Extensis™

FONT MANAGEMENT CORE SCRIPTING GUIDE

Contents

Font Management Core Scripting

Creating scripts can be a great way to automate repetitive tasks across a database or to make mass changes when deployed to multiple computers.

The font management scripting functionality of Suitcase Fusion and Universal Type Server is called the CoreCLI (Command Line Interface). This process enables changes to the font management core (FMCore) process that runs in the background for both Suitcase Fusion and the Universal Type Client.

Installation

If you have installed and run Suitcase Fusion or Universal Type Client, the CoreCLI scripting functionality is automatically installed and enabled.

CoreCLI resides within the application package and is also installed in `/usr/bin` for Mac OS X installations. For Windows installations, CoreCLI is located in `Program Files\Extensis\Suitcase Fusion #\` (where # represents the Suitcase Fusion release number) or `Program Files\Extensis\Universal Type Client\`.

During the installation process, the appropriate global path settings are set on the target computer. When developing scripts, you do not need to designate global path variables to the CoreCLI.

Scripting language

You can use any preferred scripting language to create your scripts: Shell, Python and AppleScript on Mac OS X, or perhaps batch files, Python scripts or other languages on Microsoft Windows.

CoreCLI commands

Commands are available that affect most of the processes in the FMCore.

Designating the FMCore Mode

The FMCore is used as the back end of both the Universal Type Client and Suitcase Fusion. To execute CoreCLI commands properly, you must designate which mode the FMCore is currently running immediately before executing the command.

The `-standalone` designation tells the FMCore that you are running Suitcase Fusion, and to execute commands expecting this.

The `-clientserver` designation tells the FMCore that you are running the Universal Type Client and to execute commands expecting this.

Not all commands are supported across both modes. Each command description contains code examples for all supported modes.

For example, to execute the `handshake` command for Suitcase Fusion, you would use:

```
corecli -standalone -handshake
```

The same command for the Universal Type Client would be:

```
corecli -clientserver -handshake
```

To verify which mode the FMCore is currently running in, use the `getAgentProductMode` command.

getAgentProductMode

Use this command to verify which mode the FMCore is running. This command returns one of three possible values:

- `standalone` means the product is Suitcase Fusion;
- `clientserver` means the product is Universal Type Client;
- `thinclient` means the product is unknown!

If instead you receive an error message, it probably means the FMCore is not running. Start Suitcase Fusion or Universal Type Client, exit the application, and try again. This should start the FMCore.

Syntax:

```
-getAgentProductMode
```

Example:

```
corecli -getAgentProductMode
```

Workgroup (Library) Commands

Workgroups are the largest container in the database within which fonts can be placed. Workgroup commands can be used to create, delete, query or modify workgroups in Universal Type Server.

In Suitcase Fusion, workgroups are known as "libraries." Use the workgroup commands to modify libraries in Suitcase Fusion.

The following workgroup commands are supported by the font management core (FMCore) for both Suitcase Fusion and the Universal Type Client:

- ***deleteFontsFromWorkgroup*** (page 9)
- ***getFontWorkgroupMembershipInfo*** (page 9)
- ***getWorkgroupDataByIDs*** (page 9)
- ***getWorkgroups*** (page 10)

The command to create workgroups is only supported for use with Suitcase Fusion:

- ***createWorkgroup*** (page 8)

addFontsToWorkgroup

This command adds one or more existing fonts to a workgroup.

Syntax:

```
-addFontsToWorkgroup fontIds=numbers workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
fontIds	No	A comma separated list of font ids
workgroupId	No	A workgroup id

Suitcase Fusion Example:

```
corecli -standalone -addFontsToWorkgroup fontIds=1,2,3,4,5 workgroupId=44
```

Universal Type Client Example:

```
corecli -clientserver -addFontsToWorkgroup fontIds=1,2,3,4,5 workgroupId=44
```

createWorkgroup

Creates a library (workgroup).

NOTE: This command is only compatible with Suitcase Fusion.

Syntax:

```
-createWorkgroup name=string
```

PARAMETER	OPTIONAL?	DESCRIPTION
name	No	The name of the new library

Suitcase Fusion Example:

```
corecli -standalone -createWorkgroup name="my library"
```

deleteFontsFromWorkgroup

Removes fonts from a workgroup and sets. If any fonts are not part of the supplied workgroup, then the operation is a no-op for that font.

This command now reports progress as it deletes each font. If it encounters an error while deleting a font, it will send a notify error message in the progress callback. The error message will contain the font id and workgroup id plus the error code.

Syntax:

```
-deleteFontsFromWorkgroup workgroupId=number fontIds=numbers
```

PARAMETER	OPTIONAL?	DESCRIPTION
fontIds	No	A comma separated list of font ids
workgroupId	No	A workgroup id

Suitcase Fusion Example:

```
corecli -standalone -deleteFontsFromWorkgroup workgroupId=2 fontIds=1,2,5,6
```

Universal Type Client Example:

```
corecli -clientserver -deleteFontsFromWorkgroup workgroupId=2 fontIds=1,2,5,6
```

getFontWorkgroupMembershipInfo

Returns workgroup membership info for a font.

Syntax:

```
-getFontWorkgroupMembershipInfo fontID=[fontID]
```

PARAMETER	OPTIONAL?	DESCRIPTION
fontIds	No	The ID of the font to query for workgroup membership info.

Suitcase Fusion Example:

```
corecli -standalone -getFontWorkgroupMembershipInfo fontID=123
```

Universal Type Client Example:

```
corecli -clientserver -getFontWorkgroupMembershipInfo fontID=123
```

getWorkgroupDataByIDs

Returns details about a specified workgroup.

Syntax:

```
-getWorkgroupDataByIDs workgroupIds=list of IDs
```

PARAMETER	OPTIONAL?	DESCRIPTION
workgroupIds	No	A comma-separated list of workgroup IDs. This command will request information about those sets.

Suitcase Fusion Example:

```
corecli -standalone -getWorkgroupDataByIDs workgroupIds=40001,40015,40016
```

Universal Type Client Example:

```
corecli -clientserver -getWorkgroupDataByIDs workgroupIds=40001,40015,40016
```

getWorkgroups

Returns all of the workgroups in the local database. If FMCore is running in client/server mode then it will limit the workgroups returned to those the currently logged in user belongs to.

Syntax:

```
-getWorkgroups
```

Suitcase Fusion Example:

```
corecli -standalone -getWorkgroups
```

Universal Type Client Example:

```
corecli -clientserver -getWorkgroups
```

Set Commands

Set commands can be used to create, delete, activate, deactivate, query, move and modify sets.

The following set commands are supported by the font management core (FMCore) for both Suitcase Fusion and the Universal Type Client:

The following commands are restricted for use with the FMCore only on computers running the Universal Type Client:

activateServerPermActiveSets

This command activates all fonts that are designated as “startup sets” for the current user. The term “Server Permanently Active Set” should be considered synonymous with “Startup Set.”

NOTE: This command is available for use only with the Universal Type Client.

Syntax:

