package org.sakaiproject.archive.api

public interface ArchiveService

ArchiveService takes care of exporting and importing entities.

## Package org.sakaiproject.archive.cover

public class ArchiveService

extends java.lang.Object

ArchiveService is a static Cover for the [ArchiveService](file:///C:\\Users\\administrator\\workspacefreeware\\common\\doc\\org\\sakaiproject\\archive\\api\\ArchiveService.html" \o "interface in org.sakaiproject.archive.api); see that interface for usage details.

### getInstance

public static [ArchiveService](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\archive\api\ArchiveService.html) getInstance()

Access the component instance: special cover only method.

## Package org.sakaiproject.importer.api

public interface Importable

public interface ImportService

The ImportService provides a top level framework to handled import data collected from a content package or other archive.

The default implementation of the ImportService, BasicImportService, keeps a list of available parsers and a list of available handlers, both of which are Spring-injected and configurable in a components.xml file. Here is how a hypothetical client of the ImportService might use it:

## Package org.sakaiproject.archive.impl

public class BasicArchiveService

extends java.lang.Object

implements [ArchiveService](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\archive\api\ArchiveService.html)

### archive

public java.lang.String archive(java.lang.String siteId)

Create an archive for the resources of a site.

Specified by:

[archive](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\archive\api\ArchiveService.html#archive(java.lang.String)) in interface [ArchiveService](file:///C:\\Users\\administrator\\workspacefreeware\\common\\doc\\org\\sakaiproject\\archive\\api\\ArchiveService.html" \o "interface in org.sakaiproject.archive.api)

### checkSakaiRole

public boolean checkSakaiRole(java.lang.String siteId,

java.lang.String userId)

When Sakai is importing an item archived by Sakai, check the creator's role first. The item is imported when the role is in the acceptance list,

## Package org.sakaiproject.importer.impl

The long-range plan is for Sakai to support the IMS Common Cartridge specification natively for both importing and exporting. The archive module will gradually acquire and consolidate all these capabilities, but for now it contains two distinct pieces: the original archive service, and the new import architecture. This document exclusively deals with the latter portion of this module.

The import side of things consists of five directories: import-api, import-impl, import-parsers, import-handlers, and import-pack. We will discuss each of these in turn, but first let us address a high-level overview of the approach to archive importing.

public class CommonCartridgeFileParser

extends [IMSFileParser](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\importer\impl\IMSFileParser.html)

## Package org.sakaiproject.importer.impl.translators

public class CCWebLinkTranslator

extends java.lang.Object

implements [IMSResourceTranslator](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\importer\api\IMSResourceTranslator.html)

Translating is a tedious job, and perhaps best to translate tool-by-tool. Try to do a cursory look over as many of the tool pages as possible to verify the correctness of the translation in context. If a translation doesn't look right, you can identify which property is mis-translated by selecting the 'en\_DEBUG' language locale in user preferences

## Package org.sakaiproject.importer.impl

only parser that is configured with Sakai is the original Sakai archive parser. The common cartridge parser is commented out. You can add common cartridge support to import by uncommenting the common cartridge bean.

public class SakaiArchiveDataSource

extends [BasicImportDataSource](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\importer\impl\BasicImportDataSource.html)

implements [SakaiArchive](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\importer\api\SakaiArchive.html)

The Sakai Site Archive feature has never been intended to be a general purpose import/export facility. However, when a recent project required this exact kind of behavior, the Sakai Site Archive facility was used as a starting point. This session will look at the the benefits of starting with Sakai's Site Archive system in order to create a general purpose import / export facility. We will look at how the system works with Sakai's modular tools and what problems/pitfalls come along with using it.

## Package org.sakaiproject.component.common.edu.person

public class FileSystemPhotoService

extends [BasePhotoService](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\api\common\edu\person\BasePhotoService.html)

FileSystemPhotoService to profile2 because it is our university policy that we are not aloud to automatically make an official photo a profile photo. This is something that the user has to manually choose to do (or select their own photo for their profile). Users are not aloud to change their official photo since it is used by professors to identify their students.  
  
