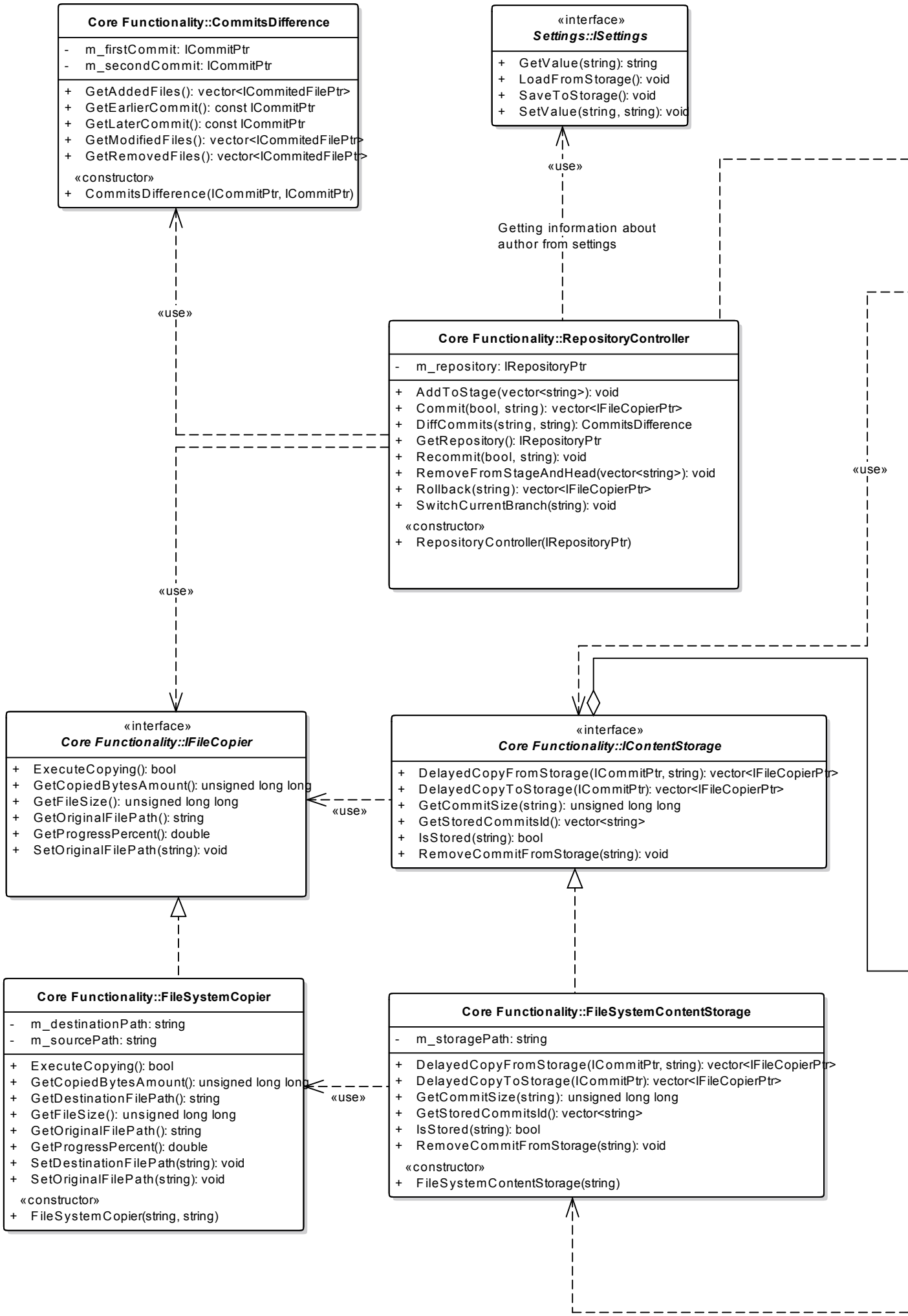
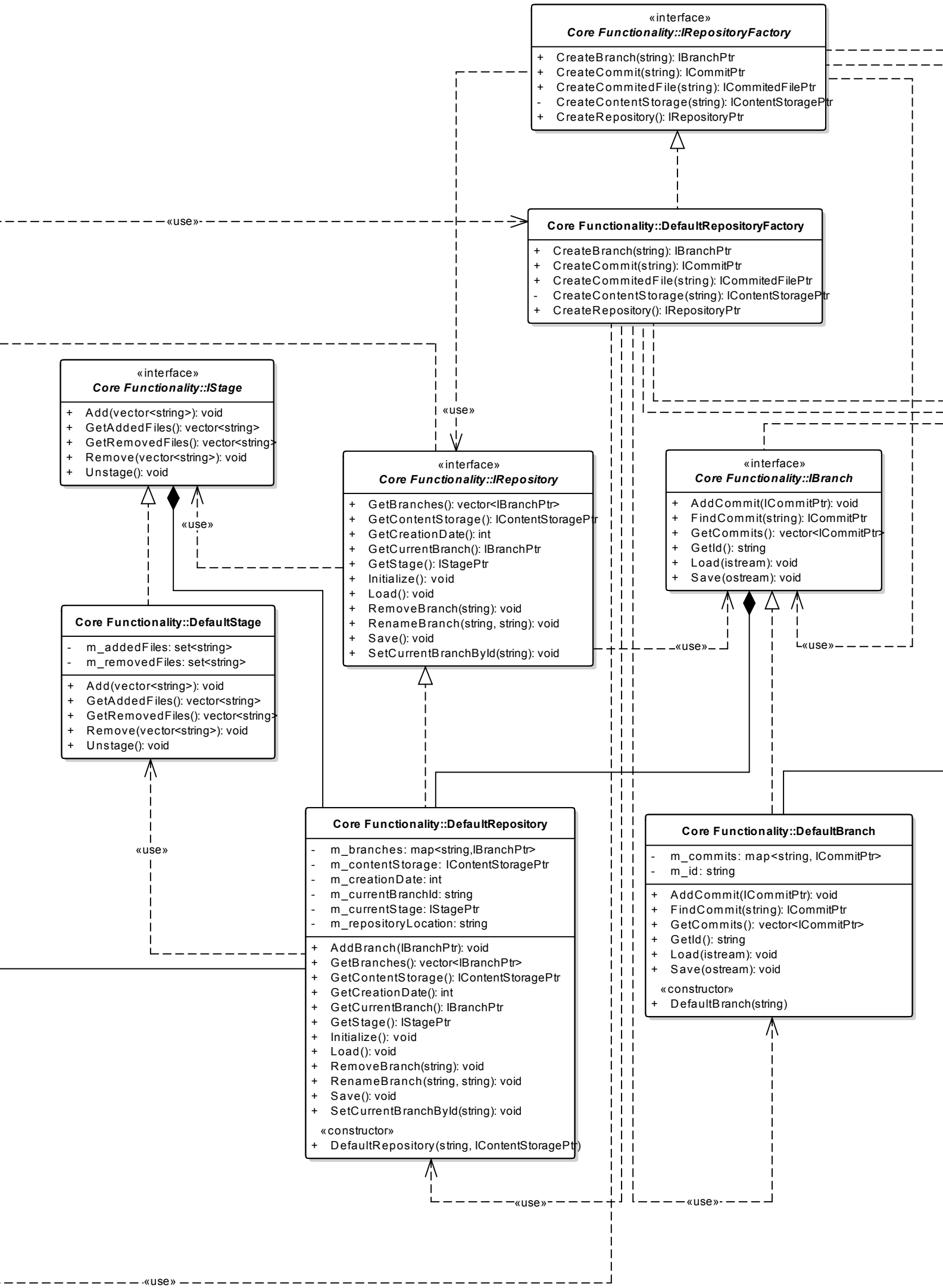
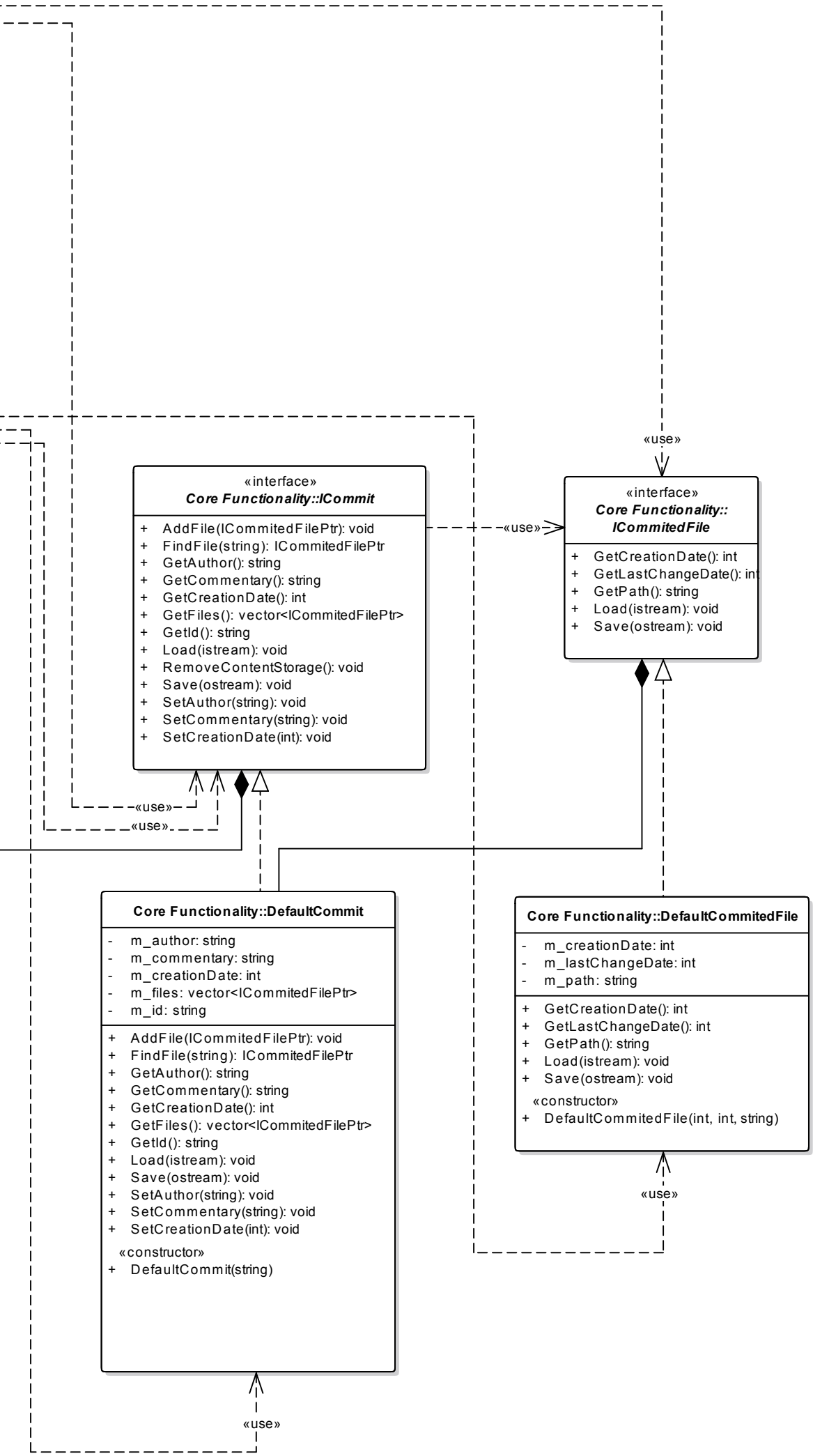

Core Functionality::CommitsDifference
- m_firstCommit: ICommitPtr
- m_secondCommit: ICommitPtr
+ GetAddedFiles(): vector<ICommitedFilePtr>
+ GetEarlierCommit(): const ICommitPtr
+ GetLaterCommit(): const ICommitPtr
+ GetModifiedFiles(): vector<ICommitedFilePtr>
+ GetRemovedFiles(): vector<ICommitedFilePtr>
«constructor»
+ CommitsDifference(ICommitPtr, ICommitPtr)
«interface»
Settings::ISettings
+ GetValue(string): string
+ LoadFromStorage(): void
+ SaveToStorage(): void
+ SetValue(string, string): void
«use»
Getting information about author from settings
«use»
Core Functionality::RepositoryController
- m_repository: IRepositoryPtr
+ AddToStage(vector<string>): void
+ Commit(bool, string): vector<IFileCopierPtr>
+ DiffCommits(string, string): CommitsDifference
+ GetRepository(): IRepositoryPtr
+ Recommit(bool, string): void
+ RemoveFromStageAndHead(vector<string>): void
+ Rollback(string): vector<IFileCopierPtr>
+ SwitchCurrentBranch(string): void
«constructor»
+ RepositoryController(IRepositoryPtr)
«use»
«use»
«interface»
Core Functionality::IFileCopier
+ ExecuteCopying(): bool
+ GetCopiedBytesAmount(): unsigned long long
