wct Documentation

Release 2.0.0

Ben O’Brien

Nov 22, 2020

Overview:

CHAPTER 1

Overview and History

1.1 Additional TODO

- Add details of work from 2017 onwards between NLNZ and KB-NL.

1.2 Introduction

This guide, designed for non-technical users, provides a background and history of the Web Curator Tool.

1.2.1 Contents of this document

Following this introduction, the Web Curator Tool Overview and History Guide includes the following sections:

- **Overview** - Covers what the Web Curator Tool is and what it is not.
- **Screenshots** - Some screenshots of the Web Curator Tool.
- **History** - Covers the history of the tool from its inception to today.
- **License** - Covers the license used.
- **Release history** - Covers significant changes made in each release.

1.3 Overview

The Web Curator Tool (WCT) is a tool for managing the selective web harvesting process, and is designed for use in libraries by non-technical users. It is integrated with v1.14.1 of the Heritrix web crawler which is used to download web material (but technical details are handled behind the scenes by system administrators).

1.3.1 The WCT supports

- Harvest Authorisation: getting permission to harvest web material and make it available.
- Selection, scoping and scheduling: what will be harvested, how, and how often?
- Description: Dublin Core metadata.
- Harvesting: Downloading the material at the appointed time with the Heritrix web harvester deployed on multiple machines.
- Quality Review: making sure the harvest worked as expected, and correcting simple harvest errors.
- Submitting the harvest results to a digital archive.

1.3.2 What it is *NOT*

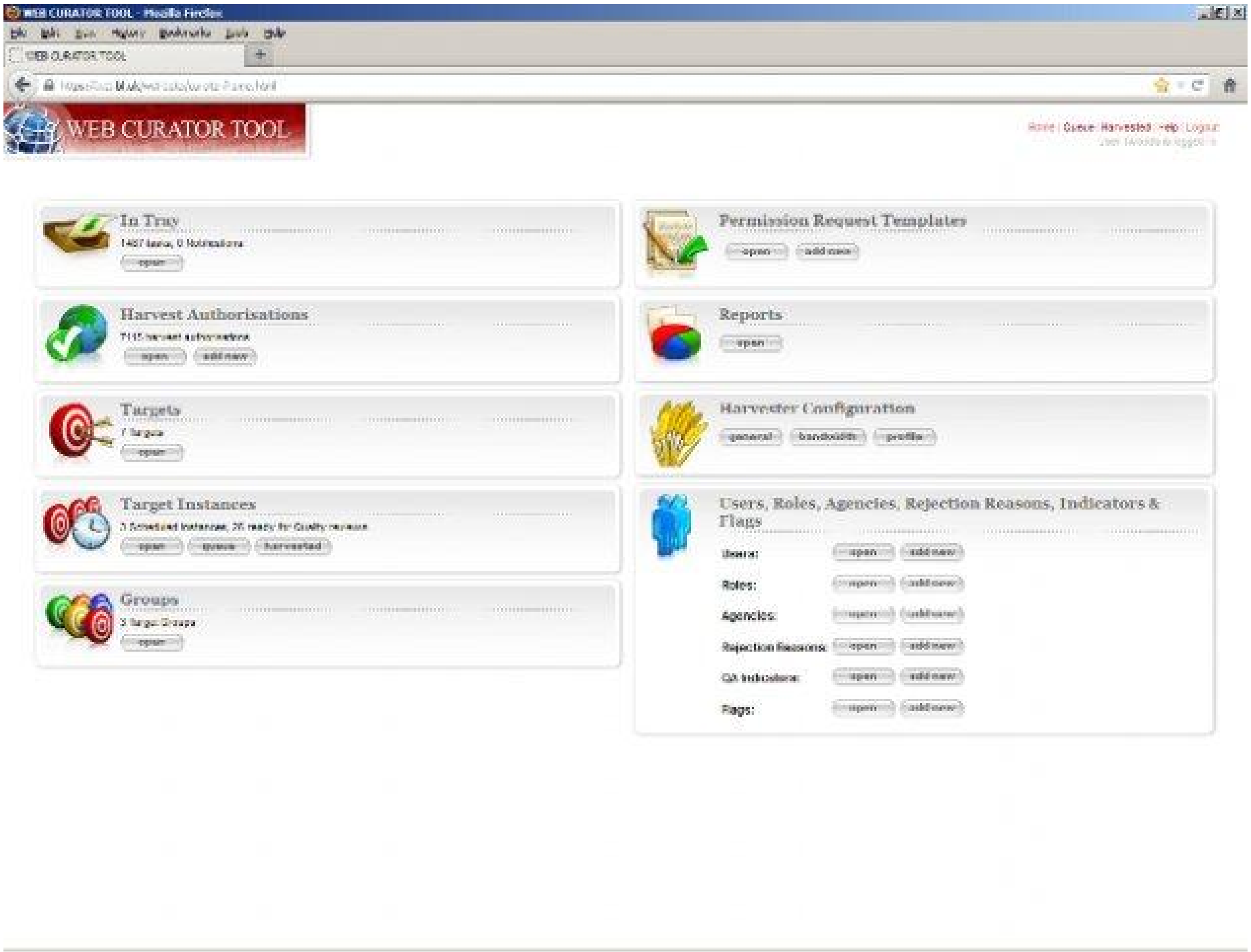
- It is *NOT* a digital archive or document repository - It is not appropriate for long-term storage - It submits material to an external archive
- It is *NOT* an access tool - It does not provide public access to harvested material - (But it does let you review your harvests) - You should use Wayback or WERA as access tools
- It is *NOT* a cataloguing system - It does allow you to record external catalog numbers - And it does allow you to describe harvests with Dublin Core metadata
- It is *NOT* a document management system - It does not store all your communications with publishers - But it may initiate these communications - And it does record the outcome of these communications

The Web Curator Tool supports a harvesting workflow comprising a series of specialised tasks:

- selecting an online resource
- seeking permission to harvest it and make it publicly accessible
- describing it
- determining its scope and boundaries
- scheduling a web harvest or a series of web harvests
- performing the harvests
- performing quality review and endorsing or rejecting the harvested material
- and depositing endorsed material in a digital repository or archive.

Most current web archiving activities rely heavily on the technical expertise of the harvest operators. The Web Curator Tool, on the other hand, makes harvesting the responsibility of users and subject experts (rather than engineers and system administrators) by handling automatically the technical details of web harvesting. The tool is designed to operate safely and effectively in an enterprise environment, where technical support staff can maintain it.

