SAS® Drug Development API 1.7.1.1
Release Notes
March 15, 2016

The SAS® Drug Development (SDD) API release 1.7.1.1 is available for use with SDD Version 4.4 and 4.5.

The table below describes changes to the SDD API in release 1.7.1.1:

Reference #	Title	Description
DE6653	Performance issues when deleting items from the recycle bin when the recycle bin already contains many items	The delete methods on the *RecycleBinService* have been updated to delete more efficiently.

The table below describes changes to the SDD API in release 1.7.1:

Reference #	Title	Description
DE4106	Depot objects should not be renamed	The rename method on the *RepositoryResourceService* will now throw a RepositoryResourceRenameException if a depot is renamed. A depot corresponds to the Files location directly underneath a organization, project, or analysis.
DE4108	Renaming in /Users location	Renaming the /Users/<userid> location will now throw an AccessDeniedException if trying to rename another user's location, and it will throw a RepositoryResourceRenameException if trying to rename the current user's location.
DE4118	Cannot delete items from recycle bin without read permission even with proper privilege	With the *RecycleBinService*, a user can now permanently delete items from the recycle bin that they do not have read access to as long as they have the Permanently Delete Items privilege.

The table below describes changes to the SDD API in release 1.7:

Reference #	Title	Description
US4352	Move a repository object	The new method *move* was added to the *RepositoryResourceService*.
US7060	Provide proxy server support	Added support for proxy server configurations. A new method on the *SessionFactory* class called *newSessionWithProxy(URL url, String userId, String password, ProxyInfo proxyInfo)* allows users to provide a *ProxyInfo* object when establishing a session. This *ProxyInfo* object provides details for various proxy server configurations so users that require a proxy server to connect to the SDD instance can use this method. Please see the API documentation for further details.
US7137	Retrieve electronic signature information	Added *SignatureService* to provide the ability to create and retrieve electronic signatures associated with repository files.
US7195	Move a workspace object	The new method *move* was added to the *WorkspaceResourceService*.
US7196	Provide type service (surface extended attribute information)	Added *TypeService* and *com.sas.hls.type* package to provide the ability of retrieving the type information about objects that can be stored in SAS Drug Development. This information includes extended attribute information for types that provide that capability.
US7198	Retrieve workspace sync information	Added new methods *getSynchronizationInfo* and *getSynchronizationInfos* on the *RepositoryResourceService* to retrieve synchronization information describing the state of files between the workspace and the repository.
US7199	SDD 4.4 compatibility	Minor implementation changes were required to support SAS Drug Development 4.4.
US7202	Provide more details when authentication fails	More specific exceptions were implemented to support various authentication failures. The new exceptions are: • *AccountDeactivatedException* – thrown if the account being accessed is inactive. • *AccountException* – superclass of all account related authentication exceptions • *AccountExpiredException* – thrown if the account being accessed has expired. • *AccountLockedException* – thrown if the account being accessed is locked. • *AccountNotFoundException* – thrown if the account being accessed does not exist. • *CredentialsException* – superclass of all credentials related authentication exceptions. • *CredentialsExpiredException* – thrown if the password supplied has expired and the user must change their password. • *CredentialsIncorrectException* – thrown if the password supplied is incorrect. • *UnknownAuthenticationException* – thrown if there was an

