

Global Format Registry Use Case: Improve Format Identification

Use Case ID	Penn-4
Description	Improve method for determining the representation format of object
Actors	<i>Registry</i> of digital format information <i>Repository</i> of digital objects <i>Maintainer</i> of repository's format identification service
Assumptions	We aim to improve reliability or performance of Harvard-1 use case. The repository has a way of identifying formats (either on its own, or by calling a service maintained elsewhere, such as one used by a Broker). The identification service uses an extendable ruleset to ID formats.
Preconditions	Registry has identification attributes not yet reflected in ID service
Triggers	Maintainer sees need for a Repository to identify a new format, or correct misidentifications.
Primary Scenario	Step 1 Maintainer requests and receives identification attributes (possibly in the form of rules) from Registry. Request may be complete, as in the harvard-1 use case, or selective, such as OAI-style "changes since last request", or "just the attributes for these formats that I'm interested in"
	Step 2 Maintainer uses attributes to update identification service's ruleset. Maintainer adjusts priorities of rules so that more efficient, discriminating, or reliable rules get invoked first. Maintainer adjusts both priorities and rule content to properly distinguish similar formats.
	Step 3. Maintainer uses Repository content to test service and its ruleset.
	Step 4. Maintainer makes new service available to Repository.
Primary Result	Repository has more complete and/or accurate format identification.
Post-Conditions	Repository users have improved success in identifying formats.
Non-functional requirements	As the harvard-1 use case notes, the primary job of the registry is to make format identification information available. The rule*set* used for ID is Repository-specific, based on policy, contents, and format repertoire. Rulesets and ID services can be shared among Repositories if desired.
Notes	TOM supports format identification and validation via content pattern-matching rules (via Perl regular expressions), file extensions, MIME types, and Macintosh type/creator patterns. Rules can be noted as necessary or sufficient, and prioritized. I've noticed that as format repertoires develop, rulesets need to be "tuned" (rules tightened or loosened, and/or priorities reworked) to keep ID accurate and efficient.
Issues	Scalability remains a concern both for reliability and efficiency. Intermediate-goal rules (e.g. "This is an MS-Word format, but we're not sure which version") may help prevent the complexity of format assignments from growing linearly with the number of formats, though writing good algorithms for this kind of ruleset can be tricky.