1.4 Screenshots



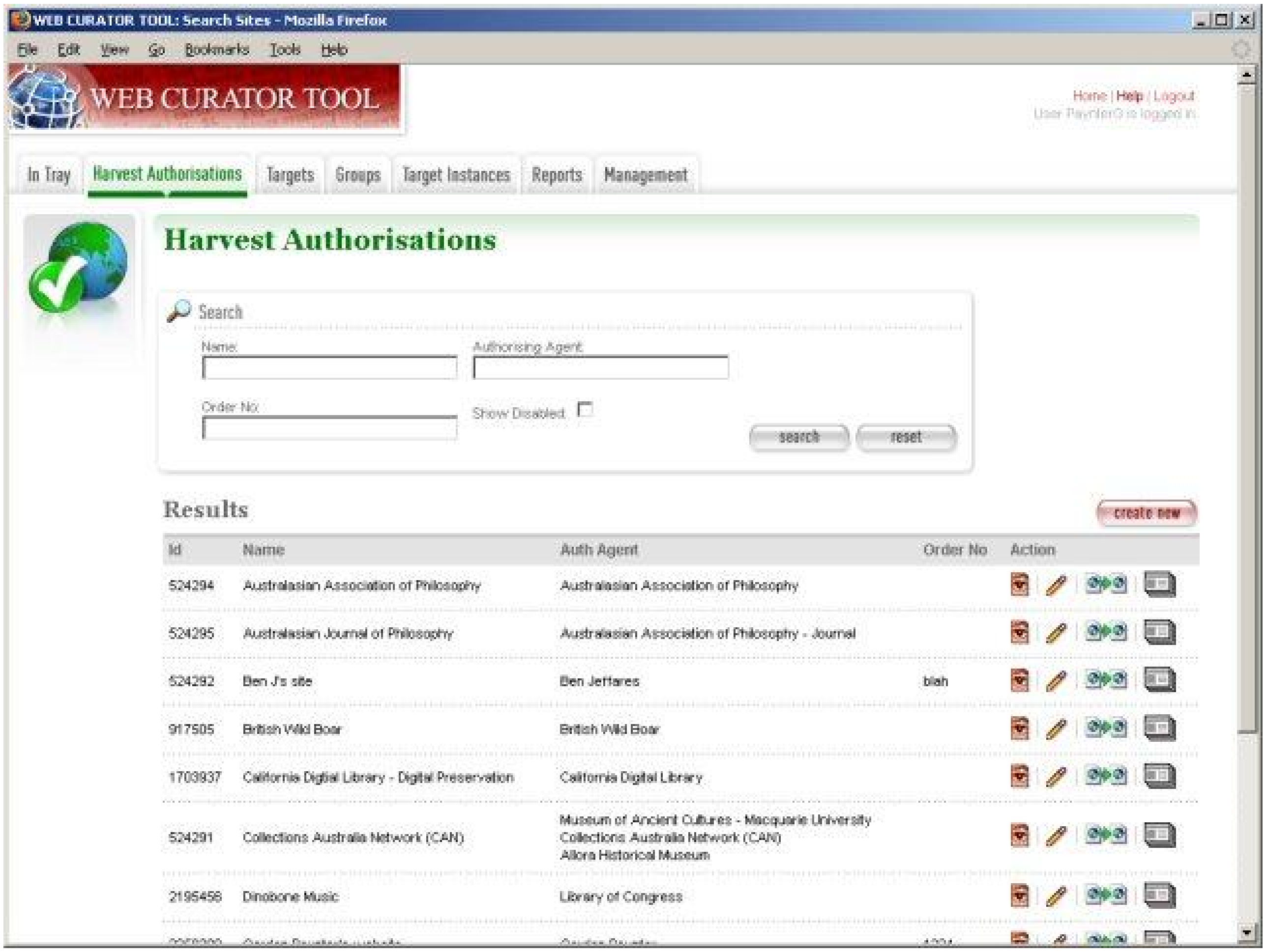
1487 tasks, 0 Notifications

7115 harvest authorisations

7 Targets

3 Scheduled Instances, 26 ready for Quality review

3 Target Groups



Harvest Authorisations

Search

Name:

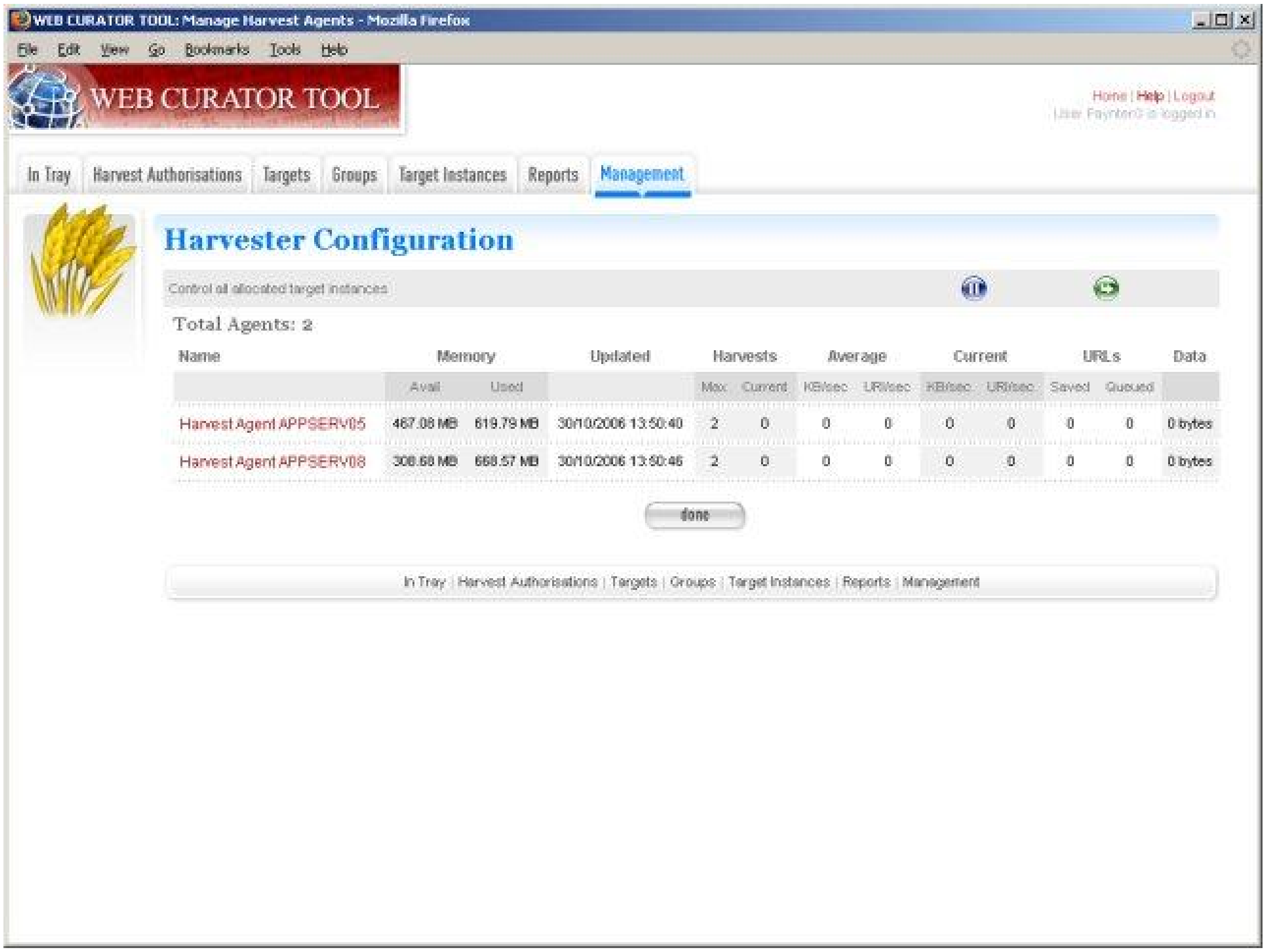
Authorising Agent:

