

NSUserDefaults

Inherits From:	NSObject
Conforms To:	NSCoding NSObject
Declared In:	foundation/NSUserDefaults.h

Getting the Shared Instance

+ (NSUserDefaults *) standardUserDefaults	Returns the shared defaults object. If it doesn't exist yet, it's created with a search list containing the names of the following domains, in order: the NSArgumentDomain (consisting of defaults parsed from the application's arguments), a domain with the process' name, separate domains for each of the user's preferred languages, the NSGlobalDomain (consisting of defaults meant to be seen by all applications), and the NSRegistrationDomain (a set of temporary defaults whose values can be set by the application to ensure that searches will always be successful). The defaults are initialized for the current user.
--	--

Getting and Setting a Default

- (id) objectForKey: (NSString *) <i>defaultName</i>	Returns the value of the first occurrence of the specified default, searching the domains included in the search list. Returns nil if the default isn't found.
- (void) setObject: (id) <i>value</i> forKey: (NSString *) <i>defaultName</i>	Sets the value of the specified default in the first domain in the search list, as well as in the first persistent domain (if the first domain isn't persistent). To save this new value to the disk, invoke savePersistentDomains .

Initializing the User Defaults

- (id)**init** Initializes defaults for the current user (who's identified by examining the environment). This method doesn't put anything in the search list. Invoke it only if you've allocated your own `NSUserDefaults` object instead of using the shared one. Returns **self**.
- (id)**initWithUser:(NSString *)userName** Like **init**, but initializes defaults for the specified user.

Specifying Which Domains to Search

- (NSMutableArray *)**searchList** Returns a mutable array of domain names, signifying the domains that **objectForKey:** will search.

Maintaining Persistent Domains

- (NSDictionary *)**persistentDomainForName:(NSString *)domainName** Returns a dictionary corresponding to the specified persistent domain. The keys in the dictionary are names of defaults, and the value corresponding to each key is a property-list data object.
- (NSArray *)**persistentDomainNames** Returns an array containing the names of the persistent domains. Each domain can then be retrieved by calling **persistentDomainForName:.**
- (void)**removePersistentDomainForName:(NSString *)domainName** Removes the named domain from the list of persistent domains.
- (void)**savePersistentDomains** Writes the persistent domains to the disk. Modifications aren't permanent until this method (or **writeObject:forKey:inDomain:.**) is called.
- (void)**setPersistentDomain:(NSDictionary *)domain
forName:(NSString *)domainName** Sets the persistent domain named *domainName* to *domain*, replacing any existing persistent domain of that name.
- (void)**updatePersistentDomains** Discards the current user's changes to the persistent domains, reverting to the state stored on disk. (The new state can include changes saved by other users in the interim.)

Maintaining Volatile Domains

- (void)**removeVolatileDomainForName:(NSString *)domainName**

- (void)**setVolatileDomain:**(NSDictionary *)*domain*
forName:(NSString *)*domainName* Removes the named domain from the list of volatile domains. Sets the volatile domain named *domainName* to *domain*, replacing any existing volatile domain of that name.
- (NSDictionary *)**volatileDomainForName:**(NSString *)*domainName* Returns a dictionary corresponding to the specified volatile domain. The keys in the dictionary are names of defaults, and the value corresponding to each key is a property-list data object.
- (NSArray *)**volatileDomainNames** Returns an array containing the names of the volatile domains. Each domain can then be retrieved by calling **volatileDomainForName:**.

Making Advanced Use of Defaults

- (NSDictionary *)**dictionaryRepresentation** Returns a dictionary containing entries (key/value pairs) for each object in the search list. The combined result doesn't preserve the information about which domain each entry came from.
- (void)**registerDefaults:**(NSDictionary *)*dictionary* Adds the contents of *dictionary* to the registration domain. If there is no registration domain yet, it's created using *dictionary*, and NSRegistrationDomain is added to the end of the search list.
- (id)**updatedObjectForKey:**(NSString *)*defaultName* Updates the value of the specified default by reading from the disk. The default is updated wherever it's present among the persistent domains in the search list. This method returns the value of the first occurrence of *defaultName* in any domain (persistent or volatile) in the search list.
- (void)**writeObject:**(id)*value*
forKey:(NSString *)*defaultName*
inDomain:(NSString *)*domainName* Sets the value of the specified default to *value* and writes it to disk, without saving any changes made to the other defaults.