```
-activateServerPermActiveSets
```

Universal Type Client Example:

```
corecli -clientserver -activateServerPermActiveSets
```

activateStaticSets

Activates the fonts contained in the given static sets.

Syntax:

```
-activateStaticSets ids=string [mode=temporary-or-permanent] [system-override=true-
   or-false] [recursive=true-or-false]
```

PARAMETER	OPTIONAL?	DESCRIPTION
ids	No	Comma-separated list of static set ids.
mode	Yes	Activation mode specified to protect deactivation (e.g. a permanent activation can't be undone by a plugin deactivation). Valid values are [mode=temporary-or-permanent].
system-override	Yes	If set, attempts to override any conflicting system font.
recursive	Yes	If true, activate the child static sets as well.

Suitcase Fusion Examples:

```
corecli -standalone -activateStaticSets ids=2
corecli -standalone -activateStaticSets ids=2,12,21 mode=temporary system-
   override=true recursive=true
```

Universal Type Client Examples:

```
corecli -clientserver -activateStaticSets ids=2
corecli -clientserver -activateStaticSets ids=2,12,21 mode=temporary system-
   override=true recursive=true
```

addFontsToSets

Adds one or more fonts to one or more static sets.

Syntax:

```
-addFontsToSets fontIds=numbers setIds=numbers
```

PARAMETER	OPTIONAL?	DESCRIPTION
fontIds	No	A comma separated list of font ids
setIds	No	A comma separated list of set ids to add the fonts

Suitcase Fusion Example:

```
corecli -standalone -addFontsToSets fontIds=1,2,3,4,5 setIds=44,23,1293
```

Universal Type Client Example:

```
corecli -clientserver -addFontsToSets fontIds=1,2,3,4,5 setIds=44,23,1293
```

copySetsToSet

Copies static sets recursively to another static set. All fonts and child sets of the source sets will be copied to the destination set. The copied sets will become child sets of the destination set and it will inherit the same set type (shared or private) of the destination set.

Syntax:

```
-copySetsToSet srcSetIds=numbers destSetId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
srcSetIds	No	A comma separated list of set ids to copy
destSetId	No	The set id of the destination set

Suitcase Fusion Example:

```
corecli -standalone -copySetsToSet srcSetIds=23,24 destSetId=123
```

Universal Type Client Example:

```
corecli -clientserver -copySetsToSet srcSetIds=23,24 destSetId=123
```

copySetsToWorkgroup

Copies static sets recursively to a workgroup. All fonts and child sets of the source sets will be copied to the destination workgroup. The copied sets will become top-level sets in the destination workgroup and they will maintain its set type (shared or private).

Syntax:

```
-copySetsToWorkgroup srcSetIds=numbers destWorkgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
srcSetIds	No	A comma separated list of the set ids to copy
destWorkgroupId	No	The id of the destination workgroup

Suitcase Fusion Example:

```
corecli -standalone -copySetsToWorkgroup srcSetIds=23,26 destWorkgroupId=6
```

Universal Type Client Example:

```
corecli -clientserver -copySetsToWorkgroup srcSetIds=23,26 destWorkgroupId=6
```

createChildSet

Creates a child static set under an existing parent static set.

Syntax:

```
-createChildSet parentSetId=number name=string serverPermActive=true-or-false
```

PARAMETER	OPTIONAL?	DESCRIPTION
parentSetId	No	The id of the target static set (the parent for this new static set)
name	No	The name to assign to the set (cannot be empty)
serverPermActive	Yes	bool if this set is a startup set. (default false)

Suitcase Fusion Example:

```
corecli -standalone -createChildSet parentSetId=123 name="My New Child Set"
```

Universal Type Client Example:

```
corecli -clientserver -createChildSet parentSetId=123 name="My New Child Set"
```

createTopLevelSet

Creates a top level static set in a workgroup.

Syntax:

```
-createTopLevelSet workgroupId=number name=string type=string serverPermActive=true-
   or-false
```

PARAMETER	OPTIONAL?	DESCRIPTION
workgroupId	No	The id of the target workgroup (where to create the static set)
name	No	The name to assign to the set (cannot be empty)
type	No	The type to assign to the set: `private` or `shared`. When used with Suitcase Fusion, this must be set to `private`.
serverPermActive	Yes	bool if this set is a server perm active set. (default false)

Suitcase Fusion Example:

```
corecli -standalone -createTopLevelSet workgroupId=123 name="My New Top-Level Set"
   type="private"
```

Universal Type Client Example:

```
corecli -clientserver -createTopLevelSet workgroupId=123 name="New Top-Level Set"
   type="shared"
```

deactivateStaticSets

Deactivates the fonts contained within one or more sets.

Syntax:

```
-deactivateStaticSets ids=string [recursive=true-or-false]
```

PARAMETER	OPTIONAL?	DESCRIPTION
ids	No	Comma-separated list of static set ids.
recursive	Yes	If true, deactivate the child static sets as well

Suitcase Fusion Examples:

```
corecli -standalone -deactivateStaticSets ids=2
corecli -standalone -deactivateStaticSets ids=2 recursive=true
```

Universal Type Client Examples:

```
corecli -clientserver -deactivateStaticSets ids=2
corecli -clientserver -deactivateStaticSets ids=2 recursive=true
```

deleteSets

Deletes one or more sets including all child sets contained within the set.

Syntax:

```
-deleteSets setIds=numbers
```

PARAMETER	OPTIONAL?	DESCRIPTION
setIds	No	A comma separated list of set ids to delete

Suitcase Fusion Example:

```
corecli -standalone -deleteSets setIds=123,99,1,32,4,5
```

Universal Type Client Example:

```
corecli -clientserver -deleteSets setIds=123,99,1,32,4,5
```

exportSets

The `exportSets` command allows you to export the specified sets to a Set Description File, which also contains information about the fonts in each of the sets and subsets.

Syntax:

```
-exportSets path=destination_file_path setIds=numbers
```

PARAMETER	OPTIONAL?	DESCRIPTION
path	No	The path to the destination Set Description File. Any existing file in this location will be replaced.
setIds	No	A comma separated list of set ids to export

Suitcase Fusion Example:

```
corecli -standalone -exportSets path=/path/to/export_file.sdf setIds=1,2,5,6
```

Universal Type Client Example:

```
corecli -clientserver -exportSets path=/path/to/export_file.sdf setIds=1,2,5,6
```

getAllParentSetsRecursively

Returns the parent sets recursively up to the top-level set for a collection of one or more child sets.

Syntax:

```
-getAllParentSetsRecursively setIds=numbers
```

PARAMETER	OPTIONAL?	DESCRIPTION
setIds	No	A comma separated list of set ids to delete

Suitcase Fusion Example:

```
corecli -standalone -getAllParentSetsRecursively setIds=123,99,1,32,4,5
```

Universal Type Client Example:

```
corecli -clientserver -getAllParentSetsRecursively setIds=123,99,1,32,4,5
```

getChildSets

Returns the child sets of a parent set.

Syntax:

```
-getChildSets parentSetId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
parentSetId	No	The id of the parent set to query for all child sets

Suitcase Fusion Example:

```
corecli -standalone -getChildSets parentSetId=23
```

Universal Type Client Example:

```
corecli -clientserver -getChildSets parentSetId=23
```

getSetDataByIDs

Returns details about a set given the set IDs.