Reference #	Title	Description
		unexpected problem issuing the authentication request.
US7208	Update properties for repository object	Added support for updating properties of repository objects. A new method *updateResource(RepositoryResource repositoryResource)* was added to the *RepositoryResourceService* to provide this functionality.
US7229	Add update method to organization service	Added new method *updateOrganization* to the *OrganizationService*. The organization lead and any editable attributes such as *description* are allowed to be updated with this method.
US7230	Add application information service	Added *ApplicationInformationService* and *com.sas.hls.configuration* package to provide the ability of retrieving build information for various components. This is used to determine if the API client version is compatible with the server version. This information is also helpful in debugging issues.
DE2353	Copy file: *RepositoryResourceNotFoundException* thrown when path is invalid	If the path being copied to has a forward slash, the copy would actual work but re-retrieving the final destination object would generate a *RepositoryResourceNotFoundException* since the "fixed" path was not used to lookup the object. This has been fixed so the destination object is returned and no exception is thrown.
DE3229	Performance enhancement for *updateAcl* and *updateAclContainer*	In prior releases, a check was made to verify updates to an owner on an ACL were valid. This check made sure the new owner being set was a valid member for the context in which the object resided. This check was fixed on the SDD server so the check in the API is no longer needed. This check was expensive in certain cases (such as when there were many groups at the context), since this was removed these methods should perform better.
DE3573	*DownloadFile* throws misleading exception when local parent directory doesn't exist	Local parent directories are now verified and created if they do not exist. If they cannot be created for any reason then an *IllegalPathException* is thrown describing the problem.
DE3577	Deleting user with membership throws wrong exception	When *deleteUser* was called on the *UserService* and the user was still a member at any context a *SystemException* was thrown describing all memberships must be revoked before deleting the user. This was corrected so a *UserDeleteException* is now thrown with a similar message.
DE3605	*GroupService* needs *isMember*	Added new method *isMember* to the *GroupService*. This method prevents having to lookup a group to see if it exists.
DE3606	*GroupService* needs *getGroupDescriptorByName*	Added new method *getGroupDescriptorByName* to the *GroupService*. This method provides a more efficient group lookup since the group members are not populated on a descriptor.
DE3609	*RoleAssignmentService* – trying to add a user that is not a member of the context fails quietly	The method *updateRoleAssignment(RoleAssignment roleAssignment)* on the *RoleAssignmentService* now throws the new exception *InvalidRoleMemberException* when a user is assigned to a role and that user is not a member. Since this is a new checked exception added to the method this is considered a breaking change so consuming code may need to change.

Reference #	Title	Description
DE3631	Looking up user ID with special characters causes *SystemException*	When attempting to retrieve a user with an invalid user ID a *SystemException* would be thrown rather than a *UserNotFoundException*. The invalid user ID would need to contain one of the following illegal characters [<space>, !, ', (,), <, =, >]. The method *userExists(String userId)* on the user service now returns false in this case, and the methods *getUserByUserId(String userId)* and *getUserDescriptorByUserId(String userId)* now throw a *UserNotFoundException*.
DE3673	Get versions call on organization throws misleading exception	If the method *getVersions* on the *RepositoryResourceService* was called using a non-file path a *RepositoryResourceNotFoundException* was errantly thrown. If the object actually exists but is not a file then an *IllegalPathException* is now thrown.
	Add constants for global privileges	Added the class *GlobalPrivileges* to the package *com.sas.hls.security.privilege* which contains a list of constants representing all global privileges. Global privileges can be assigned directly to user accounts.
	Add constants for scoped privileges	Added the class *Privileges* to the package *com.sas.hls.security.privilege* which contains a list of constants representing all scoped privileges. Scoped privileges are privileges that are available for role assignments at contexts such as the Organization, Projects and Analyses.
	RoleDefinitionService – trying to add an invalid privilege fails quietly	The methods *createRole* and *updateRole* on the *RoleDefinitionService* now throw the new exception *InvalidRolePrivilegeException* when a privilege is added to a role and the privilege is not valid at the context. Since this is a new checked exception added to the method this is considered a breaking change so consuming code may need to change.
	The method *isMember* should not throw an exception	The method *isMember* on the *GroupService* no longer throws a *GroupNotFoundException*. If the group does not exist then this method returns false. Since this a checked exception being removed from an existing method this is considered a breaking change so consuming code may need to change.
	Remove *Cloneable* from DTOs	The marker interface *Cloneable* has been removed from all Data Transfer Object definitions. This designation was not used in any implementations so it has been removed. It was never intended that consumers clone DTOs returned from the API service calls so this change is not considered a breaking change.
	Remove internal and duplicate attributes from *RepositoryResource*	In prior releases certain attributes were surfaced on the *RepositoryResource getAttributes* method call that were never intended to be surfaced or are duplicates for properties available directly on the resource. The affected attributes are: • *Context* – removed. This attribute was only available on certain object types; this value is used internally and should not be surfaced. Contexts for resources can be derived by their path. • *Owner* – removed. This attribute is used internally and should not be surfaced. The owner of an object is available on the ACL and should be retrieved from there.