Order No:

Show Disabled: □

Results

Id	Name	Auth Agent	Order No	Action
524294	Australasian Association of Philosophy	Australasian Association of Philosophy		
524295	Australasian Journal of Philosophy	Australasian Association of Philosophy - Journal		
524292	Ben J's site	Ben Jeffares	blah	
917505	British Wild Boar	British Wild Boar		
1703937	California Digital Library - Digital Preservation	California Digital Library		
524291	Collections Australia Network (CAN)	Museum of Ancient Cultures - Macquarie University Collections Australia Network (CAN) Allora Historical Museum		
2195456	Dinobone Music	Library of Congress		


WEB CURATOR TOOL: Manage Harvest Agents - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
WEB CURATOR TOOL
Home | Help | Logout
User Paynter3 is logged in
In Tray
Harvest Authorisations
Targets
Groups
Target Instances
Reports
Management

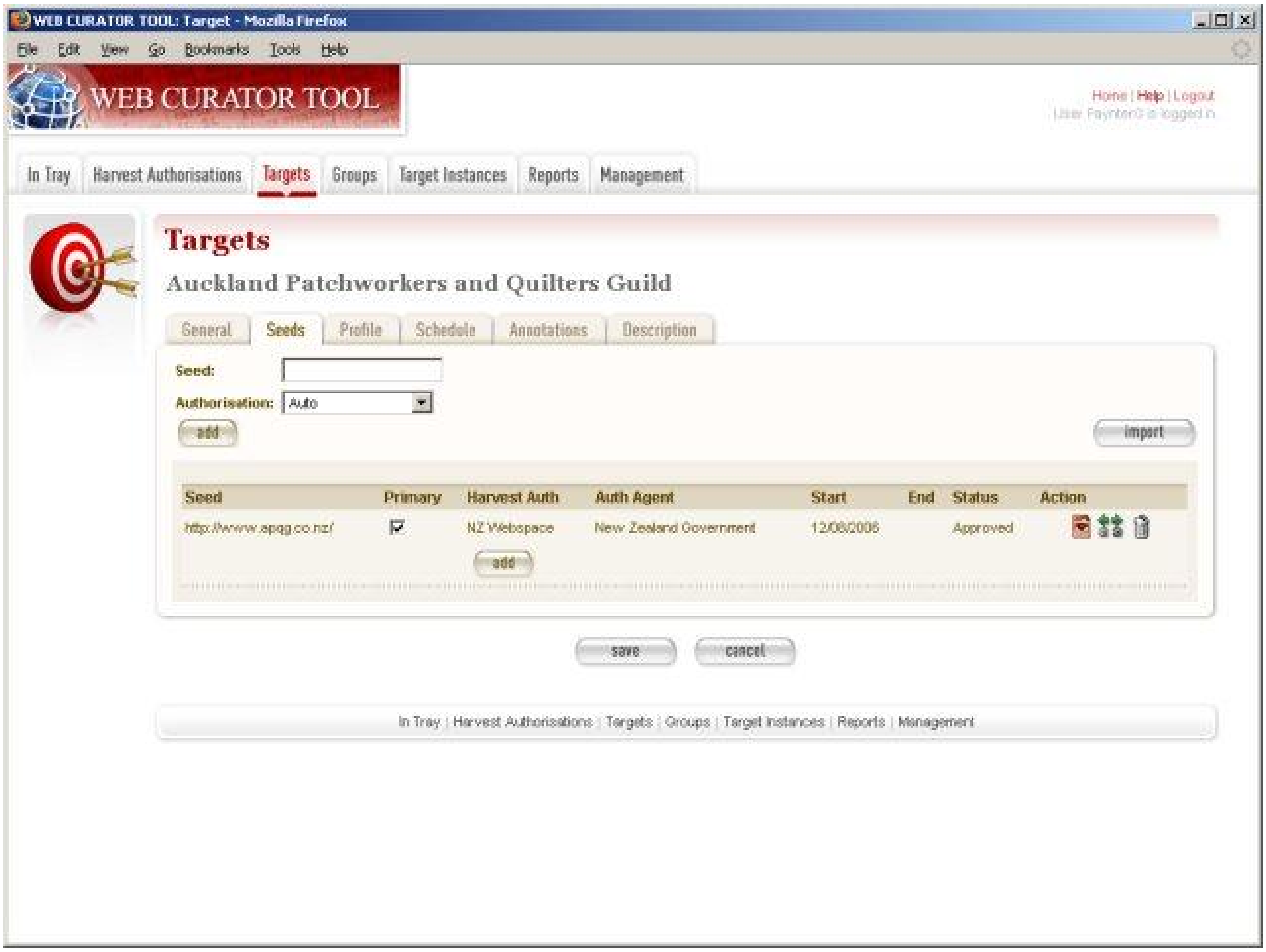
Harvester Configuration

Control all allocated target instances

Total Agents: 2

Name	Memory		Updated	Harvests		Average		Current		URLs		Data
	Avail	Used		Max	Current	KB/sec	URI/sec	KB/sec	URI/sec	Saved	Queued	
Harvest Agent APPSERV05	467.08 MB	619.79 MB	30/10/2006 13:50:40	2	0	0	0	0	0	0	0	0 bytes
Harvest Agent APPSERV08	308.68 MB	668.57 MB	30/10/2006 13:50:46	2	0	0	0	0	0	0	0	0 bytes

done
In Tray | Harvest Authorisations | Targets | Groups | Target Instances | Reports | Management