I am hoping that profile photo and official photo are kept separate in the 1.4 series of profile2. That way we can use profile2 the same way we use profile1 now. So for now I think we will go back to using profile1 with the FileSystemPhotoService. Let me know if you have any questions for me on our setup with official/profile pictures. Profile2 does look like a nice improvement to the old profile and we are hoping to use it if we can work in our requirements to keep official and profile pictures separate.

## Package org.sakaiproject.component.common.manager

public class PersistableHelper

### setSessionManager

public void setSessionManager(org.sakaiproject.tool.api.SessionManager sessionManager)

Dependency injection.

Parameters:

sessionManager - The sessionManager to set.

public class EduPersonImpl

extends [InetOrgPersonImpl](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\component\common\edu\person\InetOrgPersonImpl.html)

implements [Person](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\api\common\edu\person\Person.html), [OrganizationalPerson](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\api\common\edu\person\OrganizationalPerson.html), [InetOrgPerson](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\api\common\edu\person\InetOrgPerson.html), [EduPerson](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\api\common\edu\person\EduPerson.html)

The eduPerson initiative is aimed at providing a standard way for institutions to list individuals in their LDAP

## Package org.sakaiproject.component.common.type

public class TypeManagerImpl

extends org.springframework.orm.hibernate3.support.HibernateDaoSupport

implements [TypeManager](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\api\common\type\TypeManager.html)

Implementations of TypeManagerImpl are included in the current Common Services.

The UUID service includes a single interface for a UuidManager. This interface as a

two methods, once which generates and returns a globally unique identifier, and aother

which takes a string and returns a Uuid.

### createType

public [Type](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\api\common\type\Type.html) createType(java.lang.String authority,

java.lang.String domain,

java.lang.String keyword,

java.lang.String displayName,

java.lang.String description)

Description copied from interface: [TypeManager](file:///C:\\Users\\administrator\\workspacefreeware\\common\\doc\\org\\sakaiproject\\api\\common\\type\\TypeManager.html" \l "createType(java.lang.String, java.lang.String, java.lang.String, java.lang.String, java.lang.String))

Create a new Type.

### getType

public [Type](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\api\common\type\Type.html) getType(java.lang.String uuid)

Description copied from interface: [TypeManager](file:///C:\\Users\\administrator\\workspacefreeware\\common\\doc\\org\\sakaiproject\\api\\common\\type\\TypeManager.html" \l "getType(java.lang.String))

Retrieve Type by uuid.

### deleteType

public void deleteType([Type](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\api\common\type\Type.html) type)

Description copied from interface: [TypeManager](file:///C:\\Users\\administrator\\workspacefreeware\\common\\doc\\org\\sakaiproject\\api\\common\\type\\TypeManager.html" \l "deleteType(org.sakaiproject.api.common.type.Type))

Remove the persistent state of passed Type.

### setCacheFindTypeByUuid

public void setCacheFindTypeByUuid(boolean cacheFindTypeByUuid)

Parameters:

cacheFindTypeByUuid - The cacheFindTypeByUuid to set.

## Package org.sakaiproject.api.common.edu.person

public interface EduPerson

extends [InetOrgPerson](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\api\common\edu\person\InetOrgPerson.html)

EduPerson is an auxiliary object class for campus directories designed to facilitate communication among higher education institutions. It consists of a set of data elements or attributes about individuals within higher education, along with recommendations on the syntax and semantics of the data that may be assigned to those attributes.

public interface SakaiPerson

extends [EduPerson](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\api\common\edu\person\EduPerson.html)

### getAcademicProfileUrl

java.lang.String getAcademicProfileUrl()

Get academic profile url

public interface SakaiPersonManager

### create

[SakaiPerson](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\api\common\edu\person\SakaiPerson.html) create(java.lang.String agentUuid,

[Type](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\api\common\type\Type.html) recordType)

Creates a persistent SakaiPerson record.