Syntax:

```
-getSetDataByIDs setIds=list of IDs
```

PARAMETER	OPTIONAL?	DESCRIPTION
setIds	No	A comma-separated list of set IDs

Suitcase Fusion Example:

```
corecli -standalone -getSetDataByIDs setIds=40001,40015,40016
```

Universal Type Client Example:

```
corecli -clientserver -getSetDataByIDs setIds=40001,40015,40016
```

getTopLevelSets

Returns the top level sets in one or more workgroups.

Syntax:

```
-getTopLevelSets workgroupIds=numbers
```

PARAMETER	OPTIONAL?	DESCRIPTION
workgroupIds	No	A comma separated list of workgroup ids to query for the top-level sets

Suitcase Fusion Example:

```
corecli -standalone -getTopLevelSets workgroupIds=23,1,18
```

Universal Type Client Example:

```
corecli -clientserver -getTopLevelSets workgroupIds=23,1,18
```

importSets

The `importSets` command allows you to import set hierarchies and associated fonts from a Set Description File which has been previously exported using the `exportSets` command. Depending upon the parameters given, you can either import the set hierarchies into a workgroup or into another set.

Syntax:

```
-importSets path=import_file_path [setID=number] [workgroupID=number]
```

PARAMETER	OPTIONAL?	DESCRIPTION
path	No	The path to the destination folder to place the collected fonts in.
setID	Yes	A set ID. If this is given, then it takes precedence over a workgroupID, if that is also given.
workgroupID	Yes	A workgroup ID. If a setID is given, then this is ignored.

Suitcase Fusion Examples:

```
corecli -standalone -importSets path=/path/to/export_file.sdf setID=7
corecli -standalone -importSets path=/path/to/export_file.sdf workgroupID=2
```

Universal Type Client Examples:

```
corecli -clientserver -importSets path=/path/to/export_file.sdf setID=7
corecli -clientserver -importSets path=/path/to/export_file.sdf workgroupID=2
```

moveSetsToSet

Moves static sets and all child sets to another static set recursively.

Syntax:

```
-moveSetsToSet srcSetIds=numbers destSetId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
srcSetIds	No	A comma separated list of the set ids to move
destSetId	No	The id of the destination set

Suitcase Fusion Example:

```
corecli -standalone -moveSetsToSet srcSetIds=23,24,25 destSetId=123
```

Universal Type Client Example:

```
corecli -clientserver -moveSetsToSet srcSetIds=23,24,25 destSetId=123
```

moveSetsToWorkgroup

Moves static sets and all child sets to a workgroup. The moved sets become top-level sets in the target workgroup.

Syntax:

```
-moveSetsToWorkgroup srcSetIds=numbers destWorkgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
srcSetIds	No	A comma separated list of the set ids to move
destWorkgroupId	No	The id of the destination workgroup

Suitcase Fusion Example:

```
corecli -standalone -moveSetsToWorkgroup srcSetIds=23,27,29,100 destWorkgroupId=6
```

Universal Type Client Example:

```
corecli -clientserver -moveSetsToWorkgroup srcSetIds=23,27,29,100 destWorkgroupId=6
```

queryForSetData

Queries the set data of one or more workgroups.

Syntax:

```
-queryForSetData workgroupIds=number [result-spec-file=string]
```

PARAMETER	OPTIONAL?	DESCRIPTION
workgroupIds	No	A comma-separated list of workgroup IDs.
result-spec-file	Yes	The path to a file containing an XML result spec.

NOTE: See ***XML spec file format*** on page 48 for appropriate file structure.

Suitcase Fusion Mac Example:

```
corecli -standalone -queryForSetData workgroupIDs=1,2,19 result-spec-
    file=/tmp/query.xml
```

Suitcase Fusion Windows Example:

```
corecli -standalone -queryForSetData workgroupIDs=1,2,19 result-spec-
    file=C:\temp\query.xml
```

Universal Type Client Mac Example:

```
corecli -clientserver -queryForSetData workgroupIDs=1,2,19 result-spec-
    file=/tmp/query.xml
```

Universal Type Client Windows Example:

```
corecli -clientserver -queryForSetData workgroupIDs=1,2,19 result-spec-
    file=C:\temp\query.xml
```

removeFontsFromSetAncestries

Removes fonts from one or more set ancestries.

Syntax:

```
-removeFontsFromSetAncestries fontIds=numbers setIds=numbers
```

PARAMETER	OPTIONAL?	DESCRIPTION
setIds	No	A comma separated list of the top-level set ids for the set ancestries to remove the fonts
fontIds	No	A comma separated list of font ids to remove from the set ancestries

Suitcase Fusion Example:

```
corecli -standalone -removeFontsFromSetAncestries setIds=123,99,1,32,4,5
    fontIds=1,2,3,4,5,33,2123,90
```

Universal Type Client Example:

```
corecli -clientserver -removeFontsFromSetAncestries setIds=123,99,1,32,4,5
    fontIds=1,2,3,4,5,33,2123,90
```

removeFontsFromSets

Removes fonts from sets.

Syntax:

```
-removeFontsFromSets fontids=string setids=string
```

PARAMETER	OPTIONAL?	DESCRIPTION
setIds	No	A comma separated list of the set ids
fontIds	No	A comma separated list of font ids to remove from the sets

Suitcase Fusion Example:

```
corecli -standalone -removeFontsFromSets setIds=123,99,1,32,4,5
    fontIds=1,2,3,4,5,33,2123,90
```

Universal Type Client Example:

```
corecli -clientserver -removeFontsFromSets setIds=123,99,1,32,4,5
    fontIds=1,2,3,4,5,33,2123,90
```

renameSet

Renames a static set.

Syntax:

```
-renameSet setId=number name=string
```

PARAMETER	OPTIONAL?	DESCRIPTION
setId	No	The id of the set to rename
name	No	The name to assign to the set (cannot be empty)

Suitcase Fusion Example:

```
corecli -standalone -renameSet setId=678 name="Tastes Like Chicken"
```

Universal Type Client Example:

```
corecli -clientserver -renameSet setId=678 name="Tastes Like Chicken"
```

setIsServerPermActiveSet

Sets the type value on a top-level set so that the set is a Startup Set.

NOTE: This command is available for use only with the Universal Type Client.