WEB CURATOR TOOL: Target - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
WEB CURATOR TOOL
Home | Help | Logout
User Paynter3 is logged in
In Tray
Harvest Authorisations
Targets
Groups
Target Instances
Reports
Management
General
Seeds
Profile
Schedule
Annotations
Description
add
import
add
save
cancel
In Tray | Harvest Authorisations | Targets | Groups | Target Instances | Reports | Management

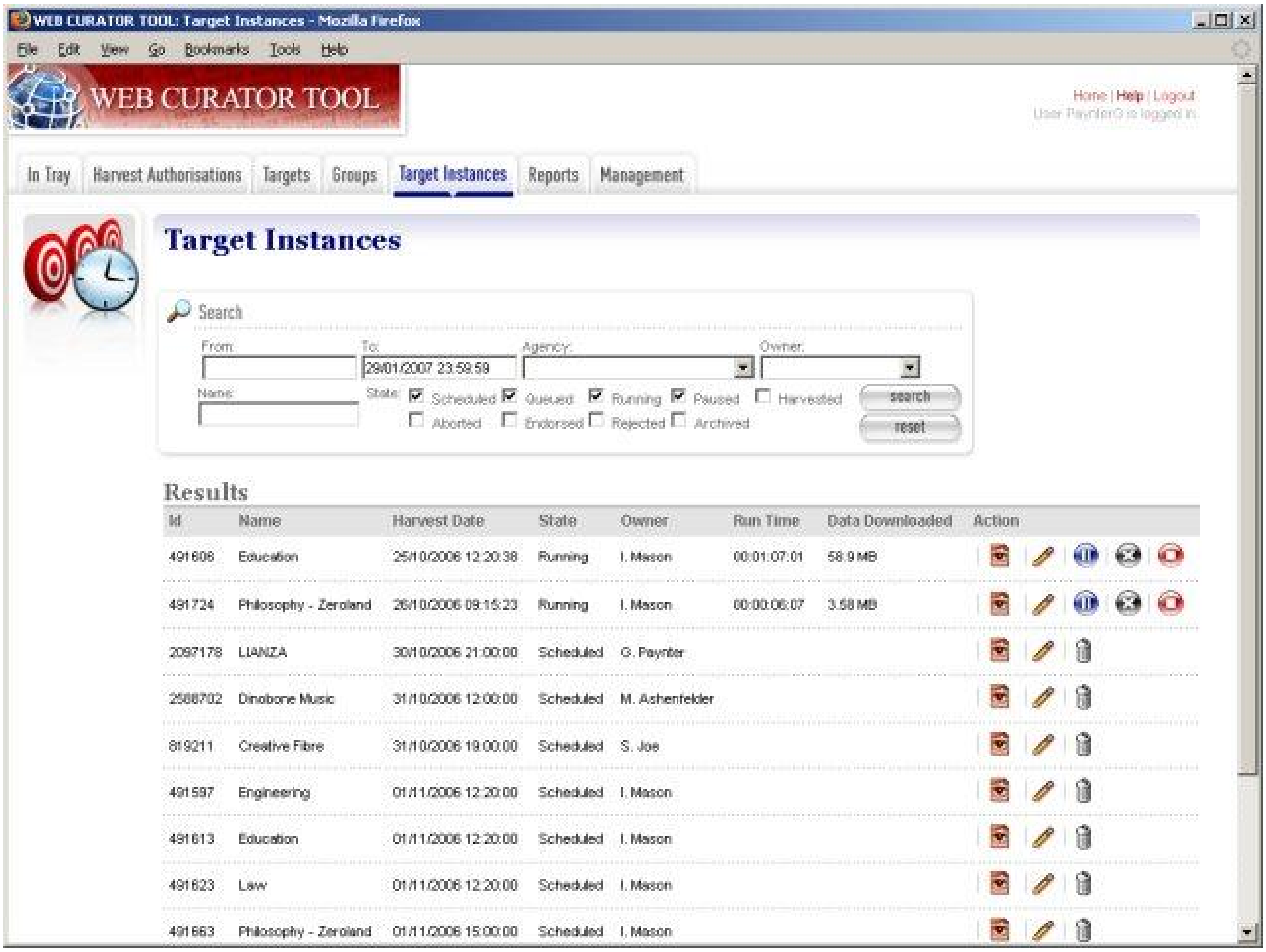
Targets

Auckland Patchworkers and Quilters Guild

Seed:

Authorisation: Auto

Seed	Primary	Harvest Auth	Auth Agent	Start	End	Status	Action
http://www.apqg.co.nz/	☑	NZ Webspace	New Zealand Government	12/08/2006		Approved	



Target Instances

From:

To: 29/01/2007 23:59:59

Agency:

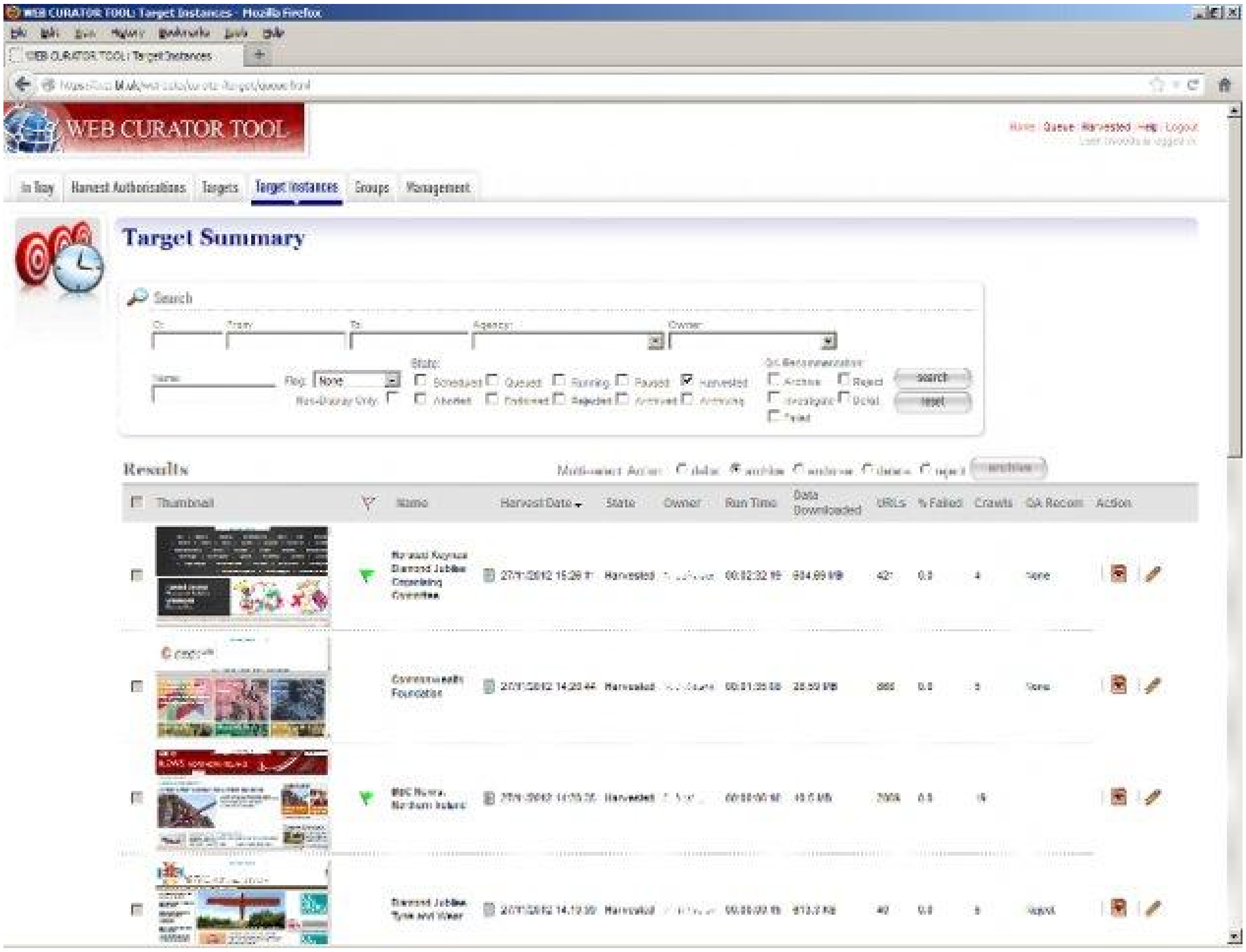
Owner:

Name:

State: ☑ Scheduled ☑ Queued ☑ Running ☑ Paused □ Harvested
□ Aborted □ Endorsed □ Rejected □ Archived

Results

Id	Name	Harvest Date	State	Owner	Run Time	Data Downloaded	Action
491606	Education	25/10/2006 12:20:38	Running	I. Mason	00:01:07:01	58.9 MB	
491724	Philosophy - Zeroland	26/10/2006 09:15:23	Running	I. Mason	00:00:06:07	3.58 MB	
2097178	LIANZA	30/10/2006 21:00:00	Scheduled	G. Paynter			
2588702	Dinobone Music	31/10/2006 12:00:00	Scheduled	M. Ashenfelder			
819211	Creative Fibre	31/10/2006 19:00:00	Scheduled	S. Joe			
491597	Engineering	01/11/2006 12:20:00	Scheduled	I. Mason			
491613	Education	01/11/2006 12:20:00	Scheduled	I. Mason			
491623	Law	01/11/2006 12:20:00	Scheduled	I. Mason			
491663	Philosophy - Zeroland	01/11/2006 15:00:00	Scheduled	I. Mason			


WEB CURATOR TOOL: Target Instances - Mozilla Firefox
File Edit View History Bookmarks Tools Help
WEB CURATOR TOOL: Target Instances
WEB CURATOR TOOL
Home | Queue | Harvested | Help | Logout
In Tray
Harvest Authorisations
Targets
Target Instances
Groups
Management
Target Summary
Search
search
reset
Results
archive

ID: From: To: Agency: Owner:

Name: Flag: None Non-Display Only □

State: □ Scheduled □ Queued □ Running □ Paused ☑ Harvested □ Aborted □ Endorsed □ Rejected □ Archived □ Archiving

QA Recommendation: □ Archive □ Reject □ Investigate □ Delist □ Failed

Multi-select Action: ○ delist ● archive ○ endorse ○ delete ○ reject

□	Thumbnail		Name	Harvest Date	State	Owner	Run Time	Data Downloaded	URLs	% Failed	Crawls	QA Recom	Action
□			[illegible] Diamond Jubilee Organising Committee	27/11/2012 15:28:1[illegible]	Harvested	[illegible]	00:02:32 [illegible]	604.69 MB	42[illegible]	0.0	4	None	
□			Commonwealth Foundation	27/11/2012 14:20:44	Harvested	[illegible]	00:01:35 [illegible]	28.59 MB	866	0.0	5	None	
□			BBC News, Northern Ireland	27/11/2012 14:[illegible]	Harvested	[illegible]	00:00:00 [illegible]	48.6 MB	2008	0.0	15		
□			Diamond Jubilee Tyne and Wear	27/11/2012 14:19:30	Harvested	[illegible]	00:00:00 [illegible]	813.2 KB	40	0.0	5	Reject	

1.5 History

The National Library of New Zealand has a legal mandate, and a social responsibility, to preserve New Zealand's social and cultural history, be it in the form of books, newspapers and photographs, or of websites, blogs and videos. Increasing amounts of New Zealand's documentary heritage is only available online. Users find this content valuable and convenient, but its impermanence, lack of clear ownership, and dynamic nature pose significant challenges to any institution that attempts to acquire and preserve it.

The Web Curator Tool was developed to solve these problems by allowing institutions to capture almost any online document, including web pages, web sites, and web logs, and most current formats, including HTML pages, images, PDF and Word documents, as well as multimedia content such as audio and video files. These artifacts are handled with all possible care, so that their integrity and authenticity is preserved. The public benefit from the safe, long-term preservation of New Zealand's online heritage is incalculable. Our online social history and much government and institutional history will be able to be preserved into the future for researchers, historians, and ordinary New Zealanders. They will be able to look back on our digital documents in the same way that the New Zealanders of today look back on the printed words left to us by previous generations.

