dddd* represents the device ID. The combination of these two fields is guaranteed to be unique. For example, ^aCPQ^o is the vendor ID for Compaq, so an ID of ^aCPQ1234^o represents the Compaq device with device ID ^a1234^o.

The 7 character format is the preferred form of the auto detect ID. However, this ID can also be expressed as a 32-bit hexadecimal number. The vendor ID is translated into a 16-bit hexadecimal number; the device ID is the same as in the other format. The layout in this format differs for each bus type. For the EISA bus, the *device* ID is in the lower 16 bits, and the *vendor* ID is in the upper 16 bits. For the PCI bus, the *vendor* ID is in the lower 16 bits, and the *device* ID is in the upper 16 bits.

Auto Recovery

Example: ^aAuto Recovery^o = ^aYES^o;

Used in IOTokenRing drivers to specify whether the driver should support automatic recovery from errors. See the IOTokenRing class specification for more information.

Block Major

Example: ^aBlock Major^o = ^a1^o;

Used by some drivers with UNIX entry points to specify the device's block major number. See the

IODevice class specification for more information.

Boot Driver

Example: `^Boot Driver^;`

Specifies that the driver must be loaded at boot time. For example, SCSI controller drivers must typically be loaded at boot time so that the system can use the disks attached to the controller.

Bus Type

Example: `^Bus Type^ = ^PCI^;`

Indicates the type of bus the device uses. The current valid values are `^EISA^` (which includes ISA), `^PCI^` and `^PCMCIA^`. If the key isn't present or valid, it defaults to `^EISA^`.

Character Major

Example: `^Character Major^ = ^15^;`

Used by some drivers with UNIX entry points to specify the device's character major number. See the IODevice class specification for more information.

Class Names

Example: `^Class Names^ = ^FloppyController FloppyDisk^;`
`^Class Names^ = ^AHAController^;`

A space-delimited list of the classes in the relocatable object file that should receive **probe:** messages. This key is preferred to the `^Driver Name^` key, which may become obsolete.

Default Table

Example: `^Default Table^ = ^ATIUltraPro^;`

Automatically inserted into **Instancen.table** files by Configure when necessary. You should never have to specify this key.

Display Mode

Example: `^Display Mode^ = ^Width: 1024 Height: 768 Refresh: 76Hz ColorSpace: RGB:555/16^;`

Used by display drivers to specify the mode the display should be in. This key's value should be equivalent to one of the values assigned to the `^Display Modes^` key in the bundle's *Language.lproj/Localizable.strings* file.

The key's value should be of the form:

^Width:width Height:height ColorSpace:(BW:bits | RGB:xyz/w) Refresh:rate Hz^

White space and ordering are ignored, but correct capitalization is required. The color space parameter should be either **BW:** followed by the bits per pixel, or **RGB:** followed by the bits per color component and then the bits per pixel.

For example, the string shown below describes a display mode that's 800 pixels wide and 600 high, supports color at 16 bits per pixel (5 bits each of red, green, and blue per pixel), and has a refresh rate of 60 Hz.

```
Width: 800 Height: 600 ColorSpace: RGB:555/16 Refresh: 60 Hz
```

DMA Channels

Example: `^DMA Channels^ = ^2^;`

`^DMA Channels^ = ^3 7^;`

A space-delimited list of DMA channels that should be reserved for the device. You must specify default values with this key if your device performs DMA. The user can change the default values with the Configure application, subject to restrictions that you impose with the `^Valid DMA Channels^` key.

Driver Name

Example: `^Driver Name^ = ^AHAController^;`

This is obsolete; use the `^Class Names^` key instead. The `^Driver Name^` key is identical to the `^Class Names^` key, except that it doesn't allow you to specify more than one class.

Driver Version

Example: `^Driver Version^ = ^PROGRAM:Wingine PROJECT:displayDrivers-14 DEVELOPER:mflynn BUILT:NO DATE SET (-B used)^;`

`^Driver Version^ = ^myTestDriver, 3.2 version, built by kw 8/18/93^;`

A string uniquely identifying the driver version. In the future, the system may display this string when appropriate.

