Boot Configuration Data (BCD) files provide a store that is used to describe boot applications and boot application settings. BCD files replace Boot.ini.

BCDedit is a command-line tool for managing BCD stores. BCDedit can be used for a variety of purposes, including creating new stores, modifying existing stores, and adding boot menu options. The BCDedit tool replaces the Bootcfg.exe in earlier versions of Windows®, and provides two major improvements:

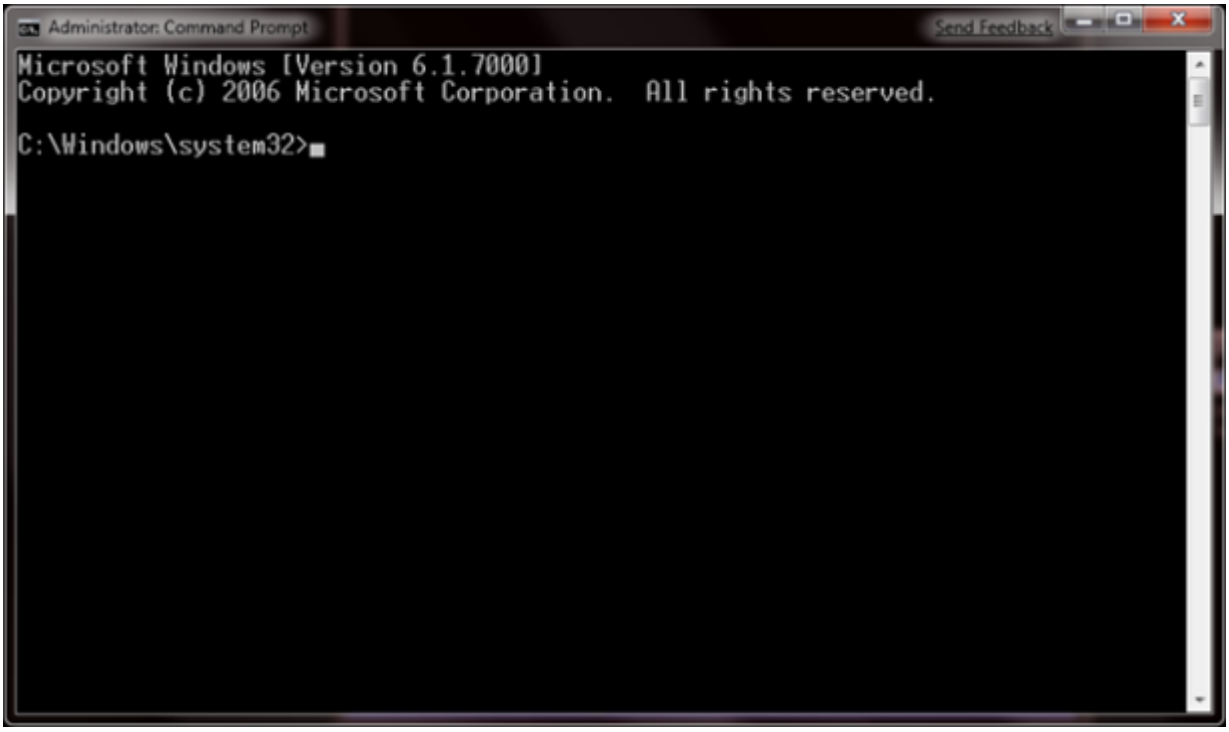
- BCDedit supports configuring a wide range of boot options.
- BCDedit improves scripting support.

Note
Administrative privileges are required to use BCDedit to modify BCD files.

BCDedit is the primary tool for editing the boot configuration of Windows Vista®, Windows® 7, and Windows Server® 2008. It is installed to the %WINDIR%\System32 folder.

BCDedit is designed primarily to perform single common changes to BCD; for example, enabling debugging, or setting the boot menu order. When working with more complex operations, consider using the BCD Windows Management Instrumentation (WMI) application programming interface (API) to create more powerful and flexible custom tools.

If you're going to work with BCDEDIT, you need to run it with Administrative Privileges. In a command prompt of course.



