

Working Draft 0.2**IdAS Interfaces and Methods**

An attempt to complete the list of methods required for each interface. ContextProvider Wiki page (<http://spwiki.editime.com/ContextProvider>) is taken as the starting point for requirements.

IContext

re: DigitalSubjects

- addSubject(IDigitalSubject)
- getSubject(Identifier uniqueID, boolean recursive) // note: Identifier == Attribute
// note: recursive specifies whether to search subcontexts
- removeSubject(Identifier uniqueID, boolean recursive) // do we need both remove's?
- removeSubject(IDigitalSubject, boolean recursive)
- getSubjects(boolean recursive)
- getSubjects(Set<Attribute> searchAttr, boolean recursive) // optional?
- getSubjects(Identifier, boolean recursive) // could be non-unique, matching multiple DigSubs
- updateSubject(IDigitalSubject new, IDigitalSubject old)
- getSubjectSchema() // format? same for all Subcontexts?
- setSubjectSchema() // format? Subcontexts?

re: Subcontexts

- addSubcontext(IContext)
- removeSubcontext(IContext)
- getSubcontexts(IContext)
- getParentContext() // is there only one? if general graph, then add/remove
- setParentContext(IContext)

re: Relationships – distinction between related Contexts and Subcontexts?

- getRelationships() // set of ContextRelationships
- addRelationship(IContextRelationship)
- removeRelationship(IContextRelationship)

re: Policy – defines who can do what to Context and DigSubs

- getPolicy() // what gets returned here?
- setPolicy() // what format?

re: RDF

- exportRDF() // what output format? do we always dump the entire context?
- importRDF() // totally replace or append to existing Context info?

IContextFactory

re: Contexts

- openContext(URI contextRef, Identifier requester) // arguments?
 - Could be factory-specific, but is there a minimum required set?
 - RDF stream
 - Context URI (could point to “remote” context)
 - Requirements mention authentication, and resolving context references.
- closeContext(IContext)

IBM Confidential

re: Policy – who can open contexts, etc.

- getPolicy() // format?
- setPolicy() // format?

IDigitalSubject

re: Context

- getContext()
- no setContext() // specified during construction? immutable (diff Context means diff DigSub)

re: Identifiers and Type

- getType()
- setType(String)
- getUniqueIdentifier()
- setUniqueIdentifier(IIdentifier) // should this be set during construction? immutable?
- getIdentifiers() // set includes unique ID
- addIdentifier(IIdentifier)
- removeIdentifier(IIdentifier)

re: Attributes and Relationships

- getAttributes()
- addAttribute(IAttribute)
- removeAttribute(IAttribute)
- getRelationships()
- addRelationship(ISubjectRelationship)
- removeRelationship(ISubjectRelationship)

IAttribute

- getType() // URI (or String?)
- setType(URI)
- getValue() // Object
- setValue(Object)
- getMetadata() // IMetadata
- setMetadata(IMetadata) // replaces existing Metadata container

IIdentifier (extends IAttribute)

- identifier-specific methods? (e.g., getCountry, getOrganization, ...)

IRelationship

- getType() // URI
- setType(URI)
- getMetadata() // IMetadata
- setMetadata(IMetadata) // replaces existing Metadata container

IContextRelationship (extends IRelationship)

- getRelatedContexts()
- addRelatedContext(IContext)
- removeRelatedContext(IContext)

ISubjectRelationship (extends IRelationship)

- getRelatedSubjects()
- addRelatedSubject(IDigitalSubject)
- removeRelatedSubject(IDigitalSubject)

IBM Confidential**IMetadata**

- `getItems()` // set of `IMetadataItem`s
- `addItem(IMetadataItem)`
- `removeItem(IMetadataItem)`

IMetadataItem

- `getType()` // URI
- `setType(URI)`
- `getValue()` // Object
- `setValue(Object)`

Exceptions

- One exception class (`IdasException`) with specific error codes?
- Multiple exception classes?
- Extend/use standard exception classes?

Open Questions

- Specifying search parameters for `IContext::getSubjects`. Jim's strawman (and list above) specifies a set of attributes. Is that general enough? What about searching for a range of attributes, or other constraints?
- Search method for retrieving Attributes from a `DigitalSubject`?
- Is Identifier a separate interface type? Or is it merely an Attribute used in a particular way? (If an Attribute, should it be considered when searching Attributes, or should it be kept separate?)
- How should policies be represented?
- What arguments are required for `IContextFactory::openContext()`? Currently, we list a context reference, and the identity of the requester. The Identifier could carry authentication tokens in its metadata, or should there be a specific parameter for authentication info?
- When should URI be used vs. String?
- When creating the RDF representation of a Context, can we somehow specify that only part of the Context should be exported? (E.g., entity that opened context may not have permission to see entire context, or may want to programmatically extract part to create a new subcontext.)
- Is Subcontext simply a particular kind of `ContextRelationship`? In the list above, it is treated differently, because we provide utility functions for searching/retrieving/managing Subcontexts.