+ GetFileSize(): unsigned long long
+ GetOriginalFilePath(): string
+ GetProgressPercent(): double
+ SetOriginalFilePath(string): void
«use»
«interface»
Core Functionality::IContentStorage
+ DelayedCopyFromStorage(ICommitPtr, string): vector<IFileCopierPtr>
+ DelayedCopyToStorage(ICommitPtr): vector<IFileCopierPtr>
+ GetCommitSize(string): unsigned long long
+ GetStoredCommitsId(): vector<string>
+ IsStored(string): bool
+ RemoveCommitFromStorage(string): void
Core Functionality::FileSystemCopier
- m_destinationPath: string
- m_sourcePath: string
+ ExecuteCopying(): bool
+ GetCopiedBytesAmount(): unsigned long long
+ GetDestinationFilePath(): string
+ GetFileSize(): unsigned long long
+ GetOriginalFilePath(): string
+ GetProgressPercent(): double
+ SetDestinationFilePath(string): void
+ SetOriginalFilePath(string): void
«constructor»
+ FileSystemCopier(string, string)
«use»
Core Functionality::FileSystemContentStorage
- m_storagePath: string
+ DelayedCopyFromStorage(ICommitPtr, string): vector<IFileCopierPtr>
+ DelayedCopyToStorage(ICommitPtr): vector<IFileCopierPtr>
+ GetCommitSize(string): unsigned long long
+ GetStoredCommitsId(): vector<string>
+ IsStored(string): bool
+ RemoveCommitFromStorage(string): void
«constructor»
+ FileSystemContentStorage(string)


«interface»