BCDEDIT is actually a little daunting at first when you run the BCDEDIT /? to see what it can do. It's powerful. Here is the output and what's in store for you.

BCDEDIT – Boot Configuration Data Store Editor

The Bcdedit.exe command-line tool modifies the boot configuration data store. The boot configuration data store contains boot configuration parameters and controls how the operating system is booted. These parameters were previously in the Boot.ini file (in BIOS-based operating systems) or in the nonvolatile RAM entries (in Extensible Firmware Interface-based operating systems). You can use Bcdedit.exe to add, delete, edit, and append entries in the boot configuration data store.

For detailed command and option information, type bcdedit.exe /? <command>. For example, to display detailed information about the /createstore command, type:

bcdedit.exe /? /createstore

For an alphabetical list of topics in this help file, run "bcdedit /? TOPICS".

Commands that operate on a store
==========
/createstore Creates a new and empty boot configuration data store.
/export Exports the contents of the system store to a file. This file

can be used later to restore the state of the system store.
/import Restores the state of the system store using a backup file
created with the /export command.
/sysstore Sets the system store device (only affects EFI systems, does
not persist across reboots, and is only used in cases where
the system store device is ambiguous).

Commands that operate on entries in a store
==========
/copy Makes copies of entries in the store.
/create Creates new entries in the store.
/delete Deletes entries from the store.
/mirror Creates mirror of entries in the store.

Run bcdedit /? ID for information about identifiers used by these commands.

Commands that operate on entry options
==========
/deletevalue Deletes entry options from the store.
/set Sets entry option values in the store.

Run bcdedit /? TYPES for a list of datatypes used by these commands.
Run bcdedit /? FORMATS for a list of valid data formats.

Commands that control output
==========
/enum Lists entries in the store.
/v Command-line option that displays entry identifiers in full,
rather than using names for well-known identifiers.
Use /v by itself as a command to display entry identifiers
in full for the ACTIVE type.

Running "bcdedit" by itself is equivalent to running "bcdedit /enum ACTIVE".

Commands that control the boot manager
==========
/bootsequence Sets the one-time boot sequence for the boot manager.
/default Sets the default entry that the boot manager will use.
/displayorder Sets the order in which the boot manager displays the
multiboot menu.
/timeout Sets the boot manager time-out value.
/toolsdisplayorder Sets the order in which the boot manager displays
the tools menu.

Commands that control Emergency Management Services for a boot application
==========

=====
/bootems Enables or disables Emergency Management Services for a boot application.
/ems Enables or disables Emergency Management Services for an operating system entry.
/emssettings Sets the global Emergency Management Services parameters.

Command that control debugging
==========
/bootdebug Enables or disables boot debugging for a boot application.
/dbgsettings Sets the global debugger parameters.
/debug Enables or disables kernel debugging for an operating system entry.
/hypervisorsettings Sets the hypervisor parameters.

"Oh my!" you eyes pop open as you wonder how you'll ever put a leash on this beast. But it's not so scary.

Really there's only a few options I've had to learn to work with to get the control I need.

Really. That's it.