Syntax:

```
-setIsServerPermActiveSet serverpermactive=true-or-false setids=string
```

PARAMETER	OPTIONAL?	DESCRIPTION
setIds	No	A list of comma separated set ids
serverPermActive	No	Bool if this set is a startup set. (default false)

Universal Type Client Example:

```
corecli -clientserver -setIsServerPermActiveSet serverpermactive=true setids=1,2,3
```

setTopLevelSetType

Sets the type on top-level sets. The type is inherited by all child sets of the top-level sets.

NOTE: This command is available for use only with the Universal Type Client.

Syntax:

```
-setTopLevelSetType setIds=string type=string
```

PARAMETER	OPTIONAL?	DESCRIPTION
setIds	No	The IDs of the top-level sets to change
type	No	The type to set on the top-level static sets. The type must be `private` or `shared`.

Universal Type Client Example:

```
corecli -clientserver -setTopLevelSetType setId=678 type="private"
```

Font Commands

Font commands can be used to add, remove, export, move, and query fonts in the database.

The following font commands are supported by the font management core (FMCore) for both Suitcase Fusion and the Universal Type Client:

The following command is restricted for use with the FMCore only on computers running the Universal Type Client:

acquireFontFaceFiles

Tries to make available any missing asset files associated with a particular font face by downloading/unobfuscating as necessary.

NOTE: This command is available for use only with the Universal Type Client.

Syntax:

```
-acquireFontFaceFiles id=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
id	No	The font face ID number.

Universal Type Client Example:

```
corecli -clientserver -acquireFontFaceFiles id=231
```

activateFonts

Activate a list of fonts.

Syntax:

```
-activateFonts ids=string [mode=temporary-or-permanent-or-plugin-or-global-or-system-
   or-server] [system-override=true-or-false]
```

PARAMETER	OPTIONAL?	DESCRIPTION
ids	No	A comma separated list of ids of fonts to activate.
mode	Yes	Valid values are `temporary` or `permanent`.
system-override	Yes	If set, attempts to override any conflicting system font.

Suitcase Fusion Example:

```
corecli -standalone -activateFonts ids=2,12,21 mode=temporary
```

Universal Type Client Example:

```
corecli -clientserver -activateFonts ids=2,12,21 mode=temporary
```

addFonts

Adds fonts to the server and local database.

Syntax:

```
-addFonts paths="string","string","string" [targetWorkgroupID=string]
    [targetSetIDs=string] [location="Vault"-or-"LeftInPlace"] [addCorruptOnly=true-or-
    false] [autoCreateSets=true-or-false]
```

PARAMETER	OPTIONAL?	DESCRIPTION
paths	No	A comma separated list of font paths
targetWorkgroupID	Yes	Must be a valid workgroup id
targetSetIDs	Yes	List of sets to add the fonts to. Must all be in the target workgroup.
location	Yes	Default is Vault. [Vault | LeftInPlace]
addCorruptOnly	Yes	Default is No. For this to be true, targetWorkgroupID must be a localPersonalWorkgroup
autoCreateSets	Yes	Default to Yes. For this to be true, targetSetID must be zero

Suitcase Fusion Example:

```
corecli -standalone -addFonts paths=/Users/bobsmith/Desktop/Fonts/
    targetWorkgroupID=2
```

Universal Type Client Example:

```
corecli -clientserver -addFonts paths=/Users/bobsmith/Desktop/Fonts/
    targetWorkgroupID=2
```

deactivateFonts

Deactivate a list of fonts.

Syntax:

```
-deactivateFonts ids=numbers
```

PARAMETER	OPTIONAL?	DESCRIPTION
ids	No	A comma separated list of ids of fonts to deactivate.

Suitcase Fusion Example:

```
corecli -standalone -deactivateFonts ids=2,12,21
```

Universal Type Client Example:

```
corecli -clientserver -deactivateFonts ids=2,12,21
```

exportFonts

Exports fonts, sets, and smartsets to a destination folders.

Syntax:

```
-exportFonts path=destination_folder_path fontIds=numbers setIds=numbers
   smartSetIds=numbers workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
path	No	The path to the destination folder to place the collected fonts in
fontIds	Yes	A comma separated list of font ids that belong to the workgroup
setIds	Yes	A comma separated list of set ids that belong to the workgroup
smartSetIds	Yes	A comma separated list of smartset ids that belong to the workgroup
workgroupId	No	The workgroup id to whcih all the fonts are members

Suitcase Fusion Example:

```
corecli -standalone -exportFonts path=/TestPath/Test_Folder/Fonts fontIds=1,2,5,6
   setIds=1,2,5,6 smartSetIds=1,2,5,6 workgroupId=2
```

Universal Type Client Example:

```
corecli -clientserver -exportFonts path=/TestPath/Test_Folder/Fonts fontIds=1,2,5,6
   setIds=1,2,5,6 smartSetIds=1,2,5,6 workgroupId=2
```

getFontDataByIDs

Returns details about a font face given the font face id.

Syntax:

```
-getFontDataByIDs Font-IDs=list of IDs
```

PARAMETER	OPTIONAL?	DESCRIPTION
Font-IDs	No	A comma-separated list of font IDs. This command will request information about those fonts.

Suitcase Fusion Example:

```
corecli -standalone -getFontDataByIDs Font-IDs=40001,40015,40016
```

Universal Type Client Example:

```
corecli -clientserver -getFontDataByIDs Font-IDs=40001,40015,40016
```

moveFonts

Moves fonts between workgroups and static sets.

Syntax:

```
-moveFonts fontids=numbers srccontainerids=numbers srctype=string
   destcontainerid=number desttype=string
```

PARAMETER	OPTIONAL?	DESCRIPTION
fontids	No	A comma separated list of font IDs
srccontainerids	No	A comma separated list of either set IDs or workgroup IDs
srctype	No	Type of container specified: `workgroup` or `set`
destcontainerid	No	ID of the destination set or workgoup
desttype	No	Type of container specified: `workgroup` or `set`

Suitcase Fusion Example:

```
corecli -standalone -moveFonts fontids=1,10 srccontainerids=1,2 sourcetype=workgroup
   destcontainerid=3 desttype=set
```

Universal Type Client Example:

```
corecli -clientserver -moveFonts fontids=1,10 srccontainerids=1,2
   sourcetype=workgroup destcontainerid=3 desttype=set
```

queryForFontData

Accesses font data stored in the local database.

Syntax:

```
-queryForFontData [file=string][xml=string]
```

PARAMETER	OPTIONAL?	DESCRIPTION
file or xml	No	File: A fully-qualified path to a file containing the xml representing the query to execute. Xml: The xml representing the query to execute.

NOTE: See ***XML spec file format*** on page 48 for appropriate file structure.

Suitcase Fusion Mac Example:

```
corecli -standalone -queryForFontData file=/tmp/query.xml
```

Suitcase Fusion Windows Example:

```
corecli -standalone -queryForFontData file=C:\temp\query.xml
```

Universal Type Client Mac Example:

```
corecli -clientserver -queryForFontData file=/tmp/query.xml
```

Universal Type Client Windows Example:

```
corecli -clientserver -queryForFontData file=C:\temp\query.xml
```

Family Groups Commands

Family group names are extracted from fonts when they are added to the database. These family group names can be updated so that a font is grouped with other fonts. The family group commands can be used to query, add, rename or revert the family group setting for specified fonts.