The software was developed as a collaborative project between the National Library of New Zealand and the British Library, conducted under the auspices of the International Internet Preservation Consortium. The Web Curator Tool has been built with support and contributions from professionals at the National Library of New Zealand, the British Library, Sytec Resources Ltd., Oakleigh Consulting, the National Library of Australia, the Library of Congress, and many others.

1.5.1 Project objectives

- Meets the needs of the National Library of New Zealand
- Meets the needs of the British Library
- Is modular and can be extended to meet the needs of IIPC members and other organizations engaging in web harvesting
- Manages permissions, selection, description, scoping, harvesting and quality review
- Provides a consistent, managed approach allowing users with limited technical knowledge to easily capture web content for archival purposes.
- The National Library of New Zealand has used the Web Curator Tool as the basis of its selective web archiving programme since January 2007. It is the primary tool and responsibility of the web archivists in the Alexander Turnbull Library.

The tool is open-source software and is freely available for the benefit of the international web archiving community.

1.6 License

The Web Curator Tool is available under the terms of the Apache License, Version 2.0.

The tool is open-source software and is freely available for the benefit of the international web archiving community.

See the *Contributing* section of the *Developer Guide* for more details.

1.7 Release history

See *Release Notes* for release notes on specific versions.

1.7.1 15 March 2016 - 1.6.2 GA

This version includes bugfixes developed by the National Library of New Zealand between June 2015 and March 2016. This release fixes bugs relating to indexing, pruning and importing, duplicate schedules and more. The changes will benefit all existing WCT users and we recommend that users upgrade to 1.6.2

1.7.2 9 May 2014 - 1.6.1 GA

This version includes bugfixes and enhancements developed by the National Library of New Zealand between July and November 2013. This release streamlines the Webcurator workflow by adding date pickers for date fields; a heat map when scheduling harvests; harvest optimisation; Target description search and non-English character support. These features will benefit all existing WCT users and we recommend that users upgrade to 1.6.1

1.7.3 05 December 2012 - 1.6 GA

This version includes bugfixes and enhancements developed by the British Library between June and September 2012. This release streamlines the Webcurator workflow and enhances the automated quality assurance (QA) features. These features will benefit all existing WCT users and we recommend that users upgrade to 1.6.

1.7.4 22 August 2011 - 1.5.2 GA

This version includes many bugfixes and new features that were commissioned by The British Library and developed during March and April of 2011 by software developers from Oakleigh Consulting in the UK. The new features will be of benefit to all existing WCT users and we recommend that all users upgrade to 1.5.2.

1.7.5 10 December 2010 - 1.5.1 GA

This version includes many bugfixes, new features and performance improvements that were commissioned by The British Library and developed over two iterations during February and June of 2010 by software developers from Oakleigh Consulting in the UK. The new features will be of benefit to all existing WCT users and we recommend that all users upgrade to 1.5.1.

1.7.6 11 November 2009 - 1.5 GA

This version is mainly concerned with the optional integration of Wayback as an additional quality review tool, and the simplification of system configuration using properties files; but also contains a small number of additional enhancements and bugfixes. This release was commissioned by The British Library and developed during July and August of 2009 by software developers from Oakleigh Consulting in the UK. The new features will be of benefit to all existing WCT users and we recommend that WCT 1.1, 1.2.7, 1.3 and 1.4.x users upgrade to 1.5.

1.7.7 27 May 2009 - 1.4.1 GA

Version 1.4.1 of the Web Curator Tool is now available on Sourceforge. This version includes many bugfixes and an upgrade to Heritrix 1.14.1. There are also some new features and performance improvements that were commissioned by The British Library and developed over two iterations during September-October of 2008 and February-March of 2009 by software developers from Oakleigh Consulting in the UK. The new features will be of benefit to all existing WCT users and we recommend that WCT 1.1, 1.2.7, 1.3 and 1.4 users upgrade to 1.4.1.

1.7.8 29 July 2008 - 1.4 GA

This version includes an upgrade to Heritrix 1.14 and Wayback 1.2 as well as many new features and some performance improvements that were commisioned by The British Library and developed during an accelerated effort in April and June of 2008 by software developers from Oakleigh Consulting in the UK. The new features will be of benefit to all existing WCT users and we recommend that WCT 1.1, 1.2.7 and 1.3 users upgrade to 1.4.0.

1.7.9 Older releases

- 19 February 2008 - 1.3 GA
- 20 August 2007 - 1.2.7 GA
- 03 April 2007 - 1.1.1 Beta
- 21 September 2006 - 1.1 GA
- 21 August 2006 - 1.0 RC
- 25 July 2006 - 0.4 Beta

CHAPTER 2

Release Notes

2.1 Introduction

This guide, designed for a Web Curator Tool developer and system administrator, covers the release notes from version 1.5. Versions are in reverse chronological order, with the most recent version first. While the *Web Curator Tool System Administrator Guide*, *Web Curator Tool Developer Guide*, *Web Curator Tool Quick Start Guide* and *Web Curator Tool User Manual* are accurate for the current release, the *Release Notes* can give some idea of how things have changed since the last major release.

2.1.1 Contents of this document

Following this introduction, the Web Curator Tool Release Notes includes the following sections:

- **Changes since 2.0.2** - Changes since the last official release *2.0.2*.
- **2.0.2** - Release 2.0.2.
- **2.0.1** - Release 2.0.1.
- **2.0.0** - Release 2.0.0.
- **1.6.2** - Release 1.6.2.
- **1.6.1** - Release 1.6.1.
- **1.6.0** - Release 1.6.0.
- **1.5.2** - Release 1.5.2.
- **1.5.1** - Release 1.5.1
- **1.5** - Release 1.5.
- **Previous versions** - Versions prior to release 1.5.

2.2 Changes since 2.0.2

This is a placeholder for changes since the official *2.0.1* release. Please add notes here for changes and fixes as they are released into the master branch.