Family

Example: `^Family^ = ^Display^;`

The family the device belongs to. Configure uses this key to group devices and to make sure that all essential device families are represented. Valid values are listed in the table below.

Value	Configure View
<code>^Display^</code>	Display (at least one is required in the system configuration)
<code>^Pointing Device^</code>	Mouse (at least one pointing device is required)
<code>^Network^</code>	Network
<code>^SCSI^</code>	SCSI
<code>^Audio^</code>	Audio
<code>^Keyboard^</code>	Other (at least one keyboard is required)
<code>^Disk^</code>	Other

The `^SCSI^` value should be used only for SCSI controllers—not for SCSI devices such as tape drives. The `^Disk^` value should be used for both disks and disk controllers (except for SCSI controllers). For example, the IDE disk and IDE controller drivers (which are in the same relocatable object file) have the value `^Disk^` in their default configuration file.

Values besides those listed in the table above are permitted, but aren't treated specially. They're included in the Configure view labeled Other. Examples of other values in use include `^Parallel^` and `^Serial^`.

Instance

Example: `^Instance^ = ^0^;`

The instance number of this configuration file. Configure automatically specifies this key in **Instancen.table** files, but you should specify `^Instance^ = ^0^` in default files.

I/O Ports

Example: `^I/O Ports^ = ^0x170-0x177^;`
`^I/O Ports^ = ^0x3f8-0x3ff 0x2f8-0x2ff^;`

A space-delimited list of I/O port ranges that should be reserved for the device. You must specify

default values with this key if your driver uses I/O ports to get access to the device. If your driver uses Configure's default inspector, the user can change the starting address of the first range (but not the length of the range) using the inspector.

IRQ Levels

Example: `^IRQ Levels^ = ^1^;`
`^IRQ Levels^ = ^4 3^;`

A space-delimited list of interrupts (IRQs) that should be reserved for the device. You must specify default values with this key if your device interrupts. The user can change the default values with the Configure application, subject to restrictions that you impose with the `^Valid IRQ Levels^` key.

Location

Example: `^Location^ = ^Slot 3^;`

The location of the device. This string is set automatically by the device auto detection software and has a different format for each bus.

EISA
`^Slot n^` where *n* is replaced by a slot number, as in `^Slot 1^`.

PCI
`^Dev:d Func:f Bus:b^` where *d* is the device number,
f is the function number, and
b is the bus number;
`^Dev:6 Func:0 Bus:0^`, for example.

Memory Maps

Example: `^Memory Maps^ = ^0x0D0000-0xD3FFF^;`
`^Memory Maps^ = ^0xa0000-0xbfff 0xc0000-0xcfff^;`

A space-delimited list of memory ranges that should be reserved for the device. You must specify default values with this key if your driver needs access to mapped device memory. If your driver uses Configure's default inspector, the user can change the starting address of the first range (but not the length of the range) using the inspector.

Post-Load

Example: `^Post-Load^ = ^InstallPPDev^;`

A user-level program to be run just after the driver is loaded. In the example above, the executable file **InstallPPDev** is a file in the driver's bundle that installs the driver's device files.

PostScript Driver

Example: `^PostScript Driver^ = ^/usr/lib/NextStep/Displays/SVGA_psdrv^;`

Used by display drivers to specify the PostScript driver that matches them. IOFrameBufferDisplays don't specify this key, since they use the default PostScript driver. IOSVGADisplay drivers, however, must specify the SVGA PostScript driver, as shown above. See the IOSVGADisplay class description for more information.

Pre-Load

Example: `^Pre-Load^ = ^RemovePPDev^ ;`

A user-level program to be run just before the driver is loaded. In the example above, the executable file **RemovePPDev** is a file in the driver's bundle that removes the driver's old device files before the driver is loaded.

Ring Speed

Example: `Ring Speed = 4` ;

Used by `IOTokenRings` to specify the speed of the Token Ring. This must be either 4 or 16. See the `IOTokenRing` class specification for more information.