The following family group commands are supported by the font management core (FMCore) for both Suitcase Fusion and the Universal Type Client:

assignFontsToFamilyGroup

Assigns fonts to a family group.

Syntax:

```
-assignFontsToFamilyGroup fontIDs=numbers familygroupID=number workgroupID=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
fontIDs	No	A comma separated list of font ids to assign to the family group
familygroupID	No	The id of the family group to assign fonts
workgroupID	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -assignFontsToFamilyGroup fontIDs=1,2,3 familygroupID=1
    workgroupID=1
```

Universal Type Client Example:

```
corecli -clientserver -assignFontsToFamilyGroup fontIDs=1,2,3 familygroupID=1
    workgroupID=1
```

getFamilyGroupDataByIDs

Returns details about a family group given for a family group id.

Syntax:

```
-getFamilyGroupDataByIDs familyGroupIDs=list of IDs
```

PARAMETER	OPTIONAL?	DESCRIPTION
familyGroupIDs	No	A comma-separated list of family group IDs

Suitcase Fusion Example:

```
corecli -standalone -getFamilyGroupDataByIDs familyGroupIDs =40001,40015,40016
```

Universal Type Client Example:

```
corecli -clientserver -getFamilyGroupDataByIDs familyGroupIDs =40001,40015,40016
```

queryForFamilyGroupData

This queries for family group data using a family group query which is very similar to the font query that one would use for `QueryForFontData`.

Syntax:

```
-queryForFamilyGroupData [file=string] [xml=string]
```

PARAMETER	OPTIONAL?	DESCRIPTION
file or xml	No	File: A fully-qualified path to a file containing the xml representing the query to execute. Xml: The xml representing the query to execute.

NOTE: See ***XML spec file format*** on page 48 for appropriate file structure.

Suitcase Fusion Example:

```
corecli -standalone -queryForFamilyGroupData file=/tmp/query.xml
```

Universal Type Client Example:

```
corecli -clientserver -queryForFamilyGroupData file=/tmp/query.xml
```

renameFamilyGroup

Renames a family group.

Syntax:

```
-renameFamilyGroup familyGroupId=number name=string workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
familyGroupId	No	The id of the family group to rename
name	No	The name to assign to the family group
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -renameFamilyGroup familyGroupId=678 name="Tastes Like Chicken"
    workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -renameFamilyGroup familyGroupId=678 name="Tastes Like Chicken"
    workgroupId=1
```

restoreFamilyGroupName

Restores the family group names back to the original scanned family group names.

Syntax:

```
-restoreFamilyGroupName familyGroupIds=numbers workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
familyGroupIds	No	A comma separated list of family group ids to restore the original scanned names
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -restoreFamilyGroupName familyGroupIds=1,2,3 workgroupID=1
```

Universal Type Client Example:

```
corecli -clientserver -restoreFamilyGroupName familyGroupIds=1,2,3 workgroupID=1
```

revertToScannedFamilyGroup

Reverts the fonts back to the original scanned family group.

Syntax:

```
-revertToScannedFamilyGroup fontIds=numbers workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
fontIds	No	A comma separated list of font ids to revert the family group to the original scanned group
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -revertToScannedFamilyGroup fontIds=1,2,3 workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -revertToScannedFamilyGroup fontIds=1,2,3 workgroupId=1
```

Keyword Commands

A global list of keywords is maintained by Suitcase Fusion as well as the Universal Type Client. The keyword commands can be used to edit the global list, as well as apply keywords to specified fonts.

The following keyword commands are supported by the font management core (FMCore) for both Suitcase Fusion and the Universal Type Client:

addKeyword

Adds a new keyword to the global list.

Syntax:

```
-addKeyword value=string workgroupID=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
value	No	The value of the keyword to add
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -addKeyword value=Foo workgroupID=1
```

Universal Type Client Example:

```
corecli -clientserver -addKeyword value=Foo workgroupID=1
```

addKeywordsToFonts

Adds keywords to fonts.

Syntax:

```
-addKeywordsToFonts workgroupID=number fontIDs=number keywordIDs=numbers
```

PARAMETER	OPTIONAL?	DESCRIPTION
workgroupId	No	The id of the target workgroup
fontIDs	No	A comma separated list of font ids to add keywords
keywordIDs	No	A comma separated list of keyword ids to add to selected fonts

Suitcase Fusion Example:

```
corecli -standalone -addKeywordsToFonts fontids=1,3,5,6,8,14 keywordids=1,2
    workgroupID=1
```

Universal Type Client Example:

```
corecli -clientserver -addKeywordsToFonts fontids=1,3,5,6,8,14 keywordids=1,2
    workgroupID=1
```

deleteKeywords

Deletes keywords from the global list.

Syntax:

```
-deleteKeywords keywordIds=numbers workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
keywordIds	No	A comma separated list of keyword ids to delete
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -deleteKeywords ids=1,2,3 workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -deleteKeywords ids=1,2,3 workgroupId=1
```

editKeyword

Edit an existing keyword in the global list.

Syntax:

```
-editKeyword keywordId=number value=string workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
keywordId	No	The id of the keyword to modify
value	No	The new value for the keyword
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -editKeyword keywordId=123 value=Foo workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -editKeyword keywordId=123 value=Foo workgroupId=1
```

removeKeywordsFromFonts

Remove keywords from one or more fonts.

Syntax:

```
-removeKeywordsFromFonts fontIDs=numbers keywordIDs=numbers workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
fontIDs	No	A comma separated list of font ids from to remove keywords
keywordId	No	A comma separated list of keyword ids to remove from fonts
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -removeKeywordsFromFonts fontids=1,3,5,6,8,14 keywordids=1,2,3
   workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -removeKeywordsFromFonts fontids=1,3,5,6,8,14 keywordids=1,2,3
   workgroupId=1
```

Foundry Commands

A font's foundry is extracted when the font is added to Suitcase Fusion or the Universal Type Client. A global list of foundries is also maintained in the database. Use the foundry commands to add or remove items from the global list, as well as update the font foundry applied to specified fonts.

The following foundry commands are supported by the font management core (FMCore) for both Suitcase Fusion and the Universal Type Client:

addCustomFoundry

Adds a new custom foundry to the global list.

Syntax:

```
-addCustomFoundry value=string workgroupID=number\
```

PARAMETER	OPTIONAL?	DESCRIPTION
value	No	The value of the foundry to add
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -addCustomFoundry value=Foo workgroupID=1
```

Universal Type Client Example:

```
corecli -clientserver -addCustomFoundry value=Foo workgroupID=1
```

assignFoundryToFonts

Assign a foundry to fonts.

Syntax:

```
-assignFoundryToFonts fontIDs=numbers foundryID=number workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
workgroupId	No	The id of the target workgroup
fontIDs	No	A comma separated list of font ids to assign a foundry
foundryID	No	The id of the foundry to assign

Suitcase Fusion Example:

```
corecli -standalone -assignFoundryToFonts fontIDs=123 foundryID=1 workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -assignFoundryToFonts fontIDs=123 foundryID=1 workgroupId=1
```

deleteCustomFoundries

Deletes a custom foundry from the global list.