### findSakaiPerson

java.util.List<[SakaiPerson](file:///C:\\Users\\administrator\\workspacefreeware\\common\\doc\\org\\sakaiproject\\api\\common\\edu\\person\\SakaiPerson.html" \o "interface in org.sakaiproject.api.common.edu.person)> findSakaiPerson([SakaiPerson](file:///C:\\Users\\administrator\\workspacefreeware\\common\\doc\\org\\sakaiproject\\api\\common\\edu\\person\\SakaiPerson.html" \o "interface in org.sakaiproject.api.common.edu.person) queryByExample)

Query-by-Example finder signature.

Parameters:

queryByExample - A SakaiPerson protoype. All non-null preoperties will be searched using a logical AND.

public class BasePhotoService

extends java.lang.Object

implements [PhotoService](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\api\common\edu\person\PhotoService.html)

## Package org.sakaiproject.api.common.manager

public interface LifeCycleAware

### isInitialized

boolean isInitialized()

If other Managers are dependent on this manager, they may need to be aware of the initialization state. This can help resolve circular dependencies at startup.

public interface Persistable

### getCreatedBy

java.lang.String getCreatedBy()

The Agent that created this persistent object.

Returns:

UUID of the Agent that created this persistent object.

### getLastModifiedBy

java.lang.String getLastModifiedBy()

The last Agent that modified the persistent state of this object.

Returns:

UUID of Agent that made last modification.

### getUuid

java.lang.String getUuid()

All persistent objects must have a UUID.

Returns:

Returns the UUID for given object.

## Package org.sakaiproject.api.common.type

public interface TypeManager

Sakai types are defined using the Sakai Type Manager. See Common Services. The Sakai TypeManager provides a means to define and persist types. Type objects can be retrieved using either an Id or by keyword string.

### deleteType

void deleteType([Type](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\api\common\type\Type.html) type)

throws java.lang.UnsupportedOperationException

Remove the persistent state of passed Type.

### getType

[Type](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\api\common\type\Type.html) getType(java.lang.String authority,

java.lang.String domain,

java.lang.String keyword)

Retrieve a Type by the "tuple".

Parameters:

authority -

domain -

keyword -

## Package org.sakaiproject.component.api

public interface ComponentManager

Basic usage of the Sakai ComponentManager. This service is used to get access to other Sakai service. You can typically get access to other Sakai services using Spring Framework semantics directly as well. This is because the Sakai ComponentManager is basically just a thin wrapper around Spring.

### SAKAI\_COMPONENTS\_ROOT\_SYS\_PROP

static final java.lang.String SAKAI\_COMPONENTS\_ROOT\_SYS\_PROP

The java system property name where the full path to the components packages.

### close

void close()

Close the component manager, shutting down any created singletons.

### get

java.lang.Object get(java.lang.Class iface)

Find a component that is registered to provide this interface.

public interface ServerConfigurationService

ServerConfigurationService provides information about how the server is configured.

The Server Configuration Service provides information about the server itself, including any custom properties used by our application.

1. Create a new RSF Hello World app using the App Builder.
2. Add the component-api to the build path.

only what the Server Configuration Service read in from the sakai.properties

### getAccessPath

java.lang.String getAccessPath()

Access the path to the access service on the server relative to the base URL for the server.

### getAccessUrl

java.lang.String getAccessUrl()

Access the URL to the access service on the server - append any additional path to the end

### getConfigData

[ServerConfigurationService.ConfigData](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\component\api\ServerConfigurationService.ConfigData.html) getConfigData()

Returns data about all the configuration values which are known to the system at the time and some stats which are useful

public static interface ServerConfigurationService.ConfigData

Defines the config data holding class

public static interface ServerConfigurationService.ConfigHistory

Defines the config item history class

public static interface ServerConfigurationService.ConfigItem

Defines the config item holding class

public static interface ServerConfigurationService.ConfigurationListener

Allows a service to be notified when configuration settings are changed, It is up to the implementor to ignore the changes they do not care about. Filter on the

public static interface ServerConfigurationService.ConfigurationProvider

Allows registration of configuration settings (config items) from outside the Server Configuration Service and the standard set of properties files which will be loaded early on the configuration cycle NOTE: the implemented ConfigurationProvider MUST be a Spring singleton and it MUST be registered in the main Sakai application context from a component (from a webapp will not work), it also must be set explicitly to not lazy initialize (lazy-init="false"), it is always possible to update the configuration later using#registerConfigItems(ConfigData) so this is mainly for loading configurations very early in the system startup

public static class ServerConfigurationService.ConfigurationListener.BlockingConfigItem

extends java.lang.Object

implements [ServerConfigurationService.ConfigItem](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\component\api\ServerConfigurationService.ConfigItem.html)

