

RUBRIC Toolkit: Managing a Repository

The Institutional Repository (IR) space is continuing to develop internationally, and it is important for the repository manager to carefully monitor sources such as discussion lists to stay abreast of new developments.

This section covers some of the key issues to be monitored once the IR is in production.

Action plans to assist with management can be found in the Digital Preservation Management section.

Staffing Structure and Staff Development

The Planning section outlined a staff structure for the project phase. Once the IR is in operation, it may be necessary to review the number, level and development needs of staff involved.

IR Managers should plan to review the staffing levels in the first two years of operation and work closely with their staff to ensure they have sufficient training and resources to match their workload (workload is likely to fluctuate during the establishment period).

[Content in Institutional Repositories: A Collection Management Issue¹](#) (Genoni 2004) provides a good list of the basic library staff responsibilities in managing an IR:

- encourage members of the university to deposit material into the IR
- provide advice to members of the university about copyright and journal embargo policies for material which they would like to deposit in the IR
- convert material to a suitable format such as html or pdf for import
- deposit material directly on behalf of members of the university who do not, or cannot self-archive their material
- review the metadata of content which has been self-archived to maintain the quality of the record.

Training in these areas is vitally important to ensure quality control, particularly in the areas of data entry and review.

Software Patches

The IR Manager needs to know:

- the frequency and schedule of software patches
- how to apply them and
- where to seek advice and peer support

Supporting documentation should include:

RUBRIC is supported by the Systemic Infrastructure Initiative as part of the Commonwealth Government's Backing Australia's Ability - An Innovative Action Plan for the Future (<http://backingaus.innovation.gov.au>)

- an upgrade policy
- upgrade checklists
- test scripts

All documentation associated with an upgrade should be reviewed and updated after each upgrade.

An upgrade policy helps staff to manage and document the work involved in minor patch fixes as well as major upgrades of the software.

A checklist for conducting software upgrades should include responses to the following questions:

- do we have a test instance to conduct all pre-upgrade testing of the software?
- how do I find out about software patches and upgrades?
- what assistance is available if we have problems with a patch or upgrade? How responsive is the assistance and do we need to have a risk management strategy in place?
- what are the downtime implications of upgrades?
- will we establish a formal upgrade cycle or schedule?
- who is responsible for implementing a patch or upgrade?
- are there any other staffing implications for the library and other areas such as IT?
- do we need approvals for major upgrades, especially if there are staffing implications in other areas?
- what is the potential impact on users?
- do customisations need to be reapplied? How long will this take?
- will we need to upgrade operating systems or any other associated software with a particular release? How might this add to the overall risk of the upgrade?
- will we develop test scripts?
- how long will we need to complete an upgrade? There will be certain areas where testing and documentation would be useful, such as re-indexing

Test scripts might include testing for the following areas, depending on the functionality in the software implemented:

- workflow – simple tests to check whether the upgrade impacts on any workflows already used by staff to enter or edit items in the IR
- authentication – a test of staff and user authentication
- web services – do you offer any associated web services which need to be reestablished after an upgrade? How would you document and test the process?
- statistics
- customisations – any customisations should be well documented so that they can easily be reapplied after an upgraded and the functionality tested

- harvesting - does it still work?
- indexing - how long does it take? Do you have standard entries that you check after a reindex? How would you test the indexing for a new entry?

Major Software Upgrades

Two types of major upgrades that might occur during the life of your IR include:

- major software upgrades on the selected software
- migration from your existing system to a new system with different software

Major software upgrades will generally occur at least once a year, together with a series of patch fixes which may be applied throughout the year. The checklists provided above will be useful in deciding how to manage the upgrade process.

Migration plans should be made for migrating data to completely different software. This is very important in terms of risk management of both software obsolescence and also data recovery. Migration is best managed by the IR Manager maintaining:

- a development plan, which looks at the organisational picture
- a migration plan which assists with operational transition

A Development Plan checklist might include the following:

- justification for system changeover (key drivers for change: customer issues, operational issues, software issues)
- any formal project plans required by your organisation
- pilots or test phases required
- risk analysis
- associated costs
- resource requirements
- responsibilities for each phase of the migration
- a plan for phasing out the legacy system, including a communication plan and a schedule for system changeover – overlap should be identified, along with support requirements, reasons and length of overlap
- user logistics issues, especially where an organisation has implemented self submission services which may be disrupted
- interface compatibility review
- communication on unavoidable system changes and user implications
- marketing for the new system
- training plan for staff and users on major system differences

A **Checklist for Migration Planning** should include the following:

- pre-upgrade checks on disk space requirements
- review of all items on the usual upgrade checklist (as above)
- progress chart, including milestones and responsibilities
- tracking system for issues reported
- test scripts and focus group plans for measuring migration success

Data Management Issues

Detailed information about managing the data in an IR has been provided in the section on Data Management.

