

Understanding Console Host Settings

Settings in the Windows Console Host can be a bit tricky to understand. This is mostly because the settings system evolved over the course of decades. Before we dig into the details of how settings are persisted, it's probably worth taking a quick look at what these settings are.

Settings Description

Name	Type	Description
FontSize	Coordinate (REG_DWORD)	Size of font in pixels
FontFamily	REG_DWORD	GDI Font family
ScreenBufferSize	Coordinate (REG_DWORD)	Size of the screen buffer in WxH characters**
CursorSize	REG_DWORD	Cursor height as percentage of a single character
WindowSize	Coordinate (REG_DWORD)	Initial size of the window in WxH characters**
WindowPosition	Coordinate (REG_DWORD)	Initial position of the window in WxH pixels (if not set, use auto-positioning)
WindowAlpha	REG_DWORD	Opacity of the window (valid range: 0x4D-0xFF)
ScreenColors	REG_DWORD	Default foreground and background colors
PopupColors	REG_DWORD	FG and BG colors used when displaying a popup window (e.g. when F2 is pressed in CMD.exe)
QuickEdit	REG_DWORD	Whether QuickEdit is on by default or not
FaceName	REG_SZ	Name of font to use (or “ DefaultTTFont ”, which defaults to whichever font is deemed most appropriate for your codepage)
FontWeight	REG_DWORD	GDI font weight
InsertMode	REG_DWORD	Whether Insert mode is on by default or not
HistoryBufferSize	REG_DWORD	Number of history entries to retain
NumberOfHistoryBuffers	REG_DWORD	Number of history buffers to retain

Name	Type	Description
HistoryNoDup	REG_DWORD	Whether to retain duplicate history entries or not
ColorTable%%	REG_DWORD	For each of the 16 colors in the palette, the RGB value of the color to use
ExtendedEditKey	REG_DWORD	Whether to allow the use of extended edit keys or not
WordDelimiters	REG_SZ	A list of characters that are considered as delimiting words (e.g. ' .-/\\= ,() [] {} ')
TrimLeadingZeros	REG_DWORD	Whether to remove zeroes from the beginning of a selected string on copy (e.g. 00000001 becomes 1)
EnableColorSelection	REG_DWORD	Whether to allow selection colorization or not
ScrollScale	REG_DWORD	How many lines to scroll when using SHIFT Scroll Wheel
CodePage	REG_DWORD	The default codepage to use
ForceV2	REG_DWORD	Whether to use the improved version of the Windows Console Host
LineSelection*	REG_DWORD	Whether to use wrapped text selection
FilterOnPaste*	REG_DWORD	Whether to replace characters on paste (e.g. Word “smart quotes” are replaced with regular quotes)
LineWrap*	REG_DWORD	Whether to have the Windows Console Host break long lines into multiple rows
CtrlKeyShortcutsDisabled*	REG_DWORD	Disables new control key shortcuts
AllowAltF4Close*	REG_DWORD	Allows the user to disable the Alt-F4 hotkey
VirtualTerminalLevel*	REG_DWORD	The level of VT support provided by the Windows Console Host

*: Only applies to the improved version of the Windows Console Host

**: WxH stands for Width by Height, it's the fact that things like a Window size store the Width and Height values in the high and low word in the registry's double word values.

The Settings Hierarchy

Settings are persisted to a variety of locations depending on how they are modified and how the Windows Console Host was invoked:

- * Hardcoded settings in conhostv2.dll
- * User's configured defaults (stored as values in `HKCU\Console`)
- * Per-console-application storage (stored as subkeys of `HKCU\Console`). Subkey names:
 - * Console application path (with `\` replaced with `_`)
 - * Console title
 - * Windows shortcut (.lnk) files

To modify the defaults, invoke the **Defaults** titlebar menu option on a Windows Console Host window. Any changes made in the resulting dialog will be persisted to the registry location mentioned above.

To modify settings specific to the current application, invoke the **Properties** titlebar menu option on a Windows Console Host window. If the application was launched directly (e.g. via the Windows run dialog), changes made in the dialog will be persisted in the per-application storage location mentioned above. If the application was launched via a Windows shortcut file, changes made in the settings dialog will be persisted directly into the .lnk file. For console applications with a shortcut, you can also right-click on the shortcut file and choose **Properties** to access the settings dialog.

When console applications are launched, the Windows Console Host determines which settings to use by overlaying settings from the above locations.

1. Initialize settings based on hardcoded defaults
2. Overlay settings specified by the user's configured defaults
3. Overlay application-specific settings from either the registry or the shortcut file, depending on how the application was launched

Note that the registry settings are “sparse” settings repositories, meaning that if a setting isn't present, then whatever value that is already in use remains unchanged. This allows users to have some settings shared amongst all console applications and other settings be specific. Shortcut files, however, store each setting regardless of whether it was a default setting or not.

Known Issues

- Modifications to system-created Start Menu and Win-X menu console applications are not kept during upgrade.

Adding settings

Adding a setting involves a bunch of steps - see `AddASetting.md`.