This is a special marker class that is used in the [ServerConfigurationService.ConfigurationListener](file:///C:\\Users\\administrator\\workspacefreeware\\common\\doc\\org\\sakaiproject\\component\\api\\ServerConfigurationService.ConfigurationListener.html" \o "interface in org.sakaiproject.component.api), returning this indicates that the config change should stop processing the change and retain the original value

## Package org.sakaiproject.component.impl

public class ContextLoader

extends org.springframework.web.context.ContextLoader

Sakai's extension to Spring's ContextLoader - adds the location of the ComponentManager shared AC, linking the local AC to it as parent, and loading localy hosted components into shared.

public class DynamicDefaultSakaiProperties

extends java.util.Properties

Handle any default sakai.properties values that need to be set dynamically.

public class MockCompMgr

extends java.lang.Object

implements [ComponentManager](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\component\api\ComponentManager.html)

This is a totally fake component manager which is just going to pretend to try to find services, maybe later this could try to use the sakai mock stuff but for now it is just here to stop the CM from firing up during unit tests

public class SpringCompMgr

extends java.lang.Object

implements [ComponentManager](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\component\api\ComponentManager.html)

SpringCompMgr manages API implementation components using the Springframework ApplicationContext.

## Package org.sakaiproject.component.locales

## sakai.properties set locales.more = en\_US\_DEBUG to display the properties key/value pairs – you will need to select this locale in preferences

public class SakaiLocales

extends java.lang.Object

This stores constants related to the Sakai default locales

### SAKAI\_LOCALES\_DEFAULT

public static final java.lang.String SAKAI\_LOCALES\_DEFAULT

This should be complete list of supported locales and should match the commented list in default.sakai.properties It MUST be a comma separated list of locale keys (be careful with your formatting)

public class PrivacyManagerImpl

extends org.springframework.orm.hibernate3.support.HibernateDaoSupport

implements [PrivacyManager](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\api\privacy\PrivacyManager.html)

### findHidden

public java.util.Set findHidden(java.lang.String contextId,

java.util.Set userIds)

Description copied from interface: [PrivacyManager](file:///C:\\Users\\administrator\\workspacefreeware\\common\\doc\\org\\sakaiproject\\api\\privacy\\PrivacyManager.html" \l "findHidden(java.lang.String, java.util.Set))

Get a set of users who have privacy enabled within a context. (Hidden)

## Package org.sakaiproject.importer.impl.importables

public class Announcement

extends [AbstractImportable](file:///C:\Users\administrator\workspacefreeware\common\doc\org\sakaiproject\importer\impl\importables\AbstractImportable.html)

This is a generic Announcement type object which holds announcements during the migration/import process.

These generic content objects still have types which identify what kind of content they are. Some examples of these content object types are: WebLink, Assessment, Announcement, etc. Each content class must implement an interface called Importable. As a group, these content objects are referred to as importables. An object called an ImportDataSource acts as a container for the group of importables. We can ask an ImportDataSource for the categories for that archive, and ask it for the Importables that match a given list of categories.

An importable is a content object that implements the Interface org.sakaiproject.importer.api.Importable. Importables are extracted as a collection by a parser, and passed one at a time to one or more handlers to be added to Sakai.

parse an archive file and get back an ImportDataSource

## https://confluence.sakaiproject.org/download/attachments/20218688/import_overview_2.png?version=1&modificationDate=1158777260000

## pass a collection of importables to the handlers, which then push content into Sakai.