Reference #	Title	Description
		• *Revision* – removed. This attribute is used internally and should not be surfaced. There is no replacement for this attribute. • *Size* – removed. This attribute was only available on files and should be retrieved from the *RepositoryFile getSize* method.
	Remove duplicate attributes from *WorkItem*	In prior releases certain attributes were surfaced on the *WorkItem getAttributes* method call that were duplicates for properties available directly on the object. The affected attributes are: • *WorkItemContext* – removed. This should be retrieved from the *WorkItem* method *getContext*. • *WorkItemOwner* – removed. This should be retrieved from the *WorkItem* method *getOwner*. • *WorkItemStatus* – removed. This should be retrieved from the *WorkItem* method *getStatus*.
	All API DTOs should be immutable when possible	Nearly all API DTOs are intended to be immutable. This means once they are retrieved from an API service very few properties on the class can be changed. In the case where the values are immutable the fields should be declared *final* with no setter method. Changes have been made to move in this direction. Properties have been made final where possible, constructors that do not initialize all fields have been deprecated, new constructors have been added that do initialize all fields and unnecessary setter methods have been deprecated. The Data Transfer Objects that have been affected are: • *Analysis* • *AnalysisDescriptor* • *Project* • *ProjectDescriptor* • *Organization* • *OrganizationDescriptor* Since these objects were never intended to be constructed by consumers this should not be considered a breaking change. All methods and constructors targeted for removal have been deprecated and not removed but will be removed in a future release.

The following changes were made to functionality delivered in Release 1.6:

Reference #	Title	Description
US6636	Support for relative paths	Added support for relative path references for job input and output locations, file and folder parameters, and job log, listing, and manifest. Added a *RelativePathUtility* that can be used to generate a relative path for a file relative to the provided base. A relative path can be used in defining a *ResourceSpecification* within a job.
US6635	Add action to audit query	A list of audit actions can now be specified on an *AuditEntryQuery* when getting audit entries.
US6634	Add masked character job parameter type	Added support for a new SDD 4.3 job parameter whose value is masked as "*****" in the SDD UI and in the job log. CHARACTER_MASKED has been added as a *ParameterType*.
US6729	Schedule Job: add method to check if schedule exists	Added *scheduleExists* method to the *ScheduleService* to check whether a schedule with that same name exists.
US6660	Get list of users	Added *getAllUsers* and *getAllUserDescriptors* methods to the *UserService*. This includes newly created and inactive users.
US6730	Schedule Job: add method to check if a job is scheduled	Added *isRepositoryJobScheduled* and *isWorkspaceJobScheduled* methods to the *ScheduleService*.
US6662	Work flow report	The *WorkItemService* provides methods for accessing information on work items defined in the system. The methods *getAllWorkItemsByContext* and *getMyWorkItemsByContext* provide information like that shown in the SDD All Work Items/My Work Items dashboard tile. Additional service methods provide the ability to get work item details such as work flow definition, active tasks, completed tasks, and attachments.
DE3013	Weekly schedules now require time zone	SDD 4.3 requires a time zone selection for weekly schedules. This is a new String field on *ScheduleTrigger*. *ScheduleTimeZoneConstants* provides the list of time zones currently supported in SDD.
DE3050	Set permissions: ACL Updates to /Users containers	The *AclService updateAcl* and *updateAclContainer* methods now throw an *AclUpdateException* when ACL updates to items within user depots (/Users) are attempted.
	Subscription events	Because API 1.6 runs with SDD 4.3, two new subscription events are now available, *DepotContainerMovePostEvent* and *DepotFileMovePostEvent*. Additionally, *JobExecuteStartEvent* is now *JobExecuteStartPostEvent*, and *JobScheduledExecuteStartEvent* is now *JobScheduledExecuteStartPostEvent*.