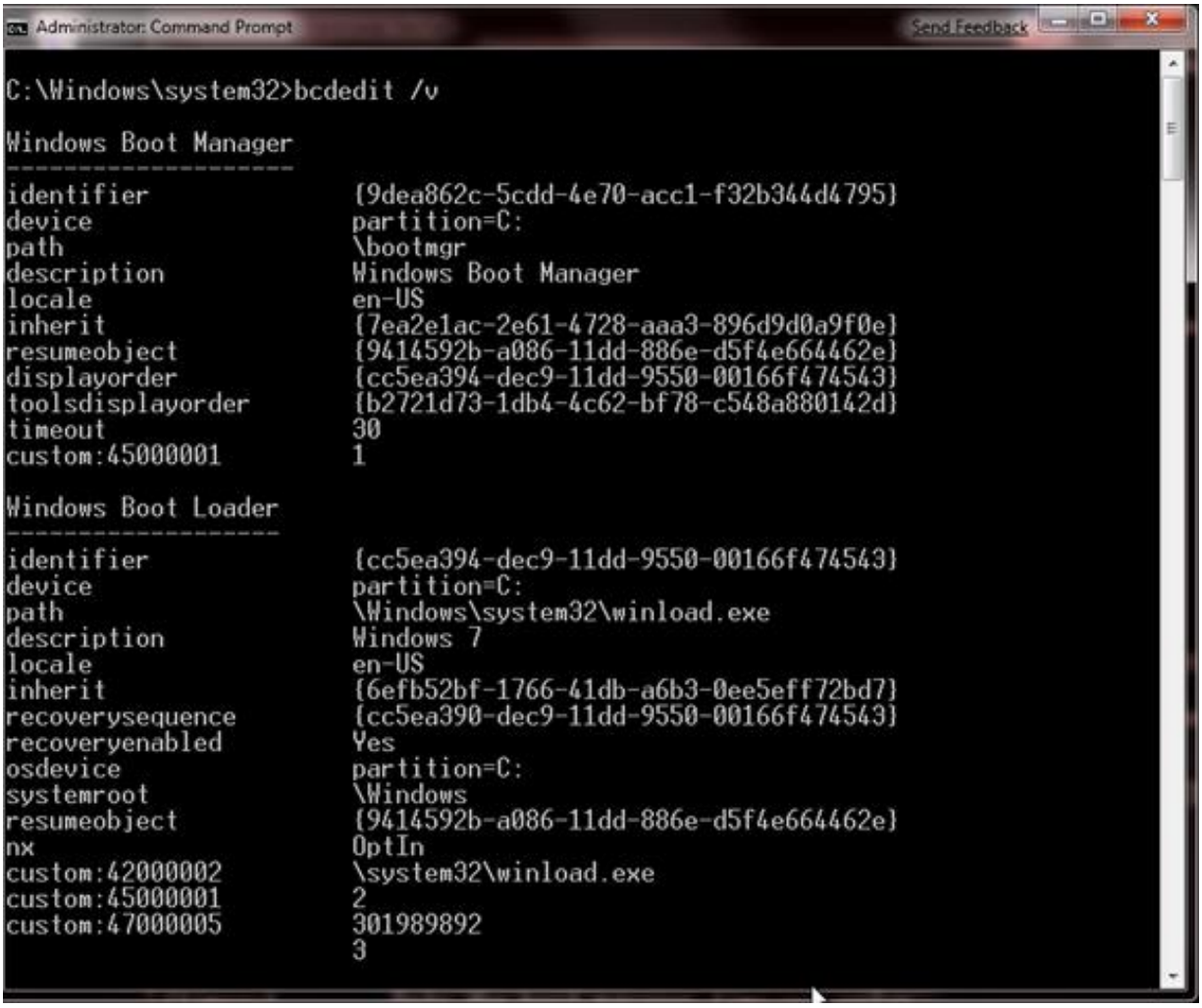
BCDEDIT /COPY (To make copy of an entry to work it)

BCDEDIT /DELETE (To delete an Entry)

BCDEDIT /SET (To set information within an entry)

So how do we use them? I'll try and keep this as simple as can be.

First you run in that Command Prompt a BCDEDIT /V to see what entries you do have.



What you're seeing is the equivalent to the old "BOOT.INI" The top half marked "Windows Boot Manager" is similar to the old [boot loader] section

[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)WINDOWS

The next section marked "Windows Boot Loader" is the equivalent to the individual lines you used to have under the [operating systems] entry in "BOOT.INI" like below

[operating systems]
multi(0)disk(0)rdisk(0)partition(1)WINDOWS="Microsoft Windows XP Professional" /fastdetect

---

So the basics.

First off, we're going to make a copy of an existing entry and work with that copy.