Syntax:

```
-deleteCustomFoundries foundryIds=numbers workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
foundryIds	No	A comma separated list of foundry ids to delete
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -deleteCustomFoundries foundryIds=1,2,3 workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -deleteCustomFoundries foundryIds=1,2,3 workgroupId=1
```

editCustomFoundry

Edit an existing foundry in the global list.

Syntax:

```
-editCustomFoundry foundryId=number value=string workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
foundryId	No	The id of the foundry to modify
value	No	The new value for the foundry
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -editCustomFoundry foundryId=123 value=Foo workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -editCustomFoundry foundryId=123 value=Foo workgroupId=1
```

revertToScannedFoundries

Reverts the foundry value assigned to fonts back to the original foundry that was scanned when the fonts were originally added to the system.

Syntax:

```
-revertToScannedFoundries fontIds=numbers workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
fontIds	No	A comma separated list of font ids to revert the foundry
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -revertToScannedFoundries fontIds=1,2,3 workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -revertToScannedFoundries fontIds=1,2,3 workgroupId=1
```

Classification Commands

A global list of classifications is stored by Universal Type Server or Suitcase Fusion. Use the classification commands to add or remove items from the global list of classifications, or to apply, remove or modify classifications from fonts.

The following classification commands are supported by the font management core (FMCore) for both Suitcase Fusion and the Universal Type Client:

addCustomClassification

Adds a new custom classification to the global list.

Syntax:

```
-addCustomClassification value=string workgroupID=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
value	No	The value of the classification to add
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -addCustomClassification value=Foo workgroupID=1
```

Universal Type Client Example:

```
corecli -clientserver -addCustomClassification value=Foo workgroupID=1
```

assignClassificationToFonts

Assign a classification to fonts.

Syntax:

```
-assignClassificationToFonts fontIDs=numbers classificationID=number
    workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
workgroupId	No	The id of the target workgroup.
fontIDs	No	A comma separated list of font ids to assign the new classification.
classificationID	No	The id of the classification to assign to the font.

Suitcase Fusion Example:

```
corecli -standalone -assignClassificationToFonts fontIDs=123 classificationID=1
    workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -assignClassificationToFonts fontIDs=123 classificationID=1
    workgroupId=1
```

deleteCustomClassifications

Deletes custom classifications from the global list.

Syntax:

```
-deleteCustomClassifications classificationIds=numbers workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
classificationIds	No	A comma separated list of classification ids to delete
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -deleteCustomClassifications classificationIds=1,2,3
    workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -deleteCustomClassifications classificationIds=1,2,3
    workgroupId=1
```

editCustomClassification

Edit an existing classification in the global list.

Syntax:

```
-editCustomClassification classificationId=number value=string workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
classificationId	No	The id of the classification to modify
value	No	The new value for the classification
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -editCustomClassification classificationId=123 value=Foo
    workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -editCustomClassification classificationId=123 value=Foo
    workgroupId=1
```

revertToScannedClassifications

Reverts the classification value assigned to fonts back to the original classification that was scanned when the fonts were originally added to the system.

Syntax:

```
-revertToScannedClassifications fontIds=numbers workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
fontIds	No	A comma separated list of font ids to revert the classification.
workgroupId	No	The id of the target workgroup.

Suitcase Fusion Example:

```
corecli -standalone -revertToScannedClassifications fontIds=1,2,3 workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -revertToScannedClassifications fontIds=1,2,3 workgroupId=1
```

Style Commands

A global list of styles is maintained by Suitcase Fusion and the Universal Type Client. Use the style commands to edit the global list, as well as apply styles to selected fonts.

The following style commands are supported by the font management core (FMCore) for both Suitcase Fusion and the Universal Type Client:

addCustomStyle

Adds a new custom style to the global list.

Syntax:

```
-addCustomStyle value=string workgroupID=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
value	No	The value of the style to add
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -addCustomStyle value=Foo workgroupID=1
```

Universal Type Client Example:

```
corecli -clientserver -addCustomStyle value=Foo workgroupID=1
```

addStylesToFonts

Adds styles to fonts.

Syntax:

```
-addStylesToFonts fontIDs=numbers styleIDs=numbers workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
workgroupId	No	The id of the target workgroup
fontIDs	No	A comma separated list of the font ids to add styles
styleIDs	No	A comma separated list of the style ids to add to the fonts

Suitcase Fusion Example:

```
corecli -standalone -addStylesToFonts fontIDs=1,2,4,5 styleIDs=1 workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -addStylesToFonts fontIDs=1,2,4,5 styleIDs=1 workgroupId=1
```

deleteCustomStyles

Deletes custom style from the global list.

Syntax:

```
-deleteCustomStyles styleIds=numbers workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
workgroupId	No	The id of the target workgroup
styleIds	No	A comma separated list of the style ids to delete

Suitcase Fusion Example:

```
corecli -standalone -deleteCustomStyles styleIds=1 workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -deleteCustomStyles styleIds=1 workgroupId=1
```

editCustomStyle

Edit an existing style in the global list.

Syntax:

```
-editCustomStyle styleId=number value=string workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
styleId	No	The id of the style to modify
value	No	The new value for the style
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -editCustomStyle styleId=123 value=Foo workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -editCustomStyle styleId=123 value=Foo workgroupId=1
```

removeStylesFromFonts

Remove styles from one or more fonts.

Syntax:

```
-removeStylesFromFonts fontIDs=numbers styleIDs=numbers workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
fontIDs	No	A comma separated list of the font ids to remove styles
styleIDs	No	A comma separated list of the style ids to remove
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -removeStylesFromFonts fontids=1,2,4,5 styleIDs=1 workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -removeStylesFromFonts fontids=1,2,4,5 styleIDs=1 workgroupId=1
```

revertToScannedStyles

Reverts the classification value assigned to fonts back to the original styles that was scanned when the fonts were originally added to the system.

Syntax:

```
-revertToScannedStyles fontIds=numbers workgroupId=number
```

PARAMETER	OPTIONAL?	DESCRIPTION
fontIds	No	A comma separated list of the font ids to revert styles
workgroupId	No	The id of the target workgroup

Suitcase Fusion Example:

```
corecli -standalone -revertToScannedStyles fontIds=123 workgroupId=1
```

Universal Type Client Example:

```
corecli -clientserver -revertToScannedStyles fontIds=123 workgroupId=1
```

Server Commands

Server commands are useful to check or set the status of a current user, check for active servers, and replicate server data. These commands are only available for use with Universal Type Server, on computers running Universal Type Client.

