View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Illinois Digital Environment for Access to Learning and Scholarship Repository

Repository Findings Commentary Using the Annotated Trusted Digital Repository Checklist

The table below is essentially our annotated checklist integrated with comments resulting from testing and evaluation of four repository software applications: DSpace, Eprints, Fedora, and Greenstone.

Correlations to TRAC requirements	TDR Modified Checklist	
	A. The Organization	
	A5. Contracts, Licenses and Liabilities	
	A5.1 If repository manages, preserves, and/or provides access to digital materials on behalf of another organization, it has and maintains appropriate contracts or deposit agreements. *The following are examples of how this may be applied to repository software applications:*	
	a. Does the repository software application have any means to manage, store, or enforce these contracts or deposit agreements? b. Does the repository software application tie specific agreements to individuals or individual items in the repository or to collections of items in the repository?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	DSpace requires submitters to click through a license agreement assigning the repository re-distribution rights to the item. These agreements may be customized by community or collection. DSpace Foundation publishes as a sort of case study the policy guidelines for DSpace at MIT LIbraries.
	EPrints	EPrints does not seem to encourage or prohibit third-party agreements. Its focus is on the individual who self-archives "digital texts of peer-reviewed research articles" (from website). The software does not appear to address management of third-party digital materials.
	Fedora	An out-of-the-box Fedora installation has no explicit means of maintaining contracts or deposit agreements. However, agreements such as these could be stored as datastreams along with the content to which they pertain, and encapsulated inside of the same digital object.

	Greenstone	Greenstone does not currently have such means, but because of its flexible metadata structures and plug-in architecture, customization could probably be done to support this type of functionality.
	A5.3 Repository tracks and manages copyrights and restrictions on use as required by contract or license or deposit agreements. a. Does the repository software application have any capabilities—such as access control lists, Internet address filters, etc.—that can be used to enforce copyright or access restrictions? b. How granular are these access controls? (For example, can different restrictions be applied to different objects in the repository?)	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	The DSpace installation supports access control with community, collection, and bitstream granularity. Access can be restricted to specific groups and users. Embargos or other time-based restrictions are not built into the software and must be manually mediated. Technical information on DSpace Authorization is included in the system documentation.
	EPrints	The software does not directly manage access, but it does have the capability to limit access to certain archives. It allows you to use HTTPS protocol. So, indirectly, EPrints allows you to filter access. Of course, other security measures external to the repository installation could control access as well. In general, EPrints, a self-archiving repository, assumes copyright/access belongs to the author and encourages open access. See the "Self-Archiving and Copyright" section on their website for a more complete explanation.
	Fedora	As of version 2.1, Fedora's access controls are specified as eXtensible Access Control Markup Language (XACML) policies. Fedora administrators may specify finely grained, machine-readable policies in XACML to control access to Fedora web services, digital objects, datastreams, and disseminations. Policies can be written to permit or deny access based on attributes of the user, attributes of digital objects, and attributes of the environment. Additionally, Fedora supports standard user-authentication through Tomcat's users file or an LDAP directory.
	Greenstone	Greenstone does not currently support access restrictions. The method they suggest to service providers is to use password protected directories and .htaccess-like security for the access-restricted materials. Using .htaccess you would have different directories with different list of users with passwords.
	B. Repository Functions, Processes & Procedures	
	B1. Ingest/acquisition of content	

B1.1. Repository identifies properties it will preserve for each class of digital object.

a. What kinds of metadata (Representation Information) does the repository software application support out-of-the-box?

b. How easy is it to find out what kinds of metadata it supports?

c. How easy is it to customize the repository software application to support other kinds of metadata?

d. Does the repository software application provide a means of referring to external metadata registries? E.g., the Global Digital Format Registry (GDFR).

**** *See also B.3.3.*

Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
DSpace	File or bitstream-level properties may be reviewed and modified interactively through the web application. DSpace supports a modified version of Dublin Core; arbitrary qualifiers can be added to existing elements and new elements can be added. Custom metadata can also be accommodated through the Metadata Schema Registry. Arbitrary namespace URLs can be entered and assigned a namespace prefix, however, the definition of the elements must be done through DSpace and must be in the form of unique element and qualifier pairs.
EPrints	EPrints is designed for "digital texts of peer-reviewed research articles" (from website). So, its default content types are HTML, PDF, PS, and ASCII. These types are associated with the primary object in the package, which is presumably the text-based article. The supported structure is flat. However, you can add any other content type fairly easily. We added an xml content type, and we associated this type as the primary object. Of course, all other files in the package—whatever their content type— could be ingested. So, except for the designated, primary object, EPrints can be agnostic with respect to content type. EPrints supports (outputs) basic Dublin Core metadata via OIA harvesting. EPrints uses its own metadata schema as input and for use within EPrints. Customization is possible. Note: The input and output metadata schemas are greatly enhanced in version 3.0 and greater. Plus, in the later version, much more input/output format customization is possible.
Fedora	Fedora Digital Objects are made up of one or more datastreams: content items of any media type, which can be stored locally in the repository or externally referenced by the digital object. The content of a datastream can be MIME-typed data or metadata. All Fedora Digital Objects have a reserved Dublin Core datastream associated with them, as well as a special datastream that records an audit trail of all changes made to the object. In addition to the reserved datastreams for object metadata, any number of custom metadata streams may be included in the digital object as well. The Format URI attribute of Fedora Digital Objects allows for referencing external metadata registries, such as the Global Digital.
Greenstone	Greenstone supports many file formats out-of-the-box and through the use of plug-ins. Greenstone is flexible with regards to supporting representation information. Greenstone does have a default set of metadata fields based upon Dublin Core. Plug-ins can be copied, expanded, and completely changed to accommodate the metadata needs of a collection. Greenstone does accommodate different metadata needs. It also can export a collection's metadata to be used with OAI or DSpace.

B1.3. Repository has an identifiable, written definition for each SIP or class of information

ingested by the repository.

The following are examples of how this may be applied to repository software applications:

a. Are the supported file formats well documented?
b. Are the supported metadata formats well documented?
c. Can new file formats be added or removed?
d. Can the metadata fields or formats be customized?

Notes on this Checklist Item from Repository Testing and Evaluations in 2007

DSpace	DSpace submission units are *items* consisting of *bitstreams* (which can be any file format), plus a contents manifest and descriptive Dublin Core metadata. However, the DC metadata must be transformed to a DSpace-specific variant based upon DC-LAP. Technical details of item submission are thoroughly documented. DSpace is file format-agnostic with respect to content. As described in B1.1, metadata can be customized. Work is now being done on developing a METS SIP that can be used for ingest: http://cwspace.mit.edu/docs/xsd/METS/SIP/profilev0p9p1/metssipv0p9p1.pdf
EPrints	Submission units are objects consisting of at least an EPrints metadata file plus the primary object. The associated files (e.g., GIFs associated with the primary object in HTML format) are not necessarily required to be ingested. Of course, in the case of the HTML file, the primary object may not be displayed properly if the associated files are not included. In our case, we viewed EPrints submission units as objects consisting of datastreams plus the EPrints metadata file. We constructed our own script, which somewhat enveloped the default, batch "import_eprints" script that comes as a PERL batch executable file with the EPrints software. The software allows for us to add file formats (as I've mentioned in B1.1). Though not well documented, EPrints supports any file format since the files are just stored as bitstreams in the file system. It also allows us to customize any of the metadata fields, adding new ones if needed. I recommend that since EPrints strongly encourages the repository to be OAI compliant, care should be taken when metadata fields are changed or especially when thinking about deleting fields.
Fedora	Fedora submission units are encoded for ingest in Fedora Object XML (FOXML) or the Metadata Encoding and Transmission Standard (METS). In upcoming releases, Fedora will also support other formats for ingest such as MPEG21/DIDL. Both FOXML and Fedora-METS are adequately documented and include schema definitions. Like DSpace, Fedora is file format-agnostic with respect to content.
Greenstone	While Greenstone does not include a well-defined Submission Information Package, it does incorporate plug-ins for many different file formats. Greenstone is designed specifically to add support for new file formats and also to customize metadata fields.