Routine checks should be conducted on an operational repository for ongoing data quality assurance. Checks can include:

- harvesting process. Are documents harvested appropriately by the nominated search engines. Can the documents be located and do the links back to your IR work?
- author authorities. Are author names duplicated in the data entry process?
- data entry review on workflows (focus groups with staff and users may be useful)
- data reporting and export processes (these should be well documented and tested regularly)

Submission Workflow Review and Development

The type of submission process applied to the IR is often decided in the planning stages of the project. Once the system is live it may be useful to review the initial decisions. For example, The University of Southern Queensland (USQ) decided to only offer managed submission initially for the purposes of quality control, usability and staff confidence and training.

There had also been insufficient time to implement the university's standard authentication protocol (LDAP) prior to going live with the system. Once the IR was running successfully, staff competence had been developed and LDAP had been implemented, USQ moved to a self-submission model.

[Populating the Repository](#)² covers information on the various types of submission available which may be appropriate at different times in the IR lifecycle. These include:

- batch ingest
- managed submission
- self submission

Sustainability

Sustainability is an ongoing issue for IR Managers. Objects loaded to an IR will become meaningless over time if they are format dependent or cannot be located by users. The main sustainability concerns are:

- preservation
- software obsolescence
- persistent identifiers
- organisational support and embedding
- disaster recovery

Preservation

Preservation activities may include:

- storage
- replication
- transformation

A preservation statement in the Collection Development policy should define:

- who is responsible for preservation activities
- the extent to which preservation is guaranteed
- the kinds of managed activities which will be undertaken to ensure an item deposited remains accessible on a continuous basis for as long as necessary

Software Obsolescence

There are two aspects of software obsolescence to consider:

- the repository software itself
- the document software at the item level of documents stored in the IR

IR Managers can minimise their risk exposure to obsolescence by ensuring:

- they use up to date repository software
- they understand how to export and migrate data from their current repository

Some discussion on these issues occurred at the APSR event “[Long-term Repositories: Taking the Shock out of the Future](#)³” which reviewed two checklists for IR managers:

- RLG's [Audit Checklist for Certifying Digital Repositories](#)⁴

- [Data Dictionary for Preservation Metadata](#)⁵ (Final Report of the PREMIS Working Group, May 2005)

System Options includes general comments regarding the sustainability of IR system.

The IR Manager should conduct a regular risk assessment on software format obsolescence in relation to the data stored in a repository. Consider whether data objects stored in an IR are so software dependent that in 5, 10 or 20 years time they will be irretrievable because of their format. Concerns should be noted in IR policy documents.

As a control measure, IR Policies may contain rules on the types of software formats initially allowed in the repository.

[AONS II](#)⁶ (Automatic Obsolescence Notification Service) is collaborative software work being undertaken by APSR (Australian Partnership for Sustainable Repositories) and the National Library of Australia. This project aims to “produce an open source, platform-independent, configurable and downloadable tool that automatically provides information from authoritative international registries such as:

- [Library of Congress Digital Formats Web Page](#)⁷
- [PRONOM](#)⁸
- [Global Digital Formats Registry](#)⁹

The AONS II software also aims to help a Collection Manager to identify an appropriate risk profile and to manage the risk, supporting decisions on preservation action required to retain access to information resources stored on a range of repository types. Thus, AONS II is designed to enable users to be informed when file formats that exist in their repositories are obsolete or at risk of becoming obsolete” (from <http://www.apsr.edu.au/aons2/index.htm>¹⁰). Software can be downloaded from [Sourceforge](#)¹¹.

[DSTC](#)¹² conducted significant work on an “Integrated Preservation Framework” through the [PANIC](#)¹³ (Preservation webservices Architecture for Newmedia and Interactive Collections) project in 2003 – 2005. Further information on software obsolescence is available from their website:

- [links](#)¹⁴ to related projects
- [publications](#)¹⁵

Persistent Identifiers

Sustainable identifier infrastructure is required to deal with the vast amount of digital assets being produced and stored within universities. This is a particular challenge for e-Research communities where massive amounts of data are being generated without any means of managing this data over any length of time.

The [PILIN Project](#)¹⁶ was funded in 2006 for 15 months by DEST to explore the use of persistent identifiers in Australia and to make recommendations for a sustainable persistent identifier utility service.

PILIN's objective is to strengthen Australia's ability to use global identifier infrastructure. The [PILIN website](#)¹⁷ will make available guidance documents and best practice advice on identifiers. PILIN is focused on building software tools that will allow applications to use persistent identifiers more easily and to maintain the identifiers over time. It is also tasked with making recommendations to DEST on options for sustaining, supporting and governing identifier management infrastructure.