2.3 2.0.2

2.3.1 H3 Javascript Extractor Module

- An additional Heritrix 3 profile option has been added to the profile editor and the various profile override screens, to turn off the javascript extractor module. This modifies the following element:

  ```
  <property name="extractJavascript" value="false" />
  ```

2.4 2.0.1

- The SOAP implementation has changed. As part of that change, the ex-libris Rosetta SDK dependency has moved from *2.2.0* to *5.5.0*. This means that the *dps-sdk-5.5.0.jar* must be installed in a local Maven repository for the maven build to work. This jar is now sourced from the github project *rosetta-dps-sdk-projects-maven-lib*, found at (https://github.com/NLNZDigitalPreservation/rosetta-dps-sdk-projects-maven-lib). The installation of this jar includes a pom with its maven dependencies so the *wct-store* and *wct-submit-to-rosetta* subprojects no longer need to explicitly include or track the dps-sdk dependencies in their project poms. It has a different *groupId* and *artifactId* from *the dps-sdk-5.5.0.jar* downloaded from *Rosetta.dps-sdk-projects* (https://github.com/ExLibrisGroup/Rosetta.dps-sdk-projects). This dependnecy is installed into the local maven repository by running the script *install_maven_dependencies.[sh|bat]*.
- Because of some classpath issues, harvest-agent-h1 now uses a modified version of heritrix that has been created with the github project https://github.com/WebCuratorTool/heritrix-1-14-adjust. This version of heritrix and its necessary dependencies are installed into the local maven repository by running the script *install_maven_dependencies.[sh|bat]*. Note that this script now requires that the program *git* works from the command line.

2.5 2.0.0

Released December 2018, this version builds on release 1.7.0, which was a proof-of-concept integrating Heritrix 3 with WCT. Version 2.0.0 completes that integration.

2.5.1 What's new

Heritrix 3 profile management

- The configuration options available for Heritrix 3 are different from the old Heritrix 1 profiles, but management of them stays the same.
- Heritrix 3 profile options are contained within a single simplified 'scope' tab. This relies on a correctly formatted set of fields within the background profile xml. Due to this, imported Heritrix 3 profiles cannot be edited through the same screen, and are only editable via an in-screen xml editor.

- Validation of Heritrix 3 profiles is achieved using an available H3 Harvest Agent. The profile is used to build a special one-off job within the agent, which in essence validates the integrity of the Heritrix 3 profile. The job is then destroyed and any unsuccessful outcome is fed back to the WCT user interface.

Targets

- Heritrix 3 Targets can now be scheduled, and will be assigned to an available H3 Harvest Agent when due to run.
- *Running* Heritrix 3 Target Instances have an H3 script console available to use. This console can be used to run scripts against the Target Instance job in Heritrix 3, similar to the scripting console available in H3's own UI.

Heritrix 1

- Heritrix 1 integration has been preserved for now, allowing for Targets to transition to using Heritrix 3. A period of experimentation is expected when replacing the old Heritrix 1 profiles.

Database installation

- The sql scripts for setting up the WCT database have been consolidated and brought up to date. The folder structure has been refactored and legacy scripts separated to reduce confusion. Any script changes have been reflected in the documentation.
- An additional parent script has been added to simplify the setup process, enabling the setup to be completed through running a single script.

Documentation

- The documentation has been migrated from PDF to the reStructedText format, and now hosted on the readthedocs.io platform. This increases the accessibility of the documentation and makes it simpler to maintain and update.
- All documentation has been brought up-to-date to reflect v2.0.0 changes.

2.5.2 Developer

- The old Harvest Agent module has been separated into a Heritrix 1 and Heritrix 3 version. This has been done with a view to using the core Harvest Agent component to interface with other crawlers in the future.
- Usage of the old heritrix-1.14 dependency, *aheritrix-1.14.1.jar*, has been upgraded where possible to use the webarchive-commons library.

2.5.3 Things to be aware of

- The Bandwidth restriction functionality is not currently applicable to the new Heritrix 3 crawling. The Bandwidth feature has been underused in recent years and was not compatible out-of-the-box with Heritrix 3. A decision on whether to develop the feature to be compatible or remove it entirely will be made in the future.
- The existing prune and import functionality within the QA tool is not currently compatible with Target Instances harvested using Heritrix 3. These components of QA functionality are no longer fit-for-purpose in version 2.0.0, and will be re-developed as part of the WCT development road-map.

- The Groups feature is not currently compatible with Heritrix 3 profiles. This is intended to be resolved in the near future with a minor release.

2.6 1.6.3

This is the *WCT 1.6.3 GA* version.

Released July 2017, this version contains minor changes to the Submit-to-Rosetta module.

2.6.1 What's new

Alma compatibility upgrades for Submit to Rosetta module

Changes required by the National Library of New Zealand to be compatible with archiving to a Rosetta DPS integrated with Alma (library cataloguing and workflow management system from Ex Libris). All changes have been implemented as backward compatible as possible. The exposure of these changes and their configuration are through the files wct-das.properties, wct-das.xml inside WCT-Store.

2.7 1.6.2

This is the *WCT 1.6.2 GA* version.

2.7.1 Obtaining the source files

The WCT code is now stored in a GIT repository on sourceforge - available from the *code* link on the main WCT sourceforge project page.

The previous versions of WCT are available via the *Legacy Code* link, if needed. This is still a CVS repository.

2.7.2 Before installing

Please ensure the user that WCT uses to login to your database has the correct permissions to create temporary tables. Failure to grant this will result in problems during the purge process.

2.7.3 WCT new features and changes for v1.6.2

UI new features and improvements

Import https urls The import functionality on the Tree View screen for a harvest, now allows https URLs. Previously the javascript validation on the page only allowed http URLs.

Configurable Rosetta access rights The Rosetta access codes that are used in the Submit-to-Rosetta module are now configurable via the *wct-das.properties* file in the wct-store app. These codes are used in the mets.xml when a harvest is archived to Rosetta.

OMS Codes (Rosetta)

```
dpsArchive.dnx_open_access=xxxx
dpsArchive.dnx_published_restricted=xxxx
dpsArchive.dnx_unpublished_restricted_location=xxxx
dpsArchive.dnx_unpublished_restricted_person=xxxx
```

Submit-to-Rosetta compatibility with newer Rosetta versions Later versions of Rosetta system complained when performing xsd validation on the mets.xml file submitted by WCT when archiving a harvest. The structure map schema used by WCT was old. As Rosetta auto generates structure maps for deposits that are missing them, structure map generation was removed from the WCT process.

Allowing the version of Rosetta you are archiving to to generate the appropriate structure map.

Bug fixes

Quality Review tool uses original seed url The harvest quality review tools were not available previously if the original target seed URL was modified.

Now the target seed URL can be changed, and the QR tool will always look for the original URL of the Target Instance instead.