B1.4. Repository has a process to ensure that the information is acquired from the expected source.

a. Does the repository software application have capabilities that limit who is allowed to submit items? For example, access control lists, internet address filters, etc.
b. Does repository software application have capabilities to limit who is allowed to modify or delete Digital Objects or Representation Information in the repository?

c. Does repository software application have any means to verify that Digital Objects or Representation Information have not been tampered with from initial receipt to ingestion? For example, through the use of checksums or digitally signed checksums.

d. Does repository software application maintain audit logs that identify by whom and when all changes to the Content Information were made?

Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
DSpace	DSpace requires an *eperson* for item submission. Submission privileges may be established at the time a collection is created. Similar restrictions may apply to modifications and deletions. DSpace associates MD5 checksums with every file in its asset store, and periodically checks that these have not changed. There is a workflow that involves the review of new submissions; administrators will be notified of new submissions and must approve them before they are made public. The log file using the default settings will log authentication attempts, changes made to content, and various other actions taken listing the session ID and the e-person who performed these actions.
EPrints	EPrints has this process of submission focused on self-archiving. This process, at least on the web side, usually involves a staging process. An author registers as an author in a workspace area and submits an EPrints article to the workspace area. The author needs to provide a valid email address and creates an EPrints account and password. Once an EPrints is submitted, an editor must move the item to the repository. The editor can make changes to the article before moving it into the repository, or the editor can reject the article and move it back to the author's own account for perhaps more changes. Each EPrints user is assigned a role (such as author, administrator, etc), which controls its access rights. It is possible to bypass the staging process and have the author deposit the article directly into the repository. In the batch process, we bypass the staging level and submit the articles/packages directly into the repository. We've created a batch-import user name, which basically has editor rights. As far as audit and logs, EPrints provides some limited probibty features that track when files have been created and changed. This tracking is automatic at a file level. EPrints produces an xml file containing the probity information for each record. EPrints names this file with a date/time stamp. So, there is a very crude audit trail. However, this trail does not include much else (i.e., no change agent is indicated). EPrints also can run this check manually on a record/package level, storing the checksum in an xml format. There is no way to tell if a file has changed *outside of the EPrints software control* unless one compares checksums-- whether created manually or automatically-- before and after the change. EPrints does provide a limited submission logging on submissions. It logs timestamps of users doing the submission process. It may be useful to monitor time taken on various pages in the submission process on the web.
Fedora	The Fedora repository service has the capability to limit who is allowed to submit items through individual user accounts, basic HTTP authorization, and finely-grained XACML policies. Fedora provides access to Digital Objects through the access and management web services: the Fedora Access Service (API-A) and the Fedora Management Service (API-M). The major function of the Fedora Access Service is to fulfill a client's request for dissemination, while the Fedora Management Service defines an interface for administering the repository, including creating, modifying, and deleting digital objects. Fedora limits access to API-A and API-M web services through authentication, SSL, and XACML policy authorization. As of version 2.2, Fedora provides the capability of datastream checksumming to verify that the contents of Digital Objects have not been

		changed, and all Fedora Digital Objects contain a special datastream that records an audit trail of all changes made to the object.
	Greenstone	Greenstone does not currently have capabilities to limit who can submit items. Greenstone limits who is allowed to modify or delete to the administrators of the installation. It currently only supports two roles: administrators and information seekers. Greenstone contains a security system, which forces people who want to build collections to log in first. Greenstone does not currently have any means to detect tampering with Digital Objects. Collection items can only be accessed by the administrators. Greenstone does currently have some logging capabilities, such as web request logging and also event logging associated with the Collector.
	B1.5. Repository obtains sufficient physical control over digital objects to preserve them. a. Where does the repository software application store the actual digital objects and their Representation Information? b. Does the storage medium itself provide for sufficient security and reliability such as clear access controls on file systems and database systems?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	DSpace stores content within its *asset-store*, a designated filesystem area. File names within the asset store are hashed and obfuscated. The asset store is subject to the security and reliability limitations of the operating system's filesystem, as long as the DSpace user can access this area the application will work. Storage can also be set up to use the Storage Resource Broker (a Data Grid Management System), in which case, the hardware is not under control of the repository. Metadata and file locations are retained within the PostgreSQL database. This separation of these components can make it difficult to re-create the object (match items with their metadata) if one of these components are corrupted.
	EPrints	EPrints also stores all data & metadata within its *asset store*, a designated filesystem area. The file names within this asset store match up with an internal EPrints Id. For example EPrints Id=123456 would match to the directory, "disk0/00/12/34/56". EPrints does support groupings of items into a collection (EPrints calls these collections "archives"). EPrints has a MySQL database to augment retrieval and management plus some administrative metadata. The separation of the files from the database component can make it difficult to restore an object if one of the components is corrupted.
	Fedora	Fedora Digital Objects are under the direct custodianship of the repository, and reside in its data store, a simple file system assigned for this purpose. Filenames are somewhat obfuscated. In addition to the file system, an SQL database contains the set of registries and metadata that enable searching, and an RDF triplestore holds properties of digital objects, datastreams, disseminations, and relationships to create an RDF-based index of the repository. If the database or triplestore were to become corrupt or deleted, they can be restored with the Fedora Rebuild utility by crawling the object metadata datastream components that make up every Fedora Digital Object on the file system. Since version 2.2, Fedora comes with the capability to compute and store checksums to verify that the contents of objects have not been changed. However, it is the responsibility of the system administrator to provide disaster protection and data security for the file system itself, through careful planning and proper backups.

	Greenstone	In Greenstone content is stored on the file system under the directory of its parent collection. Security and reliability is dependent on the underlying file system and the processes and policies adopted by the hosting organization.
B1.4 Repository's ingest process verifies each submitted object (i.e., SIP) for completeness and correctness as specified in B1.2. ***Note: In TRAC, B1.6 now reads, "B1.6 Repository provides producer/depositor with appropriate responses at predefined points during the ingest processes." (That is, old B1.7 maps to new B1.6—see row below.)***	**B1.6.** Repository's ingest process verifies each SIP for completeness and correctness. a. Does the repository software application verify the file format against the actual file that was submitted? b. If a SIP does not conform to the accepted formats what happens? c. Are there automated checks of the metadata such as to verify that a date entered into a field really is a date string? How does the repository software application verify that file format metadata is correct? d. Does the repository software application support workflow so that human reviewers can verify data after it is deposited, but before it is 'officially' accepted? e. How does the repository software application deal with incomplete data? With digital objects lacking sufficient or appropriate metadata – such as an XML file that references a DTD or XML Schema, but the DTD or Schema is not available? f. How does the repository software deal with inconsistent information – e.g., byte encodings of the content? For example, a Digital Object that is an XML file might incorrectly list some of its metadata Representation Information, such as its encoding scheme, as in the case of an XML declaration at the top of the file stating the file is encoded as UTF-8, when it is actually encoded as UTF-16.	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	Because DSpace is agnostic as to file-format, there is no mechanism to check that it is supported; likewise, no checks are made to determine that the submitted files is conformant to its specification. Aside from verifying compliance with Dublin Core, DSpace does not check metadata for correctness. It does check for well-formed XML metadata. Otherwise, bitstreams are not checked for format compliance, nor are files cross-checked for interoperability. In other words, if an XML bitstream requires an absent XSLT bitstream for formatting and display, this is not flagged by the DSpace Item Importer. Encoding issues in the item-level Dublin Core metadata may cause the Item Importer to throw a runtime exception. Otherwise, encoding issues are not detected.
	EPrints	EPrints does not check data or metadata for correctness. It may require, however, certain metadata fields to be present in order for a record/package to be ingested. These "required" fields are set up in the EPrints configuration file(s). All error-checking, it seems, will have to occur before getting to the actual EPrints ingest. For incomplete data, EPrints will leave the missing fields as null. EPrints does not seem to identify the inconsistency in the input data.
	Fedora	Most of the DSpace comments apply to Fedora. However, Fedora enforces compliance of FOXML and Fedora-METS to customized schemas, so metadata correctness-checking is notably more rigorous. It would be possible to add an object and format validation service such as JHOVE to the Fedora Service Framework, but no such service is included with Fedora out-of-the-box.