Broken web links cause frustration. In an IR environment, broken links may be caused by:

- a software upgrade and/or file restructure
- a change of domain name
- author initiated or management changes to a document
- removal of a document

[Handles](#)¹⁸ technology provides “efficient, extensible, and secure identifier and resolution services for use on networks such as the Internet” [Sefton](#)¹⁹ (2007). The application of Handles has been a recurrent discussion topic for RUBRIC partners throughout the project, some of which has been captured in [Peter Sefton's blog](#)²⁰.

There are many issues regarding the application of handles which are only now emerging as best practice and many organisations will be looking to the outcomes of the PILIN project for guidance in this area. The PILIN Project is producing software such as JAHDL (Java API for handles) to enable identifier integration. This software is available from the [PILIN website](#)²¹.

Organisational Support and Embedding

The IR Manager will be responsible for

- the development of relationships within the organisation
- presenting the IR as a solution to organisational needs

The Planning and Marketing stages of establishing an IR provide the groundwork on which the IR manager can build to ensure the repository becomes properly embedded into institutional practices and receives the necessary level of organisational support to ensure its success.

The IR Manager needs to foster a culture of deposit in the organisation. This will be influenced by the policies adopted in the pilot and planning stages, such as the deposit workflow.

Effective Strategies for making your repository popular and well loved is a useful resource compiled by Paula Callan from the Queensland University of Technology.

The University of Southern Queensland achieved a culture of deposit by:

- offering the ePrints repository as a solution for archiving 4th Year Engineering Projects
- working with the Office of Research and Higher Degrees to store reportable research publications, thus integrating with the normal research reporting process

- working with faculty librarians to communicate the benefits of an IR at the faculty level
- seeking and gaining mandated deposit for theses
- working with Mathematics and Computing to develop a web service to help them automatically create and update web pages of their faculty publications
- working with individual academics who appreciated exposure of their work and reduced workload in sending out emailed copies of pre-prints
- working with Marketing to help them promote the university's research and publications

Disaster Recovery

A disaster recovery plan should be developed as part of the risk management process. This plan should go beyond a sound backup strategy.

Consider what you would do in the following scenarios:

- the software you are using is withdrawn from circulation for some reason, such as it contains code that is in breach of copyright or infringes on patents
- the server is destroyed, and you are unable to restore the backups to run on new hardware because of subtle differences in the the operating environment
- one of the backup mechanisms you have been trusting, and which has apparently been working fails and three month's worth of data are lost

To mitigate such issues:

- always have a system-wide backup if possible, giving you a way to get a backup operational quickly. For RUBRIC virtual infrastructure, this meant backing up the disk images of virtual machines
- in addition to this have:
 - data exported in a standardised format, such as the export format from your repository.
(note: if you are using a proprietary system it may pay to have its export format independently audited; we found that at least one vendor, offering an online service at one stage offered an export that was not adequate to recover a repository).
 - the installation files for the IR and all configuration under version control. RUBRIC uses and recommends the Subversion system which is used by programmers to version-control code. Subversion allows you to keep track of every change made to configuration and to roll back, when required. It is ideal when used with a multi-tier development system as described in the Establishing a Pilot section.

References and Further Reading

Refer to the Further Reading section at the end of the Toolkit for bibliographic details of works referenced in this section.

For **action plans** to assist with repository and digital preservation management see the Digital Preservation Management section.

“RUBRIC Toolkit: Managing a Repository” produced May 2007



[Copyright](#)²² 2007 [RUBRIC](#)²³

- 1 <http://espace.lis.curtin.edu.au/archive/00000908/>
- 2 https://rubric-central.usq.edu.au/packages/RUBRIC_Toolkit/docs/Populating_the_Repository.htm
- 3 <http://www.apsr.edu.au/longterm/index.htm>
- 4 http://www.rlg.org/en/page.php?Page_ID=20769
- 5 <http://www.oclc.org/research/projects/pmwg/premis-final.pdf>
- 6 <http://www.apsr.edu.au/aons2/index.htm>
- 7 <http://www.digitalpreservation.gov/formats/intro/intro.shtml>
- 8 <http://www.nationalarchives.gov.uk/pronom/>
- 9 <http://hul.harvard.edu/gdfr/>
- 10 <http://www.apsr.edu.au/aons2/index.htm>
- 11 <http://sourceforge.net/projects/aons>
- 12 <http://www.dstc.edu.au/>
- 13 <http://metadata.net/panic/index.html>
- 14 <http://metadata.net/panic/Links.htm>
- 15 <http://metadata.net/panic/publications.htm>
- 16 <http://www.arrow.edu.au/PILIN>
- 17 <http://www.pilin.net.au/>
- 18 <http://www.handle.net/>
- 19 http://ptsefton.com/blog/2007/05/16/handles_curation_boundary
- 20 http://ptsefton.com/blog/2007/03/16/killer_handles
- 21 <http://www.pilin.net.au/>
- 22 <http://creativecommons.org/licenses/by-sa/2.5/au/>
- 23 <http://www.rubric.edu.au/>