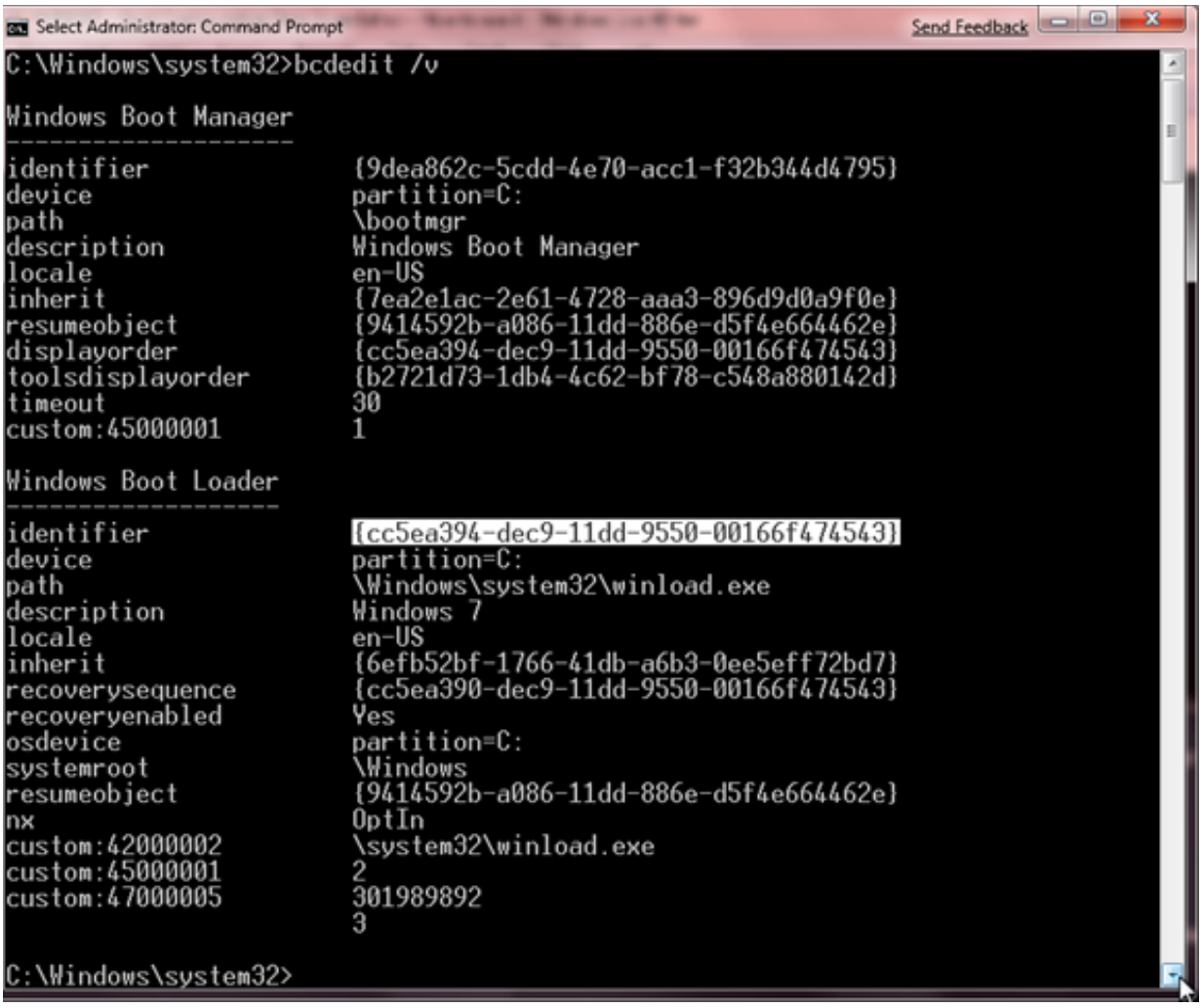
You'll notice in each entry displayed by BCDEDIT is a line marked "identifier". Since we're editing entries for bootable operating system, we're going to copy a "Windows Boot Loader" entry.

The process is dead simple. Execute a

BCDEDIT /COPY {SID} /D "Name of New Entry"

where the {SID} is the unique number to the right of the identifier line in the "Windows Boot Loader" entry you wish to copy and "Name of New Entry" is the description you wish to give that Entry.