	Greenstone	Greenstone does not have a well-defined SIP, although it can ingest METS files that are conformant to its profile. Greenstone also supports the use of a metadata.xml file that provides metadata about the individual items in a collection. Greenstone does not currently verify correctness of digital object or representation information. The collection-building process in Greenstone does check each item in the collection for valid format. If one item is not valid, the entire build process fails. This is a drawback of the software. Greenstone 3, with incremental builds, may solve both problems. Greenstone, version 2, does not currently support a robust workflow. This is in the list of things they would like to do. Right now Greenstone leaves all data verification to the administrator of the software. Greenstone currently treats XML files as text files. DTDs and schema-checking are not supported. The exception to this is the metadata.xml file and the documents in Greenstone Archive Format. Greenstone does notice when encoding declarations are incorrect. This might be related to the plug-ins and the Perl scripts that Greenstone incorporates into the software.
B1.6	**B1.7.** Repository provides Producer/depositor with appropriate responses at predefined points during the ingest processes. a. How does the repository software application notify a producer/depositor that their submission has been rejected or accepted into the repository? b. Does the repository software application notify a producer/depositor of what is needed to have submission accepted? c. Does the repository software application monitor or guide the workflow?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	The ingest process is configurable to an extent (more-so now thanks to a patch contributed which allows workflow configuration with an XML configuration file). DSpace administrators can specify custom messages to be displayed in the submission process, and certain steps in the ingest process cannot be completed without certain information present. If the submission acceptance is mediated by an administrator, an e-mail will be sent to the depositor indicating acceptance.
	EPrints	Our batch scripts inform the person who runs these scripts the results of the import, via log files. A user who inputs an article manually via the web interface will be responded to properly at appropriate intervals when submitting an article to the staging area. Once submitted to the staging area, appropriate editor(s) are emailed. Once an editor deposits an article into the repository or sends an article back to the author/submitter, EPrints does not necessarily automatically email notification to the submitter.
	Fedora	The Fedora Directory Ingest Service, part of the Fedora Service Framework, constructs Fedora objects from uploaded SIPs and ingests those objects into a Fedora repository. The Directory Ingest Service is implemented as a REST-based web service accepts input via HTTP POST and returns an XML document with the list of PIDs (persistent identifiers for Fedora Digital Objects) of the successfully ingested objects.
	Greenstone	Greenstone does supply information during the collection building process. It also creates a log of the errors and build information.
	B1.8. Repository can demonstrate that all SIPs are either accepted as whole or part of an	

	eventual AIP, or otherwise disposed of in a recorded fashion. How does the repository software application demonstrate that all SIPs are accepted as whole or part of an eventual AIP or otherwise disposed of in a recorded fashion? For example: a. When does a submission become managed by the repository software application itself – if it does? b. Does the repository software application record in an audit log each submission and transition from a SIP to an AIP? If so, is the audit log protected from intentional or inadvertent tampering? c. Does the repository software application support workflows such that a human can formally accept a package into the repository? If so, how does the software record this formal acceptance? d. Does the repository software application periodically send audit log reports to appropriate administrators? e. How does the repository software application keep track of accepted deposits vs. rejected deposits?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	The submission workflow requires that an administrator accept a submission. If a submission is rejected for any reason, the submitter will be notified and the item will be put into their "in-process"queue where it will remain until they attempt resubmission or remove it from the workflow.
	EPrints	As we mentioned in B1.4, EPrints provides only limited audit/logging facilities for any changes to the packages. It does not appear to provide a history of metadata changes within the MySQL database. It does have a datestamp metadata field. So, you're able to tell the last time metadata for a given EPrints record has been last changed. The transformation to an AIP is not explicitly logged in EPrints. The AIP is "fractured." In other words, the package and access to the package is split in EPrints between several directories and the MySQL database. The transformation to an AIP is logged implicitly in the datastore (as the filesystem datetime stamp). No description is available.
	Fedora	Once a SIP has been ingested into Fedora, the underlying repository system handles the details of storing datastream content within the repository. Although, the Fedora Management Service allows users (with the proper security permissions) to modify and delete digital objects, or components within digital objects. Inside the repository, a content versioning system keeps an audit trail of how objects have changed, and maintains a backup copy, or version, of each datastream and disseminator in a digital object. Any modifications made to a datastream or disseminator will automatically result in the creation of a new version, marked with a date and time stamp. Every Fedora object contains a record of all versions, forming a history of how objects have changed over time. The repository also maintains an audit trail record of the nature of the object change events.
	Greenstone	Greenstone does not have a well defined Submission Information Package or Archival Information Package. Collections are built by the digital librarian using the Librarian interface to gather and describe the various components that make up a digital object. Presumably it is this person's responsibility to ensure a digital object is whole and complete as represented in the Greenstone interfaces.
B2. Ingest: Creation of the Archival	**B2.** Archival storage: management of archived information	

Package	
	B2.1. Repository has an identifiable, written definition for each AIP or class of information preserved by the repository. a. Does the repository software application have written documentation that describes what data structures it manages (e.g. primary object and alternate versions vs. compound objects with embedded images)? b. Does the repository software application have written documentation that explains the storage models it supports? c. How much control over the data formats, data structures, and storage models does the software package provide? Can new classes of digital objects be added or removed? Can the metadata or digital object formats be customized?
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007
	DSpace: DSpace does describe their data model. It specifies Items in this way: "Items are further subdivided into named bundles of bitstreams. Bitstreams are, as the name suggests, streams of bits, usually ordinary computer files. Bitstreams that are somehow closely related, for example HTML files and images that compose a single HTML document, are organized into bundles." Specific uses of this structure are not specified or enforced, however. The "archival storage" of items in DSpace is made up of the interplay between the file storage (on the file system) and the related information contained in the database. Intricacies of each storage method (filesystem/SRB) are specified in the documentation of DSpace architecture. This architecture is not customizable, all items are considered to be comprised of have so new Objects can not be defined.
	EPrints: EPrints documents its structure and the contents of its structure on its web site. You can get more information via their email forums and more still from the actual configuration files and code. The formats of objects can be added or deleted. EPrints seems to do little object-specific processing except to label the object as having a particular format. EPrints does not enforce the format. The actual database structure is not discussed much at all. In general, long-term archival storage does not make sense in EPrints; rather, EPrints, can perhaps participate in the front-end interface of the OAIS model.
	Fedora: Fedora supports both FOXML and its internal variant of METS; however, extensibility is limited by the schemas. As mentioned above, DSpace and Fedora are file format-agnostic.
	Greenstone: Greenstone does not have an AIP in the strict definition of the term. However, it does have good documentation of its underlying structures. It also fairly easy to support additional data formats and metadata schemes through the use of plug-ins.
	B2.2. Repository has a definition of each AIP (or class) that is adequate to fit long-term preservation needs. a. Does the repository software application's implementation of an AIP support the use of data formats, data structures, and storage models that are amenable to long-term preservation needs? b. Does the repository software application's implementation of an AIP support the use of preservation metadata elements? For example, elements from the PREMIS schema. c. Can the repository software application's AIP be easily customized to support long-term preservation? E.g., the addition of new metadata elements.

Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
DSpace	The format-agnostic nature of DSpace places the responsibility of policing the addition of preservable data formats on the implementers/administrators. The DSpace data structure is very generic and would not enable one to reconstruct any relationship information outside of DSpace itself. PREMIS information is created during an export process; but this does not exist as PREMIS in the database, rather it is derived from other database elements upon export. The architecture itself is not modeled after the OAIS model, but a move is being made to incorporate concepts from the model.
EPrints	EPrints does not have default PREMIS support. It can, though, be configured to store preservation metadata that's amenable to long-term preservation corresponding to PREMIS' core preservation metadata elements.
Fedora	Fedora Digital Objects are defined in the XML schema language FOXML (Fedora Object XML). The basic components of a Fedora Digital Object are PID, Object Properties, Datastreams, and Disseminators. The PID provides a persistent, unique identifier for the object. Object Properties include Dublin Core metadata, an audit trail of all changes made to the object, and a special datastream that stores object-to-object relationship metadata. The Datastream component of a Fedora Digital Object is the actual MIME-typed content, which can be data or metadata, and can be stored internally or referenced by a URL. Disseminators provide extensible views of the object or constituent datastreams through external services. Since all essential information about on object's contents is encapsulated within the object itself, Fedora Digital Objects can exist independent of the repository. The current Fedora release (version 2.2.1) does not make use of the PREMIS schema for preservation metadata; however, an event-driven preservation and messaging service architecture is in the works for future releases.
Greenstone	Greenstone has no well-defined AIP. Metadata is contained in Greenstone-specific files, which are not interoperable with other repository systems. However, these files do make it easy to backup or migrate collections between different instances of Greenstone on different machines.
B2.3. Repository has a definition of how AIPs are derived from SIPs. a. Does the repository software have a well documented process by which a SIP is ingested into the repository for storage? For example, how and when does the repository software generate fixity data, such as checksums? b. Does the repository software generate or store additional technical metadata derived from the SIPs, such as by using JHOVE?	
Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
DSpace	There is really no distinction between SIP and AIP, other than where these items are in the workflow process. This derivation is simply the indication by an administrator that the item is accepted. Checksums are created upon file upload (either through the web-interface or batch import). Minimal provenance metadata is inserted into the database about when or by whom an item was submitted/accepted but to inference is made about the content beyond attempting to associate a file extension with a file type.
EPrints	As stated previously, EPrints does not process (or infer) content of file type. Like DSpace and Fedora, EPrints can ingest JHOVE metadata as a separate file in the same submission package as the content to which it refers.

	Fedora	The Fedora Directory Ingest service accepts SIPs in the form of .zip archives containing directories of files along with a FOXML (or METS) manifest file that describes the semantic relationship of datastreams in the directory hierarchy. The service processes the archive and creates a Fedora Digital Object in the repository for every file and every directory. Each datastream is processed by a checksumming algorithm, which computes and returns a digital signature for the content of the datastream. The computed datastream checksums are stored in the internal FOXML for the digital object. The relationships among the Digital Objects (e.g., collection-member, folder-document) are recorded and stored in Fedora's RDF-based relationships datastream. No additional generation of technical metadata is performed by the ingest service for an out-of-the-box Fedora installation; although additional services such as these may be integrated via the Fedora Service Framework.
	Greenstone	Greenstone does not have a well-defined process by which AIPs are derived from SIPs. It does not do fixity or checks, nor does it generate technical metadata about files beyond basic information like size and format type. However, via the use of plug-ins and defined policies and procedures an implementer could produce a well-defined derivation path.
B2.5 Repository has and uses a naming convention that generates visible, persistent, unique identifiers for all archived objects (i.e., AIPs). ***Note: Here, both old B2.4 and B2.5 map to new [TRAC] B2.5.***	**B2.4.** Repository has and uses a naming convention that can be shown to generate visible, unique identifiers for all AIPs. a. How does the repository software identify or name AIPs? Are the identifiers guaranteed to be unique within the repository, globally unique, including across time? b. How does the repository software itself uniquely identify an object — URIs, URLs, URNs, etc? c. If an AIP is a discrete entity with storage and relational mechanisms, how do you get to the Entity?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	DSpace supports CNRI handles, which ensure global uniqueness. It is possible (and encouraged) to dereference a package by its submission package identifier.
	EPrints	EPrints enforces uniqueness at the site level via an EPrints Id. Global uniqueness is the responsibility of the administrator. While it's possible, for instance, to ingest metadata having a URI handle into EPrints, EPrints does not check or enforce the URI handle's uniqueness. It's also possible (and likely!) to de-reference and fetch actual package content via the EPrints Id.
	Fedora	Persistent identifiers for Fedora Digital Objects (PIDs) may be user-defined or automatically assigned by a repository. PIDs are case-sensitive and consist of a namespace prefix and a simple string identifier (e.g., demo:1). Every Fedora object has an implicit URI associated with it. The URI for a Fedora object is constructed simply by appending the PID to the string "info:fedora/" (e.g., info:fedora/demo:1). Although Fedora enforces uniqueness at site scope, global uniqueness is the responsibility of the site administration.
	Greenstone	Greenstone does generate unique identifiers for its digital objects. However, they are only unique within the context of a particular Greenstone collection on a particular server. Moving the object to a new collection or moving the collection to a new server would make previous identifiers irresolvable.
	B2.5. If unique identifiers are associated with SIPS before ingest, they are preserved in a way that maintains a persistent association with the resultant AIP.	

	a. Does the repository software preserve pre-existing identifiers for submitted packages?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	Alternate ID's can be added into the metadata of the item, however, it does not facilitate resolution to the item via that ID. If an item had a Handle and the prefix of that handle is assigned to the managing institution then the item can retain that handle in the managing institution's DSpace (if this handle has not already been assigned to another item). If the prefix is not owned by the managing institution, the Handle service must be notified so that that ID can be resolved to the new location.
	EPrints	There are no "pre-existing" EPrints Identifiers. At ingest time, EPrints sequentially generates a local EPrints Id. If the SIP ingests metadata signifying a unique, persistent URI handle, Eprints can be configured to store this global, handle id-- just as if EPrints can be setup to store any metadata. Eprints will not treat the global, handle id metadata as anything unique or special.
	Fedora	Fedora PIDs may be user-defined. Note however that Fedora uses a simple sequential algorithm to generate default PIDs, hence it is practically necessary for the site administration to generate PIDs.
	Greenstone	Greenstone does not have a formal concept of SIPs and AIPs and thus there is no association between the two. Presumably using Greenstone's flexible metadata capabilities, an implementer could adopt processes that would preserve any previously assigned identifiers for a particular digital object, but this would be dependent on local policies and procedures plus customized metadata formats. In addition, some reviewers have reported that after using the normal Greenstone ingest processes many files have lost their original file names.[i]
B2.12	**B2.7.** Repository provides an independent mechanism for audit of the integrity of the repository collection/content. b. Can the repository software application provide a means to validate technical metadata using integrity measures such as checksums? c. Does the repository software provide audit logs of all events that have occurred in the life cycle of a package? d. How does the repository software indicate when a component of an AIP has been corrupted?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	DSpace has an automated checksum auditing tool, the DSpace Checksum Checker. This tool sends an e-mail to administrators where there are possible problems or conflicts. This does not validate any further technical details of the bitstreams.
	EPrints	EPrints does not automatically *validate* checksums for content. It does, though, automatically *generate* checksums so that an administrator can compare checksums at any time. The administrator can also manually run a program to generate checksums. Note that this checksum features can easily be made to run automatically via the operating system though a comparison tool would have to be written for the scripts to be useful. These manually generated checksums can verify that the package files have not

		been changed (or have changed!).
	Fedora	The Fedora repository can compute a checksum for each datastream of a digital object, and later use this checksum to conclusively determine whether the contents of the datastream have been changed. Changes made to datastreams within Fedora via the Management Service are tracked through content versioning and through the audit trail records contained in every Fedora Digital Object. Any modification of a datastream will trigger the creation and storage a new version of that datastream within the repository.
	Greenstone	Greenstone does not have a means to do this.
B3. Preservation Planning	**B3.** Preservation planning, migration, & other strategies	
	B3.1. Repository has documented preservation strategies. a. Does the repository software application document basic procedures that can support preservation such as back-up/restore, data integrity checking, etc.	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	The DSpace Documentation specifies what the repository is capable of.
	EPrints	EPrints is more concerned with self-archiving and access. Long-term preservation is a "misplaced" concern. Here is a question and the beginning of an answer form the EPrints website (http://www.eprints.org/openaccess/self-faq/#1.Preservation):
		"I worry about self-archiving because archived EPrints may not continue to exist or to be accessible in perpetuum on-line, the way they were on-paper."
		This worry is misplaced. It is not really a worry about self-archiving at all, but about the online medium itself. As such, it needs to be directed toward the primary database in question, which is the toll-access refereed journal literature, currently in the hands of publishers and libraries, and most of it already in both paper and digital form. That is the official version of record. If you are worried about the preservation of the online version, it is to its publishers and subscribing/licensing librarians that your worry needs to be addressed. The preprints and postprints that are being self-archived by their authors in their institutional EPrints archives today are intended to maximize impact by providing immediate open access; they are merely open-access *supplements* to that toll-based primary literature at this time, not *substitutes* for it.
	Fedora	Plenty of documentation is available at http://www.fedora.info/.
	Greenstone	Greenstone does not have any explicitly documented preservation strategies, but it does support some functionality that is amenable to different strategies. One of these is the ability of its plug-ins to convert files common file formats, such as Word, PDF, or Postscript to HTML and also keep the original files. Greenstone also makes it fairly easy to backup and restore collections using either the Greenstone Archive Format (GAF) or METS.
	B3.2. Repository implements/responds to strategies for AIP storage and migration. a. Does the repository software support migration of archival packages? b. Can AIPs be exported from the repository in a standard format such as METS or MPEG-	

