

# Technical Notes for CrystalClear Interface 2.8

Last Updated 06.10.13 for CCI 2.8.2

## System Requirements

CrystalClear Interface (CCI) 2.8 requires Mac OS X 10.6.8 ("Snow Leopard"), 10.7 ("Lion"), or 10.8 ("Mountain Lion"). System requirements are for an Intel system with at least 4GB of RAM running at least 2 processors with a minimum speed of 2.4GHz.

## Loading Mechanism

CCI loads using a ScriptingAddition, MTL.osax, which contains a "watcher" app, MTLAgent.app, that loads using a LaunchAgent when users log in. The launch agent is installed along with CCI and the MTL framework.

CCI runs as a plugin to the MarsThemes architecture, similar to — and technically indebted to — SIMBL. The core software is in a bundle called CCIFilter.bundle, located at /Library/Application Support/MarsThemes/Plugins

The watcher's function is to detect application launches, filter out apps that shouldn't load CCI, and load CCI during application launch. The software keeps track of which apps have been loaded using a property list (more on that in a moment).

## Application Resources

There are a wide variety of files in the CCIFilter.bundle's Resources folder, most of them graphics files, but two of these may be of interest to technical users who wish to further refine the way **CrystalClear Interface 2.6** handles a couple of tasks.

- *contrastOutlineViews.plist*. This property list contains a list of outline view subclasses used by various applications that load CrystalClear Interface. By default, CCI treats outline views in a manner that assumes they will keep their content's text color black even when it should change to white. For the subclasses listed in this file, CCI handles text contrast as if the view's text cells **can** change contrast from black to white (and vice versa) appropriately when the outline view's background color changes.

If you wish to extend the contrast color functionality to additional outline view subclasses, you can add a dictionary to this array, where the elements contain a key for the outline view classname, and a key for the application containing the subclass.

- *exceptedApps.plist*. This property list contains a few apps that would otherwise be excluded according to the *excludedApps* list (see next item).
- *excludedApps.plist*. By default, the MarsThemes Loader (MTL) (described below) prevents CCI from loading into applications with the following characteristics:
  - Background-only (daemon) apps that spawn no windows
  - Apps that have no identifier key
  - Apps that have no Info dictionary

In addition, MTL uses an *excludedApps* property list (stored in MTL's resources folder) to prevent loading for a range of individual applications, most of which are spawned by Mac OS X or by commonly used Apple apps (e.g., Safari). Many of these applications are already filtered out by the above rules, but many are not, because they are exceptions in one way or another to those rules. (The full contents of this property list is included at the end of these technical notes.)

Each MTL plugin, including CCI, maintains a separate *excludedApps* property list, specifying (mostly) third-party applications that are a problem for that particular plugin. This file is located in the CCIFilter.bundle's resources folder.

Users with the technical expertise to do so can edit the *excludedApps* property lists to add applications they believe should be excluded. (This has the same effect as designating an app as "disabled" within CCI itself, though it's a bit more robust.)

Beginning with support for Lion and Mountain Lion, there are now separate *excludedApps* property lists for Lion and for Mountain Lion, in addition to the default list. These lists exclude apps for those specific operating systems.

## Application Support Components

Besides the core components described above, CCI also installs and relies on the following files and helper applications. Each component is described in appropriate detail later in this section.

- An application support folder is installed at /Library/Application Support/MarsThemes/CCI/ This folder contains:
  - *CCI 2.6 User's Guide*, a PDF file containing the documentation available in HTML on the CrystalClear Interface website.
  - *CCI.app*
  - *CrystalDocslcons.plist*

- *IncompatibleList.plist*
- *Graphics* folder, which contains the graphics files used by CCI for theming.
- *Crystal Docs* folder, which contains all of the icons available for use with CCI.
- *Other Components* folder, which contains two applications used within CCI:

*QuickOpenCCIMenu.app*

*CCIUninstaller.app*

- *CrystalClear Interface.prefPane*, a system preference pane accessed through System Preferences, is installed at `/Library/PreferencePanes/`.
- CCI also makes use of some components common to all MarsThemes plugins. These items are stored in the `/Library/Application Support/MarsThemes/Common/` folder.
  - AquaticPrime.framework*
  - MTL (folder)*
  - styleForBlackGloss.plist*
- In `/Library/LaunchAgents`, CCI installs *org.musingsfrommars.CrystalMenubarLaunch.plist*.
- In the user's Desktop Pictures folder, CCI installs its set of *Frosted Crystal* desktop pictures, designed to complement the preset themes included with CCI. These pictures are left in place if CCI is uninstalled.
- CCI adds a folder to the user's `~/Library/Application Support/` folder to hold a copy of a purchased license and, when uninstalling CCI, the plist files the user may have modified while using CCI. This folder is located at `~/Library/Application Support/MarsThemes/CCI/`. (On Lion and Mountain Lion, the `~/Library` folder is invisible but can be accessed by using Finder's "Go To Folder..." menu option.)