Here's an example

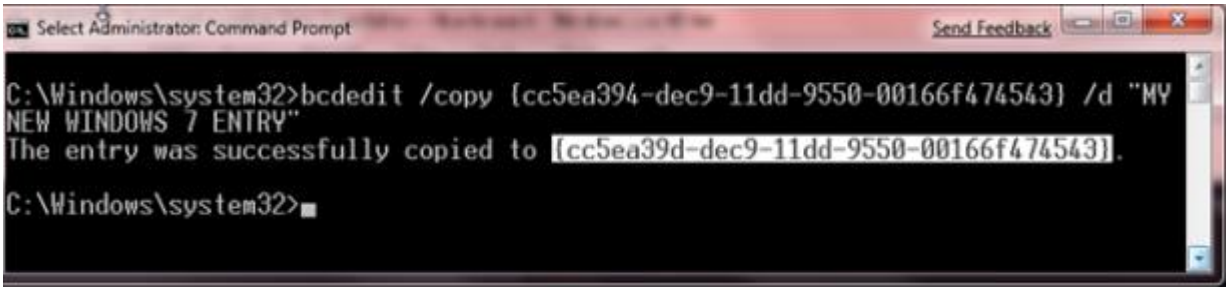


The line highlighted in WHITE is the {SID} for the Windows Boot Loader entry.

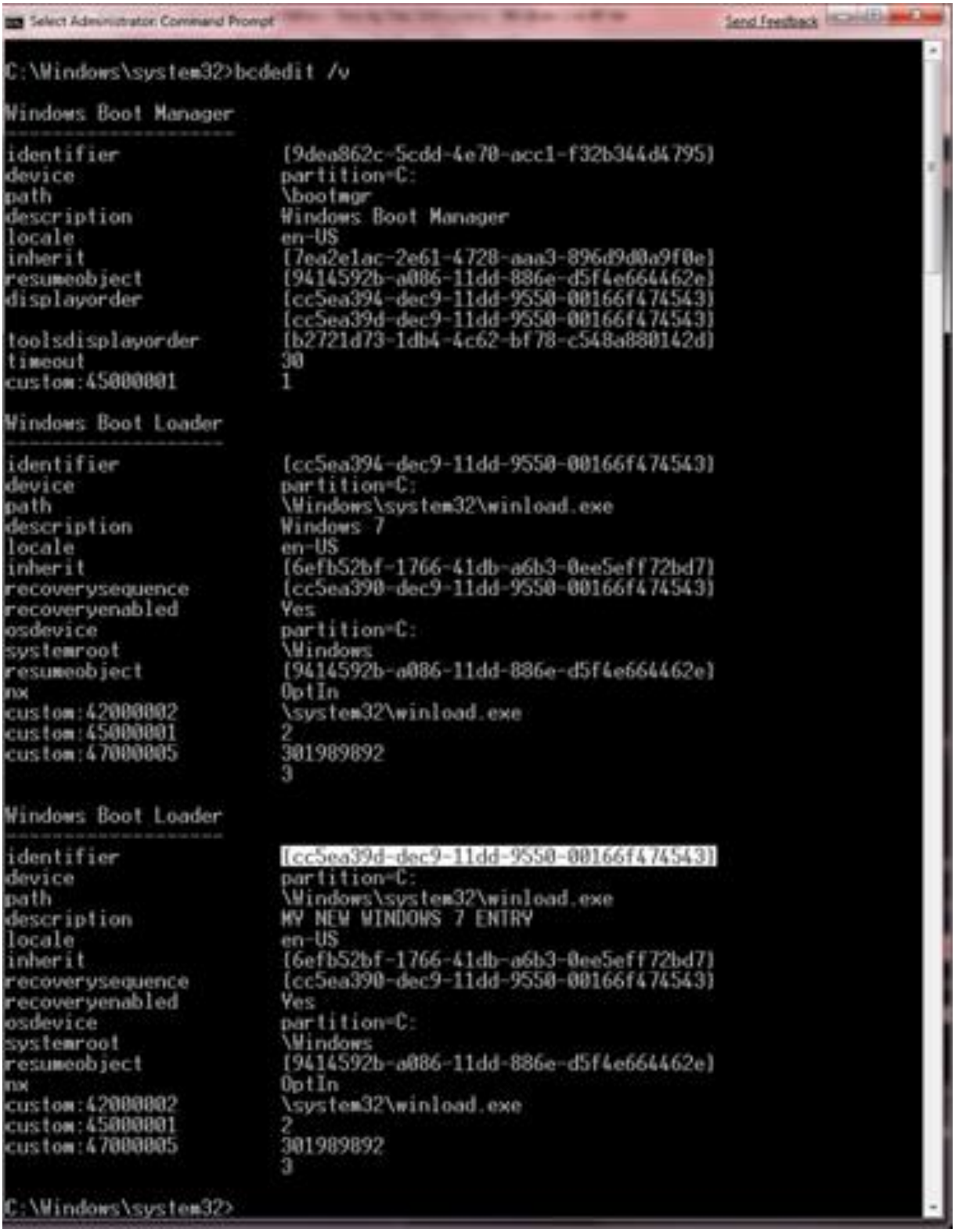
So to make a copy of that entry execute the following command