	21 DIDL? c. How is an archival package backed up such that all relevant information is preserved and can be easily restored into the same repository software system or a different one? d. Is there a clear policy regarding software upgrades? e. Are older packages guaranteed to be forward compatible, and, if not, do new versions of the software have a clear upgrade mechanism for older packages? f. How many generations removed can a package be from the version of the software that created it before it is no longer supported?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	The DSpace Item Exporter exports items as-is with the addition of extra repository metadata in the DC XML file and a handle file. The DSpace METS Exporter converts the DC metadata to METS and retains the obfuscated file names used in the datastore. The METS Exporter package is somewhat cumbersome for migration to another repository. In fact, the developers seemed to prefer the native DSpace format for migration since a migration script is included with a default DSpace installation. The migration script removes the extraneous metadata and optionally purges handle files.
	EPrints	Concerning metadata, EPrints exports as an EPrints record and it can also export as an OAI-compatible DC record. Essentially, it allows you flexibility in what metadata it will export. The content it exports (once configured) is not changed. So, if you need to transform the content (and metadata) to another format, you would likely need to apply these transformations outside of EPrints— if EPrints is not already configured for a particular export. While OAI-DC metadata is certainly compatible (and standardized) for migration to other systems, it is somewhat more limited in metadata than the EPrints metadata. EPrints does not have a clear, upgrade policy on its website although they do provide upgrade instructions for all previous versions.
	Fedora	Fedora exports metadata both in FOXML and METS formats. FOXML is better supported and much more amenable to migration. It is possible to export any Fedora object and then subsequently ingest that object into a different Fedora repository when the original repository is no longer accessible, or no longer running. Major software upgrades may require administrators to re-ingest Digital Objects via a mass export and ingest, but simple upgrades may be performed without a re-ingest. The power behind Fedora's Digital Object implementation is that the repository metadata SQL database and RDF triplestore can be rebuilt entirely from the object-level metadata streams that make up every Fedora Digital Object.
	Greenstone	Greenstone supports the import and export of METS files. Migration between servers can also be accomplished using its Greenstone Archive Format (GAF). The Greenstone project has gone to pains to ensure that collections are compatible between versions of the software or that there is a clear upgrade path. However, they offer no guarantees.
	B3.3. Repository uses appropriate international Representation Information (including format) registries a. What standards does the repository software use to describe file formats, such as Internet MIME Types? b. Does the software support any type of format registry such as PRONOM or Global Digital Format Registry (GDFR)?	

Notes on this Checklist Item from Repository Testing Evaluations in 2007	
DSpace	DSpace identifies file types by file extension and associates file extensions with mime types.
EPrints	You must assign files to format types configured in the EPrints system. EPrints does not enforce files be compliant with certain MIME types. It is setup to be an OAI-compliant archive.
Fedora	Fedora examines files to check them against a set of MIME types, but does not enforce MIME-type conformance. The Format URI attribute of Fedora Digital Objects allows for an optional format identifier for the datastream, such as those found in PRONOM and the Global Digital Format Registry (GDRF).
Greenstone	Greenstone only keeps track of file formats based on their file extensions or possibly through customized metadata. It does not support any format registries.

B3.8. Repository has contemporaneous records of actions taken associated with ingest and archival storage processes and those administration processes which are relevant to preservation.

a. Does the repository software provide audit logs of events significant to preservation that have occurred in the life cycle of a package? If so, then:
   1) What events are logged?
   2) What format are the logs in? Are they easily processed by human or machine?
   3) Are the logs tamper resistant and can they be kept 'forever'?

Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
DSpace	Minimal provenance metadata is inserted into the database about when or by whom an item was submitted/accepted. This is in a free text field so it is not easily acted upon programmatically. The main events are logged to the main DSpace log. The log file using the default settings will log authentication attempts, changes made to content, and various other actions taken listing the session ID and the e-person who performed these actions.
EPrints	There are no audit logs of events that EPrints produces. (Our scripts produce the logs). As for user access logs to EPrints, there are still no logs. However, there's PERL software available (LogFile::EPrints) that analyzes the web server log, thus providing the ability to log EPrints access.
Fedora	Any change made to a Fedora Digital Object results in a new version of the altered datastream; the original datastream is kept intact. When a datastream is versioned, the object's audit trail is updated to reflect the changes made to the object, when the change was made and by whom, and a new version of the modified datastream is added to the object. All versions of the datastream are numbered to show the relationship between the original and new versions. Audit logs are part of the object itself.
Greenstone	Greenstone does have some logging capabilities. Web logging can be enabled to track end user requests for objects in addition to web server errors. Other data associated with the Greenstone Librarian interface can also be enabled, such as warnings and errors. However, these log entries are primarily concerned with the technical operation of the Greenstone application and not with actions taken against collections or digital objects which have any bearing on preservation. The logs are not tamper resistant, and their persistence is entirely a function of the

		policies and procedures of the particular Greenstone installation, and not of the Greenstone software.
B3.2	**B3.9.** Repository has mechanisms in place for monitoring and notification when Representation Information (including formats) approaches obsolescence or is no longer viable. a. Can the repository software application monitor any standard format registries in order to ascertain format obsolescence? b. Can the repository software application support scheduled events, such that a human operator can be notified on a preset schedule to check manually for format obsolescence?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	We are unaware of format-obsolescence checks in DSpace.
	EPrints	We are unaware of format-obsolescence checks in EPrints.
	Fedora	We are unaware of format-obsolescence checks in Fedora.
	Greenstone	Greenstone does not have a means to do this. Although, it does have plug-ins that can convert currently common formats like MS Word or PDF into potentially more long-lived formats such as HTML, but the application of these plug-ins for obsolescence remediation is a matter of policy and procedures and not of the software.
B3.3	**B3.10.** Repository has mechanisms to change its preservation plans as a result of its monitoring activities. *The following are examples of how this may be applied to repository software applications:* a. Does the repository software application support the migration of metadata formats or digital object formats? b. Does the repository software application easily support the export of its data? **** *See also B3.2.*	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	Metadata is configurable, and there are cross-walking capabilities. Format conversion is possible with an optional plug-in via the OpenOffice.org API. Neither of these is explicitly tied to the monitoring activities.
	EPrints	EPrints is generally flexible enough to change. Please see discussions above. Its migration seems to take into account (e.g., it supports) data migration to an upgraded version of EPrints.
	Fedora	Please see above discussion of submission package migration.
	Greenstone	Greenstone does not have a means to do this.
B5. Information Management	**B4.** Data Management	
B5.2	**B4.1.** Repository captures or creates minimum descriptive metadata and ensures it is associated with the AIP.	