## Global and Application Default Settings

CCI modifies some global and application default settings during installation and when users change button themes while using the software. The changes are designed to support CCI's theming and other functions. CCI maintains an "invisible" property list, *.appsWithEditedPlist.plist*, to keep track of all the apps that have defaults changed—both added and removed—while CCI is being used. This file ensures that CCI settings are removed from all application preferences when a user uninstalls CCI.

*NSUseLeopardWindowValues*. This string is set initially to "NO" during installation on Snow Leopard. It is further used to correctly theme the toolbars of Carbon applications, as well as of Cocoa applications the user has disabled for CCI, when the "Glossy" theme is turned on; in these cases, it is set to "YES." When an application is reenabled for CCI, this setting is toggled back to "NO." CCI deletes this application and global setting when it is uninstalled.

*AccessibilityAPIEnabled*. CCI toggles this setting on during installation to enable an applescript that shows new users where the CCI global menu is.

*AppleEnableMenuBarTransparency*. This setting is toggled off during installation to enable proper appearance of the Crystal Menubar.

*NSUseGrayBackground*. This setting is added to application defaults and set to "YES" for certain applications that require it to function properly (*Finder*, *iTunes*, and *InterfaceBuilder*).

*AppleUseCoreUI*. When users switch to the Black Gloss button theme, this setting is added to the preferences file for *Adobe Photoshop CS4* to ensure readable button text, especially on popup buttons.

## MarsThemes Loader

The MarsThemes Loader (MTL), based on the open-source code for SIMBL, consists of:

- *MTL.osax*, added to the `/Library/ScriptingAdditions` folder
- *MTLAgent.app*, launched from the resources folder of the MTL scripting addition
- *org.musingsfrommars.MTL.Agent.plist*, a LaunchAgent for MTLAgent
- *MarsThemes* folder, used as previously described. At `/Library/Application Support/MarsThemes`, a *Plugins* folder holds the core bundles for CCI and any other MTL plugins that the user may have.
- *MarsThemes Common* folder, within the MarsThemes folder, stores the AquaticPrime framework (which validates a user's license) and an MTL folder. This MTL folder contains two components that enable MTL to turn itself off (stop the MTL Agent from launching) if a user uninstalls CCI and has no other MarsThemes plugins installed.
- *.appsLoaded.plist*, stored in the MarsThemes application support folder, is updated whenever a user launches or quits an application, keeping track of which apps currently have CCI loaded.

## Components Details

### *CCI.app*

A small background application loaded at login that provides support for CCI's Single Application Mode (SAM) functionality.. It also incorporates the Sparkle framework for notifying users of software updates.

### *CrystalDocsIcons.plist*

A property list containing information on the Crystal Document icons available to be installed. This file gets updated as the user makes choices about installing/uninstalling different icon sets. When CCI is uninstalled, a copy of this file is stored in ~/Library/Application Support/MarsThemes/CCI so that it remembers the user's icon settings in case CCI is installed again later on.

### *IncompatibleList.plist*

A property list containing information on applications for which CCI is known to be incompatible in one way or another. This file contains a brief explanation of the problem when the user launches one of the listed applications. This file gets updated as the user encounters applications in the list.

### *Graphics folder*

In this folder, each set of graphics is stored in two zip files—one with the original graphics, and another with the CCI variants. The zipped original graphics are used during uninstall. The CCI variant folders contain subfolders corresponding to each button theme, as the button themes install some graphics specific to that theme. The list of files and folders is extensive, but briefly, this folder holds:

- *Applications*. This folder contains application-specific graphics for 27 applications, some from Apple and some from third-party developers. The complete list is shown at the end of these Notes.
- *CoreArt*. This folder contains the main system graphics files that implement the alternative button themes, menubars, and other systemwide items. In addition, when a user installs CCI, the installer stores a backup of the pre-CCI system graphics files that can be used to restore system graphics to the defaults as needed.  
In the *CCI* subfolder of *CoreArt* are four folders that contain graphics files specific to each button theme. Each button theme folder has the *ArtFile.bin* file (installed to the Apple *CoreUI* framework) and folders/subfolders corresponding to the *Extras2.rsrc* file (controlling various graphics, such as scrollbars, as well as the text color of the system menubar) and the *SArtFile.bin* file (controlling the menubar graphics among many others).
- *FontPanel*. This folder contains the Black Gloss graphics (introduced in Crystal Black) that are used to theme Apple's standard Font and Color Panels.
- *MenuExtras*. This folder has two subfolders that contain the white and black versions of both Apple and Third-Party icons for 60 applications accessed from the system menubar. (These graphics are commonly referred to as "menu extras.") A complete list of the applications covered by this folder is at the end of these Notes.
- *PrefPanels*. This folder is organized like that for *Applications* graphics and contains graphics for theming various "panes" in the System Preferences application, including *Bluetooth*, *Expose*, *Keyboard*, *Network*, *TimeMachine*, *Accounts* and *ParentalControls*.
- *System*. This folder has five subfolders, organized as described for *Applications* graphics.
  - *CoreServices*. Contains CCI graphics for the *login window* and alternative icons for Mac OS X "Core Types" of documents, toolbars, and other items.
  - *Developer*. Contains CCI graphics used for a few elements in *Xcode*.
  - *Frameworks*. Contains CCI graphics for the *AddressBook*, *Automator*, *ImageKit*, *PreferencePanels*, and *WebKit* frameworks.
  - *Other*. Contains CCI graphics for *Finder* and the *Character Palette* (Special Characters) application.
  - *PrivateFrameworks*. Contains CCI graphics for the *iLifeMediaBrowser* and *CoreWLANKit* frameworks, which are "private" Mac OS X frameworks.

### *AquaticPrime*

CCI uses the AquaticPrime framework for generating and authenticating user licenses. If this framework is not present on the user's system, it gets installed along with CCI. This folder, *AquaticPrime.framework*, gets installed at */Library/Frameworks/*

### *CCI Uninstaller*

*CCIUninstaller.app* is a Cocoa application that removes all components of CCI when run. It can be launched either from this folder or from the CCI global menu "Uninstall CCI". During uninstall, CCI also restores the default "Menu Extras" icons, and offers to restore any Crystal Docs icons the user may have installed. After uninstall, users are instructed to log out.

## Launch Agents

CCI uses two launch agents:

- *org.musingsfrommars.CrystalMenubarLaunch.plist* is used to launch — and keep running — *CCI.app* on user login.
- *org.musingsfrommars.MTL.Agent.plist* is used to launch — and keep running — *MTLAgent.app*.

## Crystal Document Icons

These icons are available in both compressed (.zip) and uncompressed (.icns) formats, and both formats include the original icons as well as the Crystal Document versions. The zipped files are used by CCI as users install and/or restore icons through the CCI Preferences window. Each contains two zip files—one with the original icons, and another with the CCI variants. The icons are stored in the `/Library/Application Support/MarsThemes/CCI/CrystalDocs/` folder.

## Uninstall Information

Considering all the components that CCI modifies or otherwise touches during installation and use, users are strongly discouraged from attempting a manual uninstall. If you cannot get the uninstaller to remove all components, your options are either to restore the system (relatively easy on Lion/Mountain Lion) or to try removing CCI manually. Steps for a manual uninstall would be:

1. *Restore core system graphics files.* The easiest way to restore the core graphics files is to change your button theme to "Aqua" while still running CCI. Otherwise, you can locate the "Original" Core Art files for your system as described above and copy them to the default locations. You will have to unzip the files first. The files `ArtFile.bin` and `SArtFile.bin` belong in `/System/Library/PrivateFrameworks/CoreUI.framework/Resources` The `Extras.rsrc` and/or `Extras2.rsrc` files belong in `/System/Library/Frameworks/Carbon.framework/Frameworks/HiToolbox.framework/Resources`
2. *Restore graphics files.* Within the Graphics tab in the CCI Preferences pane is a "Manage Graphics" tab that lets you restore original graphics for Applications, Preference Panes, and System frameworks. This is the only practical way to restore default graphics for these kinds of apps. To restore manually, you will find an "Original" subfolder within each category of graphics in the CCI application support Graphics folder. This folder contains zip files for all the graphics that need to be restored, except for Menu Extras. In general, these graphics files are stored within the application bundle's `Contents/Resources` folder; however, there are exceptions, and using the Manage Graphics functions will handle those for you.
3. *Restore menu extras graphics.* Menu Extras graphics, for both Apple and third-party apps, are stored in many disparate locations on your Mac. For third-party apps they are often stored in completely different folders from their parent applications. For this reason, your best option to restore menu extras graphics is to first switch to use of a menubar that requires the default (black) menubar graphics -- for example, the light opaque gradient menubar option. This will restore the default, black versions of your menubar graphics (with one or two exceptions).
4. *Restore document icons.* If you have installed the Crystal Documents icons during use of CCI, you must uninstall these as part of removing CCI from your system. In the Graphics/Icons tab, deselect all of the items that are checked, and this will restore the default document icons for your system.
5. *Restore system default settings.* Using the Terminal application (from Applications/Utilities) enter the following commands and hit return:
  - `defaults delete -g NSGrayBackground`
  - `defaults delete -g NSUseLeopardWindowValues`
  - `defaults delete -g AppleUseCoreUI`