BCDEDIT /COPY {cc5ea394-dec9-11dd-9550-00166f474543} /D “MY NEW WINDOWS 7 ENTRY”

You’ll get the following result. Yes I wasn’t very creative with my naming was I? :)



The new entry will be identified by the line highlighted in White above. The end results can be viewed with a BCDEDIT /V command



The new entry is highlighted in White (Sorry, my command prompt may not be the prettiest environment in colours. But it is incredibly powerful) ;)

Setting or edit changes, now that you understand where the {SID} is, is quite easy.

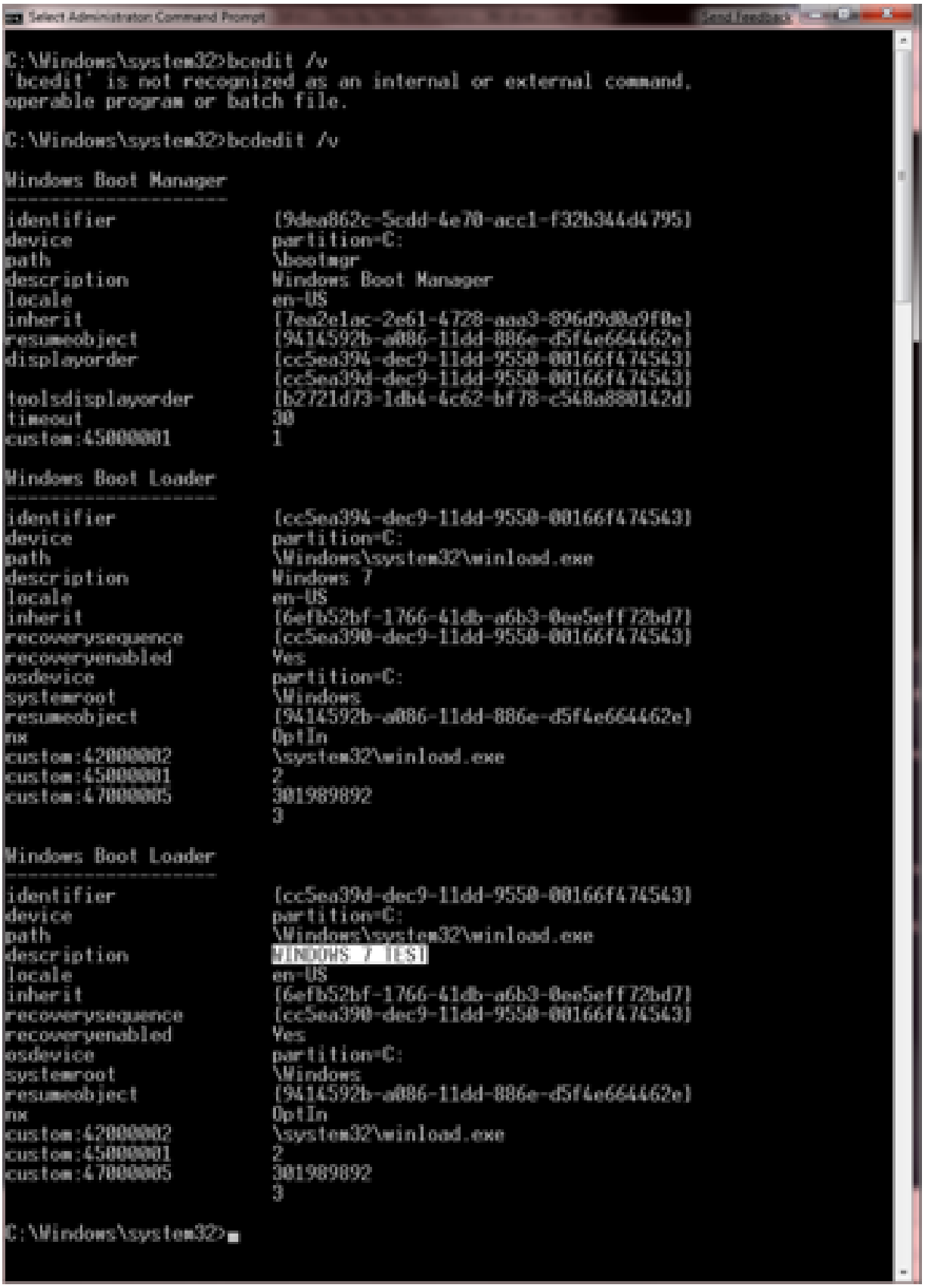
To change a value in this entry use the

BCDEDIT /SET {ID} <datatype> <value>

We're going to change the "description" from "MY NEW WINDOWS 7 ENTRY" to something more useful like "WINDOWS 7 TEST"

BCEDIT /SET {cc5ea39d-dec9-11dd-9550-00166f474543} description "WINDOWS 7 TEST"

You will find the entry is now changed. As can be seen below


Select Administrator: Command Prompt
Send Feedback
C:\Windows\system32>bcedit /v
'bcedit' is not recognized as an internal or external command,
operable program or batch file.
C:\Windows\system32>bcdedit /v
Windows Boot Manager
identifier {9dea862c-5cdd-4e70-acc1-f32b344d4795}
device partition=C:
path \bootmgr
description Windows Boot Manager