	a. Can the repository software application infer or derive descriptive metadata from its digital objects? b. How does the repository software application associate descriptive metadata with the AIP? For example, is the descriptive metadata encapsulated within the AIP, or is it referenced from the AIP? Does this create any concern that associations between the descriptive metadata and the AIP may be vulnerable?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	The web interface enforces that the minimum metadata elements are associated with the item submitted.
	EPrints	EPrints does not infer or derive descriptive metadata from its digital objects. Actually, the only metadata element that it derives is an update date metadata element, but this derivation has more to do with actions upon the object rather than the digital object itself. The web interface enforces that the minimum metadata elements are associated with the item submitted, requiring users to fill in fields before continuing the submission & editing processes. The batch submission process, though, seems to be able to bypass at least some of the required fields.
	Fedora	Descriptive metadata is encapsulated within the Fedora Digital Object.
	Greenstone	Greenstone does have some capability to do this via its plug-ins which can extract text and in some cases basic metadata from some common file formats.
B5.3	**B4.2.** Repository can demonstrate that referential integrity is created between all AIPs and associated descriptive information. a. How does the repository software application maintain links between the descriptive metadata and the Digital Objects? b. How are the links maintained for other types of metadata?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	DSpace assigns handles to objects including the community, collection, item, and bitstream. Relationships between these are maintained in the database. Pointers to the bitstream's location on the filesystem are kept in the database as well.
	EPrints	EPrints stores all data & metadata within its *asset-store*, a designated filesystem area. The file names within this asset store match up with an internal EPrints Id. For example, EPrints Id=123456 would match to the directory, "disk0/00/12/34/56". The EPrints Id is stored in the EPrints database.
	Fedora	Fedora datastreams are identified by FILE elements in FOXML. Each datastream is associated with an ID and URL. The datastream IDs are typically sequential, but this is specified in the metadata. Datastream IDs and URLs must be unique within a package.
	Greenstone	Metadata about a collection and its components are stored in the Greenstone Archival Format (GAF) file which references digital files stored on the disk. Greenstone does not have any technical means to ensure the integrity of these links.
B6	**B5.** Access Management	

B6.5	**B5.1.** Repository access management system fully implements access policy. a. How does the repository software application provide access to the Content Information? b. How flexible or customizable is the repository software application with regard to access? c. Can the repository software application be set up to allow or deny access based on access rules, such as IP address restrictions, user id and password, or other means?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	DSpace is packaged with a customizable web-based interface for community and collection creation, eperson management, and all aspects of item submission, modification, and removal. There is also a suite of command-line tools for batch processing and automation.
	EPrints	EPrints is packaged with a customizable web-based interface for account management and all aspects of item submission, modification, and removal. There is also a suite of command-line scripts for batch processing and automation.
	Fedora	The Fedora Access Service (API-A) is a SOAP-enabled web service for accessing Digital Objects stored in the repository. It includes operations necessary for clients to perform disseminations on objects in the repository and to discover information about an object using object reflection. Access is controlled by fine-grained, machine-readable security policies in XACML.
	Greenstone	Greenstone currently relies on the access restriction mechanisms of the underlying file system and web server. This allows fairly granular control based on IP addresses or user ids. However, these will need to be maintained outside of the Greenstone software and is not scalable for complex access restriction requirements.
B6.6	**B5.2.** Repository logs all access management failures, and staff review inappropriate "access denial" incidents. a. Does the repository software application log all access attempts, both successful and unsuccessful? b. Are access denials flagged in any special manner by the software? c. How accessible are the logs to either human- or machine-processing and interpretation?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	DSpace logs authentication attempts and failures. There are no built in mechanisms that allow notification for suspicious activity.
	EPrints	I'm not aware of any log facilities of users. EPrints does list the number of items in the repository associated with a particular user, not mentioning how many attempts it took to get to that number. I suspect the apache logs would generate any erroneous logon attempts and will perhaps indirectly (via URLs in logs) tell you other user log information.
	Fedora	Fedora uses Log4J for logging. Reports of inappropriate "access denial" incidents are sent to the server logs and/or the security logs, depending on configuration.

	Greenstone	Greenstone can be set up to log all access to a collection both successes and failures. This is done by through the regular web server logs. However, review or notifications based on those logs is outside the scope of the Greenstone software.
B6.7. Repository can demonstrate that the process that generates the requested digital object(s) (i.e., DIP) is completed in relation to the request.	**B5.3.** Repository can demonstrate that the process that generates the DIP is completed in relation to the request. Can the repository software application communicate to the user the status of their request, including indicating whether the process that generates the DIP is complete or incomplete in relation to the request? a. Does repository software application ever deliberately return a partial response to a request and if so how does it notify the user? b. How does repository software notify user when it is unable to respond to a request?	
B6.8. Repository can demonstrate that the process that generates the requested digital object(s) (i.e., DIP) is correct in relation to the request.	**B5.4.** Repository can demonstrate that the process that generates the DIP is correct in relation to the request. Does the recipient of the DIP have any means at their disposal to verify the correctness of a DIP? E.g., checksums to verify the integrity of the bitstreams received; schema that could be used to programmatically validate the correctness of a DIP.	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	In the web interface, only the constituent bitstreams are available for download. The checksums of these bitstreams are not exposed to the users, so they have no way to determine the validity of their downloads. Because this works over HTTP, unavailability may be indicated by appropriate HTTTP Status Codes.
	EPrints	EPrints has no way automatically to verify the contents and correctness of a DIP. Only through the web interface via the staging process can Eprints allow an editor to verify the correctness of a DIP. Even there, the editor does not have instant access to checksums.
	Fedora	Fedora Digital Objects may be exported in FOXML or Fedora Mets through the Management Service (API-M) by a repository administrator. To verify that the content of the datastreams have not been damaged or corrupted, the "compareDatastreamChecksum" utility may be invoked from the Fedora Management Service. It computes a new checksum using the same checksum algorithm as the original and compares it to the checksum stored for the specified datastream.
	Greenstone	Greenstone does not have any means to technically ensure that dissemination is correct.
B6.9	**B5.5.** Repository demonstrates that all access requests result in a response of acceptance or rejection. a. How does the repository software application notify the end-user that an access request is accepted or rejected? b. Does the repository software application log all access requests?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	

	DSpace	Because this works over HTTP, availability may be indicated by appropriate HTTTP Status Codes. Otherwise, this notification will be done by the client used to download the bitstream.
	EPrints	EPrints does not log access requests. EPrints can be programmed so that each time a user wants to ingest a record into the repository, the administrator/editor will be notified via email. The editor can then look at the EPrints record. This notification occurs in the web interface. It can just as easily be turned off.
	Fedora	The only access to the actual data objects in Fedora is through the Management Service API-M (restricted to administrators). All other access is limited to datastream disseminations through the Access Service API-A. Since these are web services, any request would typically trigger an HTTP status code.
	Greenstone	Greenstone does not have any means to do this, but all access requests can be logged via the web server log.
B6.10	**B5.6.** Repository enables the dissemination of authentic copies of the original or objects traceable to originals. a. Does the repository software application support validation of the original document — for example, by using checksums, digital signatures, file comparisons, etc? b. Does the repository software application support provenance metadata, and make it accessible in some manner to users? c. Does the DIP contain provenance information for the digital object — e.g., does it track file transformations? *** *See ref to B5.3.*	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	In the web interface, only the constituent bitstreams are available for download and are pulled directly from the asset store. The checksums of these bitstreams are not exposed to the users, so they have no way to determine the validity of their downloads. There is no DIP specifically, but administrative export of an object does track minimal provenance information.
	EPrints	The administrator can add the checksums to the metadata for the EPrints record so presumably EPrints can be configured to display this checksum to the requester and presumably the requester can compare this checksum to the original checksum— of which the requester should have access. The checksums are as tamper-proof as the operating system on which EPrints is running. EPrints does support *existing* provenance metadata—loaded with the content if necessary, but it does *not create* new provenance metadata except for the checksum and the "Deposited On" date. I'm not aware of any versioning support mechanism.
	Fedora	Disseminations of constituent datastreams can be requested via the Access Service API-A. They are not validated, nor are any provenance metadata provided. Repository managers may obtain access to the actual data through public exports via the Management Service API-M. Exported DIPs contain provenance information for the digital object, datastream versions, and an audit trail.
	Greenstone	Greenstone can allow access to original and derived copies of a digital object, but it does not maintain verifiable provenance information that can be used to ensure which is the