Core Functionality::IRepositoryFactory
+ CreateBranch(string): IBranchPtr
+ CreateCommit(string): ICommitPtr
+ CreateCommitedFile(string): ICommitedFilePtr
- CreateContentStorage(string): IContentStoragePtr
+ CreateRepository(): IRepositoryPtr
«use»
Core Functionality::DefaultRepositoryFactory
+ CreateBranch(string): IBranchPtr
+ CreateCommit(string): ICommitPtr
+ CreateCommitedFile(string): ICommitedFilePtr
- CreateContentStorage(string): IContentStoragePtr
+ CreateRepository(): IRepositoryPtr
«interface»
Core Functionality::IStage
+ Add(vector<string>): void
+ GetAddedFiles(): vector<string>
+ GetRemovedFiles(): vector<string>
+ Remove(vector<string>): void
+ Unstage(): void
«use»
«interface»
Core Functionality::IRepository
+ GetBranches(): vector<IBranchPtr>
+ GetContentStorage(): IContentStoragePtr
+ GetCreationDate(): int
+ GetCurrentBranch(): IBranchPtr
+ GetStage(): IStagePtr
+ Initialize(): void
+ Load(): void
+ RemoveBranch(string): void
+ RenameBranch(string, string): void
+ Save(): void
+ SetCurrentBranchById(string): void
«interface»
Core Functionality::IBranch
+ AddCommit(ICommitPtr): void
+ FindCommit(string): ICommitPtr
+ GetCommits(): vector<ICommitPtr>
+ GetId(): string
+ Load(istream): void
+ Save(ostream): void
«use»
Core Functionality::DefaultStage