locale en-US
inherit {7ea2e1ac-2e61-4728-aaa3-896d9d0a9f0e}
resumeobject {9414592b-a086-11dd-886e-d5f4e664462e}
displayorder {cc5ea394-dec9-11dd-9550-00166f474543}
{cc5ea39d-dec9-11dd-9550-00166f474543}
toolsdisplayorder {b2721d73-1db4-4c62-bf78-c548a880142d}
timeout 30
custom:45000001 1
Windows Boot Loader
identifier {cc5ea394-dec9-11dd-9550-00166f474543}
device partition=C:
path \Windows\system32\winload.exe
description Windows 7
locale en-US
inherit {6efb52bf-1766-41db-a6b3-0ee5eff72bd7}
recoverysequence {cc5ea390-dec9-11dd-9550-00166f474543}
recoveryenabled Yes
osdevice partition=C:
systemroot \Windows
resumeobject {9414592b-a086-11dd-886e-d5f4e664462e}
nx OptIn
custom:42000002 \system32\winload.exe
custom:45000001 2
custom:47000005 301989892
3
Windows Boot Loader
identifier {cc5ea39d-dec9-11dd-9550-00166f474543}
device partition=C:
path \Windows\system32\winload.exe
description WINDOWS 7 TEST
locale en-US
inherit {6efb52bf-1766-41db-a6b3-0ee5eff72bd7}
recoverysequence {cc5ea390-dec9-11dd-9550-00166f474543}
recoveryenabled Yes
osdevice partition=C:
systemroot \Windows
resumeobject {9414592b-a086-11dd-886e-d5f4e664462e}
nx OptIn
custom:42000002 \system32\winload.exe
custom:45000001 2
custom:47000005 301989892
3
C:\Windows\system32>