The following server commands are available:

changePassword

Changes a logged in user's password. This command only functions if the user has permission to change their password.

NOTE: This command is available for use only with the Universal Type Client.

Syntax:

```
-changePassword password=string
```

PARAMETER	OPTIONAL?	DESCRIPTION
password	No	The new password

Universal Type Client Example:

```
corecli -clientserver -changePassword password=IThinkImFunnyButImNot
```

clearCachedServers

Clears the list of cached servers that are returned by the `getCachedServers` command.

NOTE: This command is available for use only with the Universal Type Client.

Syntax:

```
-clearCachedServers
```

Universal Type Client Example:

```
corecli -clientserver -clearCachedServers
```

getCachedServers

Lists all of the servers cached in the local database.

NOTE: This command is available for use only with the Universal Type Client.

Syntax:

```
-getCachedServers
```

Universal Type Client Example:

```
corecli -clientserver -getCachedServers
```

getSession

Lists the server session that is being tracked in the local database.

NOTE: This command is available for use only with the Universal Type Client.

Syntax:

```
-getSession
```

Universal Type Client Example:

```
corecli -clientserver -getSession
```

getUmaURL

Returns the URL of the Users and Workgroups Administration application for the currently logged in server.

NOTE: This command is available for use only with the Universal Type Client.

Syntax:

```
-getUmaURL
```

Universal Type Client Example:

```
corecli -clientserver -getUmaURL
```

goOffline

Sets the user's session to offline mode.

NOTE: This command is available for use only with the Universal Type Client.

Syntax:

```
-goOffline
```

Universal Type Client Example:

```
corecli -clientserver -goOffline
```

goOnline

Sets the user's session to online mode.

NOTE: This command is available for use only with the Universal Type Client.

Syntax:

```
-goOnline
```

Universal Type Client Example:

```
corecli -clientserver -goOnline
```

isLoggedIn

Returns whether a user is logged in or not.

NOTE: This command is available for use only with the Universal Type Client.

Syntax:

```
-isLoggedIn
```

Universal Type Client Example:

```
corecli -clientserver -isLoggedIn
```

isOnline

Returns whether a session is online.

NOTE: This command is available for use only with the Universal Type Client.

Syntax:

```
-isOnline
```

Universal Type Client Example:

```
corecli -clientserver -isOnline
```

login

Logs in to a server.

NOTE: This command is available for use only with the Universal Type Client.

Syntax:

```
-login [(bonjourName=string server=string port=number username=string
    password=string)]
```

PARAMETER	OPTIONAL?	DESCRIPTION
bonjourName	Yes	The Bonjour name of the Universal Type Server. If this is supplied, server and port parameters will be ignored and Bonjour will be used to look up that information.
server	Yes	Must be supplied with port, username, and password. This is the name or IP address of the server.
port	Yes	Must be supplied with server, username, and password. This is the port number on the server.
username	Yes	Must be supplied with port, server, and password. This is the username of the user to be logged in.
password	Yes	Must be supplied with port, server, and username. This is the password of the user to be logged in.

Universal Type Client Examples:

```
corecli -clientserver -login server=mystique.server.extensis.com port=8080
    username=jsmith password=IThinkImFunnyButImNot
corecli -clientserver -login server=10.2.1.23 port=80 username=smalone
    password=Cheers
```

replicateServerData

Initiates data replication from the server database to the local database.

NOTE: This command is available for use only with the Universal Type Client.

Syntax:

```
-replicateServerData mode=[replication mode]
```

PARAMETER	OPTIONAL?	DESCRIPTION
mode	Yes	The replication mode to use, one of: bulk, incremental, best (default)

Universal Type Client Example:

```
corecli -clientserver -replicateServerData mode=bulk
```

Permission Commands

Permission commands allow you to query the global and workgroup permissions that are set for the current Universal Type Client user.

The following permission commands are supported by the font management core (FMCore) for the Universal Type Client only:

- ***queryForUserGlobalPermissions*** (page 44)
- ***queryForUserWorkgroupPermissions*** (page 44)

queryForUserGlobalPermissions

Queries the local database for the global permissions assigned to the currently logged in user.

NOTE: This command is available for use only with the Universal Type Client.

Syntax:

```
-queryForUserGlobalPermissions [file=string]
```

PARAMETER	OPTIONAL?	DESCRIPTION
file	Yes	The path to a file containing an XML result spec.

NOTE: See ***XML spec file format*** on page 48 for appropriate file structure.

Universal Type Client Example:

```
corecli -clientserver -queryForUserGlobalPermissions file=../SomeFileName.xml
```

queryForUserWorkgroupPermissions

Queries the local database for the workgroup permissions assigned to the currently logged in user.

NOTE: This command is available for use only with the Universal Type Client.

Syntax:

```
-queryForUserWorkgroupPermissions workgroupId=number [file=string]
```

PARAMETER	OPTIONAL?	DESCRIPTION
workgroupId	No	The workgroup to query permissions
file	Yes	The path to a file containing an XML result spec.

NOTE: See ***XML spec file format*** on page 48 for appropriate file structure.

Universal Type Client Example:

```
corecli -clientserver -queryForUserWorkgroupPermissions workgroupId=23
   file=../SomeFileName.xml
```

Database Commands

Database commands allow you to query the database schema type or perform specific SQL statements on the font management core database.

The following database commands are supported by the font management core (FMCore) for both Suitcase Fusion and the Universal Type Client:

getSchemaVersion

Displays the current schema version for the local database of the font management core.

Syntax:

```
-getSchemaVersion
```

Suitcase Fusion Example:

```
corecli -standalone -getSchemaVersion
```

Universal Type Client Example:

```
corecli -clientserver -getSchemaVersion
```

sql

Makes a direct connection to the SQLite database used by Suitcase Fusion and Universal Type Server and performs the SQL statement.

Syntax:

```
-sql db=string query=true-or-false [file=string] [sql=string]
```

PARAMETER	OPTIONAL?	DESCRIPTION
db	No	Path to a SQLite database file
query	No	Boolean value indicating whether the sql in the sql argument or in the file contains a query. For anything other than a SELECT statement, this should be false
file	Yes	Path to a file containing a SQLite-compatible SQL query, insert, update or delete statement
sql	Yes	The raw sql to execute against the database

Suitcase Fusion Example:

```
corecli -standalone -sql db=SuitcaseFusion.db query=false file=input1.txt
```

Universal Type Client Example:

```
corecli -clientserver -sql db=UniversalType.db query=false file=input1.txt
```

General Commands

General commands include a number of basic commands to query and interface with the font management core.

The following commands are supported by the font management core (FMCore) for both Suitcase Fusion and the Universal Type Client:

about

Displays a message describing the CoreCLI application.

Syntax:

```
-about
```

Suitcase Fusion Example:

```
corecli -standalone -about
```

Universal Type Client Example:

```
corecli -clientserver -about
```

handshake

Checks whether the font management Core is running.