The following changes were made to functionality delivered in Release 1.5:

Reference #	Title	Description
US4437	Export Audit Trail	Added *AuditService* to query audit trail entries.
US5397, US5398	Lock and Unlock File	Added *lockFile* and *unlockFile* methods to the *RepositoryResourceService*.
US4435	Batch check in and check out	Added *BatchRepositoryResourceService* which allows file and container paths to be specified for check in and check out. If a container path is specified, all files within that container will be included.
US4738	Event subscription	Added *SubscriptionService* to support creating subscriptions to select events for generating notifications.
US5917	Download as zip	Added *BatchRepositoryResourceService* and *BatchWorkspaceResourceService* which allows files and container paths to be specified for download to a zip file. If a container path is specified, all files within that container will be included.
US5918	Upload and expand	Added *BatchRepositoryResourceService* and *BatchWorkspaceResourceService* which allows a compressed file to be uploaded and expanded into the repository.
DE2125	*InvalidMemberException* no longer thrown in certain cases	*InvalidMemberException* is no longer thrown in the *MembershipService removeMember* method if a group is attempted to be removed from the context in which it was created.
	Deprecated methods removed	All deprecated methods have been removed. Please update existing code using deprecated methods.

The following changes were made to functionality delivered in Release 1.4:

Reference #	Title	Description

US5688	Compatibility updates for SDD 4.2	Code changes in SDD 4.2 required changes to the API code base. They are:
		
		• *MembershipService* – the *removeMember* and *removeMembers* methods throw an additional exception, *InvalidMemberException*. It is thrown when the member for removal is not an assigned member.
		• *MembershipService* – optimistic locking has been put in place for *updateMembership* which means if you try to update a membership and that membership is not the same one you got from the server (it was updated since you retrieved it), a runtime exception will be thrown.
		• *MembershipService* – *getAssignedMembers* in API 1.3 returned assigned and unassigned members. In API 1.4, *getAssignedMembers* returns only explicitly assigned members. To get all members, assigned and unassigned, use *getMembership(<context>).getMembers()*.
		• *MembershipDescriptor* – the context field type has changed from *MembershipContext* to *ObjectIdentity*. This is consistent with the context fields on *GroupDescriptor* and *RoleDescriptor*.
		• GroupService – the *createGroup* and *updateGroup* methods throw an additional exception, *InvalidGroupMemberException*. *PrincipalNotFoundException* is no longer thrown. *InvalidGroupMemberException* is thrown if a member is not found or a member is not a member of the current context.
		• *GlobalPrivilegeAssignmentService* - no code changes were made but *getAvailablePrivileges* now reflects global privilege changes in SDD 4.2. They are:
		    o Access WebDAV privilege added.
		    o Administer Work Items privilege removed.
		• *PrivilegeService* – no code changes were made to this method but the *getAvailablePrivileges* method now reflects the scoped privilege changes in SDD 4.2 for work item contexts. They are:
		    o Administer Work Items privilege added.
		    o Manage Work Items privilege added.

US5689	Support run-as functionality for jobs	The *JobService* has two new methods, *getRunAsOwner* and *setRunAsOwner*, which reflects the ability to set the run-as owner flag. This flag was introduced in SDD 4.2. By default, when a job is created, this value is false, and jobs are run as the submitter. Because the default in 4.1.1 was to run jobs as the owner, any 4.1.1 jobs migrated to 4.2 will have this flag set to true.