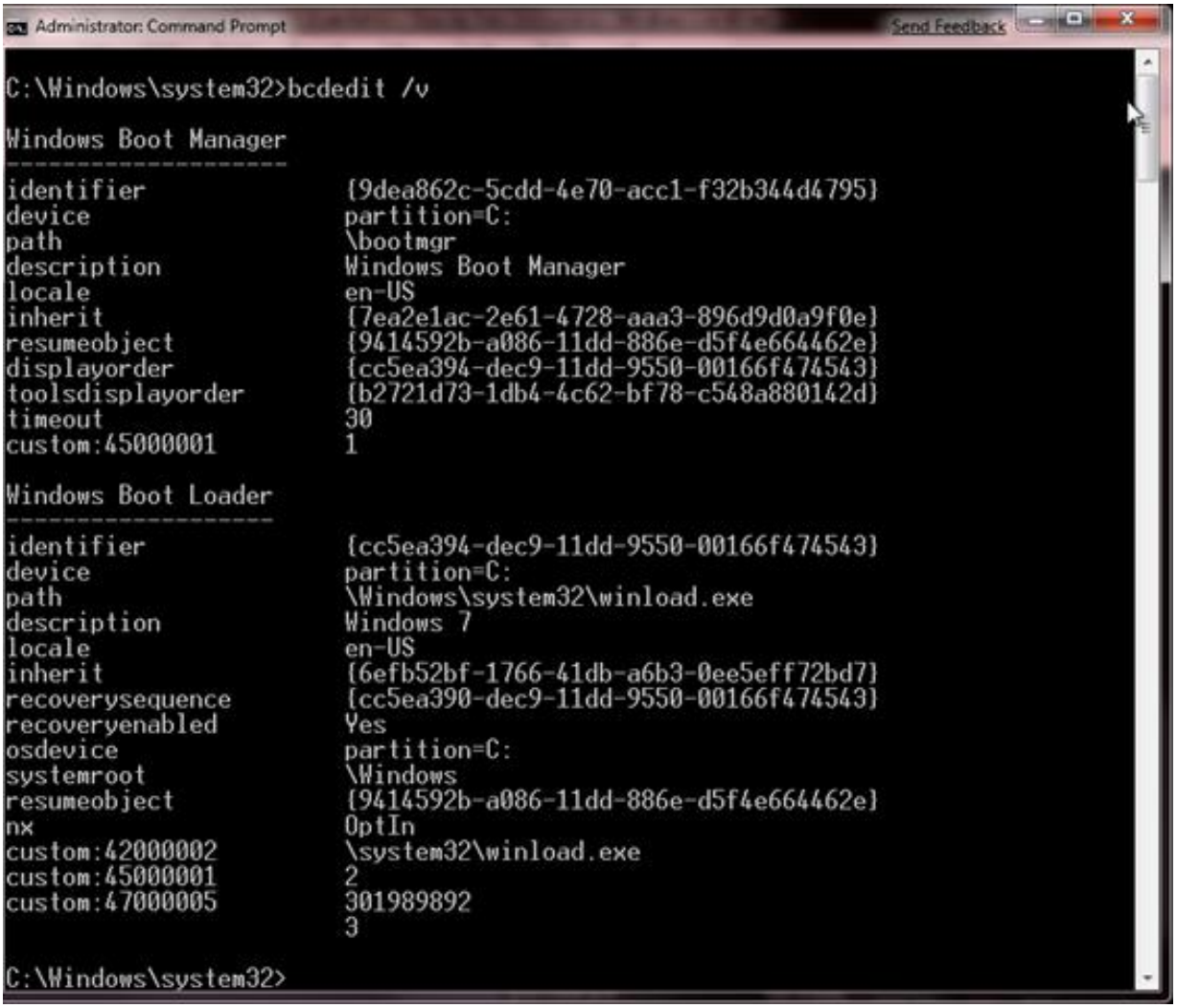
IF you wish delete this entire “Windows Boot Loader” entry just execute a

BCDEDIT /DELETE {ID}

Like so…

BCDEDIT /DELETE {cc5ea39d-dec9-11dd-9550-00166f474543}

and if you run a BCDEDIT /V you’ll find the list is now missing the new entry since we have deleted it.



And now all your work is gone. (or done depending on how you view it) Personally I prefer to use MSCONFIG to delete the entry. Somehow feels a little safer.