- m_addedFiles: set<string>
- m_removedFiles: set<string>
+ Add(vector<string>): void
+ GetAddedFiles(): vector<string>
+ GetRemovedFiles(): vector<string>
+ Remove(vector<string>): void
+ Unstage(): void
«use»
Core Functionality::DefaultRepository
- m_branches: map<string,IBranchPtr>
- m_contentStorage: IContentStoragePtr
- m_creationDate: int
- m_currentBranchId: string
- m_currentStage: IStagePtr
- m_repositoryLocation: string
+ AddBranch(IBranchPtr): void
+ GetBranches(): vector<IBranchPtr>
+ GetContentStorage(): IContentStoragePtr
+ GetCreationDate(): int
+ GetCurrentBranch(): IBranchPtr
+ GetStage(): IStagePtr
+ Initialize(): void
+ Load(): void
+ RemoveBranch(string): void
+ RenameBranch(string, string): void
+ Save(): void
+ SetCurrentBranchById(string): void
«constructor»
+ DefaultRepository(string, IContentStoragePtr)
Core Functionality::DefaultBranch
- m_commits: map<string, ICommitPtr>
- m_id: string
+ AddCommit(ICommitPtr): void
+ FindCommit(string): ICommitPtr
+ GetCommits(): vector<ICommitPtr>
+ GetId(): string
+ Load(istream): void
+ Save(ostream): void
«constructor»
+ DefaultBranch(string)
«use»
«use»
«use»
«use»
«use»


«use»
«interface»
Core Functionality::ICommit
+ AddFile(ICommitedFilePtr): void
+ FindFile(string): ICommitedFilePtr