Syntax:

```
-handshake
```

Suitcase Fusion Example:

```
corecli -standalone -handshake
```

Universal Type Client Example:

```
corecli -clientserver -handshake
```

setLoggingLevel

Sets logging levels to control amount of output from the FMCore.

Syntax:

```
-setLoggingLevel level=string [logid=string]
```

PARAMETER	OPTIONAL?	DESCRIPTION
level	No	The name of the logging level being applied . It must (case-insensitively) match: `error`, `warn`, `info`, debug, none.
logid	Yes	The logging identifier string to apply the logging level to. If not provided, it applies the logging level to the default logging identifier of `default`.

Suitcase Fusion Example:

```
corecli -standalone -setLoggingLevel level=warn logid=metadatacache
```

Universal Type Client Example:

```
corecli -clientserver -setLoggingLevel level=warn logid=metadatacache
```

shutdown

Requests that the font management core perform a coordinated shutdown.

Syntax:

```
-shutdown
```

Suitcase Fusion Example:

```
corecli -standalone -shutdown
```

Universal Type Client Example:

```
corecli -clientserver -shutdown
```

top

Queries the font management core for the list of tasks that are scheduled and/or actively running by the core.

Syntax:

```
-top
```

Suitcase Fusion Example:

```
corecli -standalone -top
```

Universal Type Client Example:

```
corecli -clientserver -top
```

XML spec file format

Several CoreCLI commands require the input of an XML spec file to dictate the type of information is returned. The commands include very powerful queries that can reveal a great deal of information, including:

- ***queryForFontData*** (page 25)
- ***queryForFamilyGroupData*** (page 27)
- ***queryForUserGlobalPermissions*** (page 44)
- ***queryForUserWorkgroupPermissions*** (page 44)

The following two examples indicate the format required to appropriately configure an XML file.

Example XML file #1

The following example queries for the font ids and postscript names of fonts with a foundry name that contains the text **Adobe**.

```
<font-query>
   <paging-options offset="0" size="0" />
   <column-specs>
         <column-spec id="FontFace__id" sort-option="NoSorting" />
         <column-spec id="FontFace__postscript_name" sort-option="NoSorting" />
   </column-specs>
   <workgroup-ids />
   <set-ids />
   <quick-find value="Adobe">
         <filter>foundry</filter>
   </quick-find>
</font-query>
```

Example XML File #2

The following query example searches for font ids and Postscript names in the set with `id=1`.

```
<font-query>
   <paging-options offset="0" size="0" />
   <column-specs>
         <column-spec id="FontFace__id" sort-option="NoSorting" />
         <column-spec id="FontFace__postscript_name" sort-option="NoSorting" />
   </column-specs>
   <workgroup-ids />
   <set-ids>
         <id>1</id>
   </set-ids>
   <quick-find value="" />
</font-query>
```

`column-spec` xml element

The `column-spec` element is the id attribute that identifies the column to select from the database. The `id` attribute value contains two parts that must be separated by two underscore characters. The

first part of the `column-spec` element is the name of the table, and the second is the name of the column.

For example, to select the `FontFace.type` database column in a query use `FontFace__type` as the id value.

In addition, the `column-spec` element supports three sort-option values:

- `NoSorting`
- `Ascending`
- `Descending`

`quick-find` value filters

The `quick-find value` element allows you to filter the results of your query by including designated filter fields. In the following example, all of the supported fields are included, and searched for the value **Adobe**.

```
<quick-find value="Adobe">
    <filter>foundry</filter>
    <filter>family</filter>
    <filter>postscriptname</filter>
    <filter>classification</filter>
    <filter>fullname</filter>
    <filter>displayname</filter>
</quick-find>
```

Contacting Extensis

Extensis

1800 SW First Avenue, Suite 500
Portland, OR 97201
Toll Free: (800) 796-9798
Phone: (503) 274-2020
Fax: (503) 274-0530
Web: *http://www.extensis.com*

Extensis Europe

Suite 18, Newton House
Kings Park Road, Moulton Park
Northampton NN3 6LG, United Kingdom
Phone: +44 (0)1604 654 270
Fax: +44 (0)1604 654 268
Email: *info@extensis.co.uk*

Customer Service

Phone: (800) 796-9798
Email: *info@extensis.com*
Web: *http://www.extensis.com/customerservice/*

Celartem, Inc.

Phone: +81 3 5574 7236
Email: *sales_ap@celartem.com*
Web: *http://www.celartem.com/en/*

Corporate Buying

Phone: (503) 274-4492
Email: *NA_sales@extensis.com* (North America)
Web: *http://www.extensis.com/store/corporate-buying/* (all regions)

Technical Support

Technical support for current products is available by phone or through the Extensis website.

- **North America:** (800) 796-9798 , select option 3 (8:00 am–4:00 pm Pacific time, Monday–Friday)
- **US:** (503) 274-7030
- **Europe:** +44 (0)1604 654 270
- **Web:** *http://www.extensis.com/support/*
- **Web Support Form:** *https://secure.extensis.com/technical-support/* (requires login)

Answers to frequently asked questions, troubleshooting tips, and more can also be found at the *Extensis support page*.

Extensis also maintains a searchable *Knowledge Base* (*http://support.extensis.com/Support/58278/58411/en-us/Portal/Index*) of in-depth articles on various technical topics.

Creating a tech support case

If you are experiencing a problem with a current product, you can submit a tech support case using the *Extensis Web Support Form*.

To access the form, you must log in using your Extensis account. Once you have logged in, provide as much of the following information as you can:

- Product name and version number;
- Serial number, if you have it available;
- Computer operating system version;
- Other details about your computer system, including RAM, hard drive size and free space, and processor type and speed;
- A description of the problem, including any error message that might be displayed;
- Your phone number if you want to have a representative contact you.

Support Policy

Extensis provides full support for the current version of all shipping products. In addition, Extensis provides limited support for older products up to one year after the product version is no longer offered for sale. For complete details see the *Extensis Product Support Policy*.

For details on currently supported products, see:

- *Portfolio Support Guide*
- *Universal Type Server Support Guide*
- *Universal Type Client Support Guide*
- *Suitcase Fusion Support Guide*

Priority Support

If you have a current Annual Service Agreement, you are entitled to priority support.

If you are in North or South America or the Caribbean:

- **Email:** *p1support@extensis.com*

If you are in Europe, Africa, the Middle East, India, Australasia, or Asia (except Japan):

- **Email:** *EuroASASupport@extensis.com*

If you are evaluating a demo copy of the product, please contact your sales representative for assistance.

Community Support

Extensis maintains community forums on all current and many older products. Often, problems you may be experiencing have been discovered and answered here. In addition, suggestions you have may help others resolve issues.

The forums also serve as a way for Extensis to take the pulse of our user community so that we can identify bugs and other issues and gather suggestions for improving our software.

Please visit the *Extensis Forums* and bookmark the page.

Index

G

H

I

K

L

M

P

Q

R

S

T

W