Server Name

Example: `Server Name = ATI`;

Indicates the name of this driver's bundle, minus the `.config` suffix. You shouldn't need to specify this key, since it's inserted automatically by the Driver Kit makefiles. For information on using the Driver Kit makefiles, refer to Chapter 4.

Share IRQ Levels

Example: `Share IRQ Levels = Yes`;

Indicates whether the device uses shared interrupts or not. On EISA and PCI systems, using shared interrupts implies using level-triggered interrupts. The value is either `Yes` or `No` with the default being `No`. Shared interrupts are not supported on ISA bus computers.

SVGA PostScript Driver Extension

Example: `SVGA PostScript Driver Extension = CirrusLogicGD542X_psdvr`;

Used by `IOSVGADisplay` drivers to specify the driver-specific module to be loaded into the SVGA PostScript driver. See the `IOSVGADisplay` class description for more information.

Valid DMA Channels

Example: `Valid DMA Channels = 0 1 3 5 6 7`;
`Valid DMA Channels = 2` ;

A space-delimited list of DMA channels that can be used by the device. When the user inspects the device, `Configure` automatically dims every DMA channel that isn't valid, so that the user can select only valid channels. See also the `DMA Channels` key.

Valid IRQ Levels

Example: `Valid IRQ Levels = 1`;
`Valid IRQ Levels = 11 12 14 15` ;

A space-delimited list of interrupts (IRQs) that can be used by the device. When the user inspects the device, `Configure` automatically dims every IRQ that isn't valid, so that the user can select only valid IRQs. See also the `IRQ Levels` key.

Note: IRQ 2 can't be used on ISA- and EISA-based machines, so it should never be in the `Valid IRQ Levels` list.

Version

Example: `Version = 1.0`;
`Version = 2.1`;

A floating point number that describes the version of this driver. In the future, the system may warn the user whenever the user attempts to install a driver that has a lower version than the already installed version of the same driver. By convention, the number before `.` should change only when the driver is incompatible (for user-level clients) from earlier versions. `Configure` display this version string.

VGA Memory Maps

Example: `^VGA Memory Maps^ = ^0xa0000-0xbfff 0xc0000-0xcfff^;`

A space-delimited list of memory ranges used for VGA access. Every display driver's default configuration table must include this key with the value `^0xa0000-0xbfff 0xc0000-0xcfff^`.

16Mb Early Token

Example: `^16Mb Early Token^ = ^YES^;`

Used in IOTokenRing drivers to specify whether the driver should support early token release. See the IOTokenRing class specification for more information.

System Configuration Keys

The keys described in this section are used in `.table` files in the system bundle. You don't usually have to specify any of the keys except perhaps the `^Kernel Flags^` key.

Active Drivers

Example: `^Active Drivers^ = ^EtherExpress16 ParallelPort ATI Beep^;`

Drivers to be loaded automatically and probed after boot time. Configure automatically adds drivers to either this list or the `^Boot Drivers^` list whenever the user adds a driver to the system configuration. By default, drivers are added to this list; if the default table contains the `^Boot Driver^` key, however, the driver is added to the `^Boot Drivers^` list.

Boot Drivers

Example: `^Boot Drivers^ = ^PS2Keyboard BusMouse DPT2012 IDE Floppy^;`

Drivers to be loaded and probed at boot time. See also `^Active Drivers^`, above.

Boot Graphics

Example: `^Boot Graphics^ = ^Yes^;`

Specifies whether graphics (instead of system messages) should be displayed during boot time.

Bus Type

Example: `^Bus Type^ = ^ISA^;`
`^Bus Type^ = ^EISA^;`

The system bus architecture. This key isn't currently used for the System Configuration.

Kernel

Example: `^Kernel^ = ^mach_kernel^;`

The name of the kernel to use.

Kernel Flags

Example: `^Kernel Flags^ = ^rootdev=sd1a^;`

Options to pass to the kernel.

Machine Name

Example: `^Machine Name^ = ^Dell 450DE/2 DGX^;`

The system manufacturer name/model. This key isn't currently used.

Version

Example: `^Version^ = ^1.0^;`

Used by the Configure application.