+ GetAuthor(): string
+ GetCommentary(): string
+ GetCreationDate(): int
+ GetFiles(): vector<ICommitedFilePtr>
+ GetId(): string
+ Load(istream): void
+ RemoveContentStorage(): void
+ Save(ostream): void
+ SetAuthor(string): void
+ SetCommentary(string): void
+ SetCreationDate(int): void
«use»
«interface»
Core Functionality::
ICommitedFile
+ GetCreationDate(): int
+ GetLastChangeDate(): int
+ GetPath(): string
+ Load(istream): void
+ Save(ostream): void
«use»
«use»
Core Functionality::DefaultCommit
- m_author: string
- m_commentary: string
- m_creationDate: int
- m_files: vector<ICommitedFilePtr>
- m_id: string
+ AddFile(ICommitedFilePtr): void
+ FindFile(string): ICommitedFilePtr
+ GetAuthor(): string
+ GetCommentary(): string
+ GetCreationDate(): int
+ GetFiles(): vector<ICommitedFilePtr>
+ GetId(): string
+ Load(istream): void
+ Save(ostream): void
+ SetAuthor(string): void
+ SetCommentary(string): void
+ SetCreationDate(int): void
«constructor»
+ DefaultCommit(string)
Core Functionality::DefaultCommitedFile
- m_creationDate: int
- m_lastChangeDate: int
- m_path: string
+ GetCreationDate(): int
+ GetLastChangeDate(): int
+ GetPath(): string
+ Load(istream): void
+ Save(ostream): void
«constructor»
+ DefaultCommitedFile(int, int, string)
«use»
«use»