Note that some of these settings are set within applications' preferences in addition to the global settings above. To find any remaining settings for individual apps, first identify the apps whose defaults need modifying by entering the following command in Terminal: `defaults find NSGrayBackground` (follow the same step for the `NSUseLeopardWindowValues` default)

If this command finds anything, you need to then remove `NSGrayBackground` from each app's defaults by entering a command like the following: `defaults delete com.apple.appstore NSGrayBackground`

6. *Remove CCI components.* You will need to authenticate to do so, but the last step is to remove all of the CCI components described earlier in this technical note.

- /Application Support/MarsThemes folder
- CCI Preference Pane
- MTL framework
- LaunchAgents

**Contents of excludedApps.plist as of CCI 2.8 (this covers all excludedApps lists described above):**

Apple	Third-Party
App Store Automator Launcher Dashboard Dashboard Client Dock Doc.Extra DVD Player FaceTime FeedbackServicesServer Folder Actions Dispatcher iCal Helper ImageCapture Extension iTunes Helper Metadata mds Metadata mdworker QTKit Server QuickLook Daemon Reminders (ML) Safari 6.0 (Lion/ML only) ScreenSaver Engine Speech Synthesis Server Spotlight SystemUIServer Web Process WebKit PluginAgent WebKit PluginHost WebKit PluginProcess	1PasswordAgent AMRunner (AutoPilot) ASRunner (AutoPilot) BackTrack5BA Canon Scanner Register CCI Crystal Menubar Chrome Helper Chronosync Backgrounder CleanApp Daemon DevonThink Sorter DragThing Evernote Helper FolderGlance Loader Google Keystone Agent Growl Helper Hardware Growler Hazelfolderwatch MTL Agent MyLivingDesktop Perian Photoshop (ML only) Quicken ScreenSieve SIMBL Agent ShortcutObserver SparkDaemon TaggingService TechToolPro Daemon TextSoap Agent TextExpander Daemon YouControlTunes Engine

**Applications with specific CCI graphics as of CCI 2.8:**

Apple	Third-Party
iCal iPhoto iWeb Keynote Mail Numbers Pages Preview Safari Terminal Xcode	1Password AutoPilot Camino CoverSutra DEVONagent DEVONthink Pro Endicia Evernote Leap LittleSnapper Platypus PlistEdit Pro Querious TextExpander VectorDesigner Xtorrent

**Menu Extras with white variants as of CCI 2.8:**

Apple	Third-Party
-------	-------------

Airport  
Battery  
Bluetooth  
Display  
Eject  
iChat  
Ink  
Keychain  
Notifications  
Script  
Spaces  
Spotlight  
Sync  
TimeMachine  
UniversalAccess  
User  
Volume  
VPN  
WLAN

1Password  
Alfred  
Anxiety  
AutoPilot  
Bowtie  
BusyCalAlarm  
Caffeine  
Chronosync  
Cloud  
Clusters  
CoverSutra  
Default Folder X  
DragThing  
DropBox  
DropCopy  
Evernote  
FastScripts  
GeekTool  
Growl  
Hazel  
Hyperspaces  
I Love Stars  
iClip  
iKey  
Last.fm  
Linkinus  
MagicPrefs  
MenuPop  
My Living Desktop  
Nocturne  
OpenMenu X  
Overflow  
Proxi  
ScreenFlow  
ShoveBox  
Tags  
TextExpander  
TotalFinder  
Twitterrific  
Voila  
XtraFinder