		original and which is the derivative.
In TRAC, Section C is now "Technologies, Technical Infrastructure, and Security." What was in Section C in the TDR Audit Checklist (right-hand column) now has been integrated into the rest of TRAC.	**C.** Designated Community and the Usability of Information **C3.** Use and Usability	
	C3.2. Repository has implemented a policy for recording all access actions (includes requests, orders, etc.) that meet the requirements of the repository and information Producers/depositors. a. Does the repository software have an event log? What events are logged?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	The main events are logged to the main DSpace log. The log file using the default settings will log authentication attempts, changes made to content, and various other actions taken listing the session ID and the e-person who performed these actions.
	EPrints	See above discussions.
	Fedora	Fedora uses Log4J, which supports customizable log levels. Log4J has many configuration options; for more information, see the Log4J manual: http://logging.apache.org/log4j/docs/manual.html.
	Greenstone	Greenstone does logging. In Greenstone error logs record internal errors and user logs record usage. User activity logs record all queries made to every Greenstone collection (though this facility can be disabled).
	C3.3. Repository ensures that agreements applicable to access conditions are adhered to. b. What sort of access restrictions and authentication mechanisms can the repository software application support? c. How does the repository software application document and implement access restrictions?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	It is up to the repository administrators to set up the access conditions of the e-people and groups.
	EPrints	It is up to EPrints Administrators to set up user access controls. See the above discussions for more information.
	Fedora	Fedora uses the XACML policy language to describe general access control requirements. XACML allows queries to be written that ask whether or not a given action should be

		allowed, and responses to be interpreted. The XACML standard is developed and maintained by the OASIS Consortium: http://www.oasis-open.org/.
	Greenstone	Greenstone currently restricts collection building. Greenstone also enables a specified user (or users) to authorize others to build collections and add new material to existing ones. Greenstone, through htaccess' password protection, can restrict access of digital objects.
	D. Technologies & Technical Infrastructure	
C1	**D1.** System infrastructure	
C1.1	**D1.1.** Repository functions on well-supported operating systems and other core infrastructural software. a. Are the systems and services required by the repository software application still current and supported by the developer or manufacturer? (For example, these might include the operating system, web servers, database servers, and particular computer language run-time environments.) 1) A related consideration: If a key software component fails, can it be replaced? For example, if there is a security hole in the database, are there patches available? Or can the software component be switched for a different database application?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	DSpace runs on both Linux and Microsoft operating systems, it utilizes PostgreSQL or Oracle as the database system, it is known to run on the latest version of Java (1.6.2 as of 9/17/07).
	EPrints	EPrints is written in PERL and runs on Linux (and, I assume, Linux-compatible systems). It uses Apache (with mod perl) as the web server. Its backend database is MySQL. The software prerequisites are here. I believe the hardware requirements should be determined mostly by the application (e.g., if you are storing a huge repository, you need a more disk space than if you were storing a small repository).
	Fedora	Fedora runs on Windows or Linux. It is designed to be RDBMS-independent, but it comes with a java database called McKoi. Fedora also supports MySQL, Oracle 9 and PostgreSQL databases, but they must be installed separately. Fedora requires a servlet container to run, and the installer includes Tomcat 5.0.28. Later versions of Tomcat have been shown to work as well. For its triplestore, Fedora uses the Kowari metadata store.
	Greenstone	Greenstone has been available and in development for nearly a decade and runs on many common platforms. It is open source and most of the underlying components upon which it relies are also open source. However, some of the underlying components are showing their age, such as the Perl programming language. However, Greenstone version 3 which was not part of our official review is likely to remedy this situation.
C1.2 Repository ensures that it has adequate hardware and software support for backup functionality sufficient for the repository's services and for the	**D1.2.** Repository ensures that all platforms have a backup function, sufficient for the repository's services and for the data held, e.g., metadata associated with access controls, repository main content, etc. a. What sorts of backup strategies does the architecture of the repository software application afford?	

data held, e.g., metadata associated with access controls, repository main content.	b. Does the repository software application explicitly require any particular backup strategy, or does it just rely on system-level backup plans, like periodic disk backups to tape? c. In the event of a disaster, how would the repository software application be restored?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	No particular backup strategy is required. Restoration would include replacing the asset-store and rolling back the database to the point in time current with the state of that asset-store.
	EPrints	EPrints does not have any built-in back up tools. The backup strategy would involve system level strategies, saving the asset store and backing up the database tables.
	Fedora	Since Fedora stores its data in a simple file system; it requires no extraordinary backup strategies. In the event of a disaster, everything can be restored from a simple file-system backup.
	Greenstone	Greenstone affords a couple different backup strategies. Although, the actually implementation of those strategies are outside actual Greenstone system itself. The first strategy relies on normal system-level backup and restore procedures. The Greenstone Archival Format file and all associated files can be backed up from disk and restored as usual. Alternately, METS files can be exported from Greenstone and then backed up along with the associated files. The only advantage of METS is that it would potentially be more portable to other systems than the native GAF format.
C1.3. Repository manages the number and location of copies of all digital objects.	**D1.3.** Repository stipulates the number and location of copies of all digital objects. For a basic, non-distributed repository systems this should be part of basic functionality, but for a more complex distributed architectures the management of multiple copies of digital object may be a concern, for example you should consider the following questions: a. Does the repository software application have any explicit support for parallel operations, for example multiple duplicate copies of the service running on different servers, to support data redundancy in the event of a disaster, or for load balancing in high usage scenarios? b. Does the repository software have any explicit support for distributed operations such that different parts of the system are running on different servers, either in terms of data or in terms of functionality? For example, packages belonging to collection A are stored on server X, but packages belonging to collection B are stored on server Y, or metadata are stored on server X, but Digital Objects are stored on server Y. *****See also D1.4.*	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	You can specify that the asset store live on a network (mounted) volume, but because the files names are hashed, there is no way to specify that certain items should live in specific locations.
	EPrints	I'm not aware of any explicit parallel-processing support for EPrints. I believe a redundancy operation could be implemented at the hardware level. I'm also not aware of any explicit support for distributive operations. Though, EPrints can easily be distributive via the operating system, which can, for instance, store collection A on machine X and collection B on machine Y while running EPrints on machine Z.

	Fedora	The Fedora Digital Object allows for constituent datastreams to be stored offsite, in which case the object holds a pointer to the content in the form of a URL. Disseminations may be performed on these datastreams, as well as management operations, as if they were local to the repository.
	Greenstone	Greenstone does not support distributed or parallel storage beyond what is provided natively by the underlying file system. In general only a single copy of any given file is stored unless there are procedures external to Greenstone to ensure that multiple copies are stored.
C1.4	**D1.4.** Repository has mechanisms in place to insure any/multiple copies of digital objects are synchronized. Consider: a. In cases of distributed or parallel systems, how is synchronization of the data ensured? What sorts of latencies are involved when objects are changed? b. How does the repository software handle duplicates? For example, what happens if the same submission package is submitted multiple times? *****See D1.3 above.*	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	DSpace allows duplicate item packages within a collection to be archived multiple times, generating new handles for each duplicate, but there is no synchronization of these items.
	EPrints	Eprints allows duplicate item packages within a collection to be archived multiple times, generating new EPrints Ids for each duplicate.
	Fedora	If the same submission package were submitted multiple times, the repository would treat it as a new submission, and give it a unique PID.
	Greenstone	Greenstone does not have means to do this.
C1.5	**D1.5.** Repository has effective mechanisms to detect data corruption or loss. a. What techniques are used to detect data corruption or loss? Possible techniques include (from less to more effective) comparison of files sizes, use of digest algorithms, comparisons of secondary copies of files, and digitally signed files. b. Related to detection is the issue of repair. Can the repository offer any means to correct corrupt files? Certain architectures may be more amenable to file repair than others, for example architectures where multiple copies of files are distributed. *****See D1.6*	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	DSpace has an automated checksum auditing tool, the DSpace Checksum Checker. DSpace associates MD5 checksums with every file in its asset store, and periodically checks that these have not changed. This tool sends an e-mail to administrators where there are possible problems or conflicts.
	EPrints	I'm unaware of any technique Eprints uses to detect data corruption or loss.