The following changes were made to functionality delivered in Release 1.3:

Reference #	Title	Description
US5396	Get, Set and Update Permissions	Added *AclService* and *com.sas.security.acl* package to support the getting and modification of permissions on repository containers and objects.
US5399	Schedule Job	Added *ScheduleService* and *com.sas.hls.schedule* package to support the scheduling of objects which support the Schedulable interface.
US5482a	Update user, project and analysis.	Added update methods to the *UserService*, *ProjectService*, and *AnalysisService*. This includes the ability to lock and unlock a user account, and to change the state of a project or analysis.
US5492b	Get Reachable Groups	Added the *getAllReachableGroupsByMember* and *getReachableGroupsByMember* methods to the *GroupService* so that non-direct group membership can be accessed via the API.
US4353	Rename object or folder	Added rename methods to the *RepositoryResourceService* and *WorkspaceResourceService* to enable the renaming of existing objects or folders.
US4736	Mark for add (single file)	Added the *markForAddAndCheckinFileFromWorkspace* method to the *WorkspaceResourceService*.
US4442	Manage Recycle Bin	Added *RecycleBinService* and *com.sas.hls.recyclebin* package to manage deleted items that the current user can access.

The following changes were made to functionality delivered in Release 1.2:

Reference #	Title	Description
US4578-a	Create, Rename, Delete Project	Added *ProjectService* and *com.sas.drug.project* package to manage the life cycle of a project.
US4578-b	Create, Rename, Delete Analysis	Added AnalysisService and *com.sas.drug.analysis* package to manage the life cycle of an analysis.
US4351, US4966	Copy File, Copy Folder	Added the ability to copy files and folders to workspace and repository resource services. *WorkspaceResource* and *RepositoryResource* objects now implement a marker interface *com.sas.hls.resource.Resource* which serves as a common parent for all resources in SAS Drug Development.
US4359	Create User, Delete User and Reset User Password	Added *UserService* and *com.sas.hls.security.user* package to support the creation and deletion of users. This service also supports resetting a user's password.
US4354, US4355, US4356	Create, Update, and Delete Group	Added *GroupService* and *com.sas.hls.security.group* package to manage groups within SAS Drug Development. Groups in SAS Drug Development are scoped to a context where a context is a location where a group can be defined such as Organization, Project or Analysis.
US5060	Assign Principals to Membership context	Added *com.sas.hls.security.membership*, *com.sas.hls.security.principal*, and *com.sas.hls.organization* packages and *MembershipService* to manage the membership of contexts. Principals may be assigned to membership contexts such as Organization, Project or Analysis.
US5071	Assign Global Privileges to Principals	Added *GlobalPrivilegeAssignmentService* to manage assignment of global privileges granted to users. These privileges are defined globally and not scoped to a context.
US4350, US4349	Create, Update Job	Added *JobService* and *com.sas.hls.job* packages to support the creation and modification of jobs in the workspace and repository. *Note: Deletion of jobs may be performed using WorkspaceResourceService or RepositoryResourceService.*
US5123	Job Execution	Added *JobExecutionService* to support execution of jobs in the workspace or in the repository. The current status of the submitted job can also be obtained.
US5127, US5128	Role Definition Role Assignment	Added *RoleDefinitionService*, *RoleAssignmentService*, *PrivilegeService* and *com.sas.hls.security.role* and *com.sas.hls.security.privilege* packages for managing role definition and assignment. *RoleDefinitionService* enables defining roles in SAS Drug Development at a context and inheriting roles from parent contexts through assignment of the defined roles to child contexts. A Role context can be an Organization, Project or Analysis. *RoleAssignmentService* enables assignment of Principals to roles assigned to

Reference #	Title	Description
		a context. *PrivilegeService* provides a way to get the set of context-specific privileges available for defining roles at a context.

The following changes were made to functionality delivered in Release 1.1:

Reference #	Title	Description
DE1559	DBCS characters and encoding issues in resource services	Resource file names and version comments containing DBCS characters were not handled correctly in 1.1. This problem has been addressed. *Note: End users must not perform any encoding or decoding of method parameters. Any previous reference to the UriAndPathUtils class (formerly shipped with the API) should be removed.* **For this reason, com.sas.hls.utils.UriAndPathUtils has been removed from the API.**
DE1350	*createParentFolders* not honored in some cases	The *createParentFolders* option was not honored in some file creation methods in workspace and repository resource services. This problem has been addressed.
	RepositoryResourceService method name change	Method *createContainer* has been changed to *createFolder*.