	Fedora	Whenever content is added or modified via Fedora management functions, a checksum is computed for the content. The computed datastream checksums are stored within each digital object. Fedora provides no apparent means for correcting corrupt datastreams. However, any change to a datastream made through the Management Service will result in a new version of that datastream. If subsequent versions are found to be corrupted, the earlier versions may still be intact.
	Greenstone	Greenstone does not have means to do this.
C1.6	**D1.6.** Repository reports to its administration all incidents of data corruption or loss, and steps taken to repair/replace corrupt or lost data. a. Does the repository software application log events that occur on a package, specifically the detection of data corruption or loss? If so, how are these errors logged? b. Can administrators be automatically notified of these types of events, for example by email or RSS feeds? *****See D1.6*	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	DSpace has an automated checksum auditing tool, the DSpace Checksum Checker. DSpace associates MD5 checksums with every file in its asset store, and periodically checks that these have not changed. This tool sends an e-mail to administrators where there are possible problems or conflicts.
	EPrints	Eprints has no specific tool concerning data corruption or loss.
	Fedora	Neither automatic data-checking nor automatic administrator notification mechanisms are included in a standard Fedora installation (as of version 2.2.1). Although, an event-driven preservation and messaging service architecture is in the works for future releases.
	Greenstone	Greenstone does not have means to do this.
C1.7	**D1.7.** Repository has defined processes for storage media migration. How can the repository software application support migration? For example, you may want to consider the following: a. Does the repository software support migration of archival packages? Can they be exported from the repository in a standard format such as METS or MPEG-21 DIDL? How is an archival package backed up such that all relevant information is preserved and can be easily restored into the same repository software system or a different one? b. Can the repository software ingest its own dissemination packages without data loss? c. Does the repository software documentation provide recommendations for how to execute storage media migration—such as dumping data to a temporary storage space while swapping out disks?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	It is possible to copy the asset store to a new location and simply change the asset store directory configuration in DSpace to access this store in the new location. The DSpace Item Exporter can disseminate packages in bulk in a format that can be immediately resubmitted. There is no data loss other than possible down-casting of metadata from more descriptive formats to DC metadata, which is an issue of original submission rather

		than re-submission.
	EPrints	See above discussion for EPrints data migration. The repository does have bulk imports and exports. These scripts are rudimentary and do not have logging. We created scripts using these rudimentary scripts but adding logging and more advanced features (for our Hub and Spoke purposes).
	Fedora	Fedora Digital Objects can be exported in either FOXML or METS. They may be exported as fully self-contained archive packages, migrated to another repository, or re-ingested into the same repository. The export and migration processes are fully documented at http://www.fedora.info/.
	Greenstone	Greenstone has plug-ins for migrating from common file formats to HTML or plain text. Greenstone can import and export METS files which are conformant to its own profile. In addition, the GAF files and associated data files can be moved between instances of Greenstone relatively easily. Although, search indexes will need to be regenerated.
C1.9	**D1.9.** Repository has a process for testing the effect of critical changes to the system. This is primarily an organizational issue. Consideration related to the repository software application might include: a. How does the group or development community responsible for the repository software application test new releases, patches, fixes, etc., that are relevant to the core system as it's generally distributed?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	Tests of patches should be done in an environment that closely resembles the environment of the production installation.
	EPrints	EPrints has no built-in ways to test critical changes. An appropriate way for testing any change is to set up a test environment, which replicates the production environment's software in every way except for the changes you are testing.
	Fedora	Since Fedora Digital Objects are self-containing, they may be backed-up and safely stored away from the repository while system administrators test new releases, or apply patches and fixes to the repository core.
	Greenstone	Greenstone, as a long-lived open source project, presumably has such processes in place.
C1.10. Repository has a process to react to the availability of new software security updates based on a risk-benefit assessment.	**D1.10.** Repository has a process to stay current with the latest operating system security fixes. *This is primarily an organizational issue. Consideration related to the repository software application might include:* a. Does the group or development community responsible for the repository software have a process for dealing with security issues either related to the operating systems or any dependent software that may affect the repository software? b. Does the repository software application documentation report what operating systems and dependent software it requires?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	There are recommendations of recommended environments in the documentation, but many developers and implementers have noted success on other platforms.

	EPrints	See above discussion of software upgrades.
	Fedora	The Fedora Repository System is built on open-system standards and is highly modular, which allows for future evolution through component replacement. The code is written in Java, runs under the Java Virtual Machine, and is operating-system independent.
	Greenstone	Greenstone is well documented and under active development, including a major upgrade to better align itself with current technologies and trends.
C.2	**D2.** Appropriate technologies	
C2.1	**D2.1.** Repository has hardware technologies appropriate to the services it provides to its designated communities and has procedures in place to monitor and receive notifications when hardware technology changes are needed. *This is primarily an organizational issue. Consideration related to the repository software application might include:* a. What are the hardware requirements for the repository software application? b. How are implementers and managers of the repository software notified when new repository software application versions are available that may have different hardware requirements?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	The installation guide outlines the requirements for DSpace. If these requirements change between releases, then the changes should be reflected in the installation guide for that specific release.
	EPrints	The hardware specifications, and any changes in a new release, for EPrints will be reflected on the Eprints site.
	Fedora	There are no explicit hardware requirements for running Fedora, although the computer must be able to run the Java SE Development Kit (JDK) 5.0, and have sufficient storage space.
	Greenstone	Greenstone was designed for use in developing countries and supports hardware appropriate to that use. It is also cross-platform and runs on most major operating systems. Based on their long history, the Greenstone developers seem to be aware of changes in operating systems and hardware and have kept their software up-to-date accordingly.
C2.2	**D2.2.** Repository has software technologies appropriate to the services it provides to its designated communities and has procedures in place to monitor and receive notifications when software technology changes are needed. *This is not primarily a repository software application issue. However, considerations related to the repository software application might include:* a. Is the repository software application amenable to modification if needed? For example, do the repository software application developers have a mechanism for receiving feedback from implementers requesting bug fixes and additional functionality–such as surveys, focus groups, etc.? b. How are implementers and managers of the repository software notified when new repository software application versions are available that may have different software	

	requirements?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	The DSpace project accepts bug reports and feature requests through their project page at SourceForge. If requirements change between releases, then the changes should be reflected in the installation guide for that specific release.
	EPrints	EPrints is open source actively maintained by its creators at the main EPrints Web Site. Any new version would be announced on that web site.
	Fedora	The development community for Fedora have a Bugzilla page (http://www.fedora.info/bugzilla/) to help them effectively keep track of outstanding bugs in their product.
	Greenstone	
C3	**D3.** Security	
C3.2. Repository has implemented controls to adequately address each of the defined security needs.	**D3.2.** Repository has implemented mechanisms (processes) to adequately address each of the defined security needs. a. Does the repository software application support different levels of access for different types of users or different types of objects? How does it provide security? b. Does the repository software application afford the appropriate levels of security required for your user community?	
	Notes on this Checklist Item from Repository Testing and Evaluations in 2007	
	DSpace	The DSpace installation supports access control with community, collection, and bitstream granularity. Access can be restricted to specific groups and users. Embargos or other time based restrictions are not built into the software and must be manually mediated. Technical information on DSpace Authorization is included in the system documentation.
	EPrints	EPrints does initiate some sort of user security. Users can be given different levels of responsibility and are required to log into the public space in order t make changes to the EPrints content. EPrints allows any user to be able to search the system (of course, the operating system and web server can certainly limit those user searches).
	Fedora	Fedora's access controls are specified as XACML policies. Fedora administrators may specify finely-grained, machine-readable policies in XACML to control access to Fedora web services, digital objects, datastreams, and disseminations. Policies can be written to permit or deny access based on attributes of the user, attributes of digital objects, and attributes of the environment. Additionally, Fedora supports standard user-authentication through Tomcat's users file or an LDAP directory.
	Greenstone	Greenstone has three levels of security: administrators, collection builders (librarians), and information seekers. For any granularity beyond this Greenstone relies on the mechanisms of the underlying file system and web server for its security needs.

i Han, Y. (2004). Digital content management: The search for a content management system. *Library Hi Tech*, 22(4). Retrieved November 1, 2007, from Emerald Group Publishing Ltd. database, ISSN 0737-8831.