Pruning and importing for warc files fixed Pruning and importing on warcs in the Tree View screen was encountering a bug. When parsing a warc, the input stream was over-reading the number of bytes in the warc-info header, causing unexpected characters to be read when trying to access the next record. This was mainly visible when trying to import and prune.

Indexing breaking for compressed warcs Harvesting as compressed warc was breaking the indexing of a harvest. The Heritrix class handling the reading of the compressed warc was missing the functionality to move to the next record. The Heritrix library included has been recompiled to include a fix.

Duplicate schedules when saving annotations When creating/editing a Target - if a schedule is created/edited without saving the Target, and then the Target is saved whilst adding an annotation, WCT creates target instances for that schedule but the Target remains in a state where it contains a cache of new a schedule(s). So if the Target is then saved via the bottom save button, another group of target instances will be generated for the new schedule(s).

This bug has now been fixed. If a schedule already has target instances generated (at Annotations tab), then WCT will flag this to prevent any duplicates from being generated.

No *strippedcrawl.log* generated on non-windows os WCT was hard-coded to use a Windows file path separator when saving this log file. Now system specific file path separator is used.

Development related

Git stripping carriage returns Only affected JUnit tests for Submit-to-Rosetta module. The tests read in an arc file which originally contained a mix of lines ending in carriage returns + line feeds and line feeds. Once the project was moved to git, the carriage returns were stripped out, invalidating the character offset values in the arc file. The arc file is now stored in the test class as a string, in order to preserve all formatting.

Build process special characters All non-utf8 characters have been converted to utf8, and project POM files changed to build as utf8.

Code repository moved to Github Code repository moved to Github, along with all old content that possible to take from Sourceforge.

2.8 1.6.1

2.8.1 UI new features and improvements

Date pickers for date fields All date fields in WCT now have associated date pickers to aid in selection.

Edit button for view screens All possible view screens now have buttons to enable edit mode where the user has authority to edit the associated record.

Harvest optimization incl. global option There is now the option to specify harvest *optimization* on any target. This allows the harvesters to perform harvesting of the associated target instances earlier than the schedule otherwise permits. The window for this look-ahead is configurable, and defaults to 12 hours.

This feature can also be disabled on a global basis, temporarily, from the *Management->Harvester Configuration->General* screen. Upon restart this setting is enabled.

Harvester queue pause The queue for harvesters can now be paused on a per-harvester basis. This pause only affects harvests which have not yet started - it is still possible to pause harvests using the traditional mechanism. To activate/deactivate this feature, click the pause/play icon in the *Accept tasks* column on the *Management->Harvester Configuration->General* screen.

The intent of this is to be able to pause a specific harvester in order to stop it and perform maintenance once harvests are finished.

Scheduling heat map A heat map is now available on the target scheduling screen. This allows a user to see a rough overview of when jobs are scheduled in the next few months or so, in order to choose a day where harvesting is least intensive.

To view the heat map, visit the *Target->Schedule->edit/new schedule* page. Click the calendar icon labelled *heat map* - the days will be colored based on how many harvests are scheduled on those days.

The color of the heat map, and the thresholds used to display the colors, are configurable on the *Management->Harvester Configuration->Bandwidth* page. This allows organizations of any size to customize the heat map to the capabilities of their harveters.

Import profile to any agency The profile import page now has the ability to select any agency to import a profile into. This option is only available when the logged in user has the authority to manager the profiles for all agencies. When this authority is not present, that user's agency is used for the import.

Ability to delete all intray tasks There is now a button to allow the deletion of all intray tasks, intended mainly for organizations that do not make use of the tasks.

Ability to hide intray tasks Similar to the ability to delete all tasks, the tasks can also be hidden from view on a per-agency basis. The configuration for this feature is on the edit agency page.

Target URL edit It is now possible to edit Target URLs once they have been created. Note that this will affect all existing and future scheduled target instances!

Target description search It is now possible to search inside the description of targets on the target listing screen.

Reply-to email address in permissions There is now a configurable *reply-to* email address on the permission template edit screen. This will show in most modern email clients as *Reply-to:* and should be used as the default reply to address in clients which support it.

Annotations prompt to save When clicking the *add* button for annotations, a prompt now asks whether the user wants to save the associated target, target instance, or group.

Indicator flag color picker improvement The indicator flag color picker now updates when colors are selected, rather than having to click the color wheel icon in the bottom right.

Completed harvests can be *harvested now* where user has authority, state is reset to *Approved* Where a user has authority to reinstate and approve a target, they do not need to manually change the state to *approved* when adding a new schedule or using *harvest now*.

Completed harvests can have schedules added where user has authority, state is reset to *Approved* Where a user has authority to reinstate and approve a target, they do not need to manually change the state to *approved* when adding a new schedule or using *harvest now*.

Groups with sub-groups can now be styled using CSS The text for groups with sub-groups in the group listing screen can now be styled using CSS.

Rejection reason is shown against rejected harvest results The rejection reason was not visible in any UI element for a rejected harvest result. This has been added to the harvest result listing screen.

2.8.2 Bug fixes

Non-english character support for all WCT screens (providing database is configured correctly) When the database is configured to support UTF-8 characters, the user interface now supports non-english characters on all screens, including permissions emails.

If you are experiencing problems with UTF-8 characters after this release, ensure that the database tables explicitly support UTF-8.

Non-existant scheduling alert When attempting to create a Target schedule which falls on non-existant dates, an alert will be displayed. This is show for custom schedules as well as any schedule with a frequency of monthly or less.

For example, a monthly schedule on the 30th day of the month will not fire in February, and a monthly schedule on the 31st day of the month will only fire seven months a year, as February, April etc have less than 31 days.

Profile null pointers fixed Null pointer exceptions caused by the absence of a default profile have been fixed. This was especially a problem when users were creating targets using the *bootstrap* user, and was generally experienced by new users of WCT.

Various other null pointers fixed A variety of other *NullPointerException* errors have been fixed.

Permissions orphan records The database was amended so that permissions records were not duplicated then orphaned when any change to permissions was made. In organizations where a lot of permissions changes were made, this could result in a large number of orphaned records.

Indicator flags can now only be applied to targets for the same agency Previously if a user had the "manage flags" authority they could assign any indicator flag to any target instance. This can result in users without that privilege from being able to find those target instances during a TI search by indicator flag.

Updated Target Instance edit screen to only allow indicator flags for the same agency as the owner of the target.

Viewing other TIs in harvest history changes the TI being reviewed When reviewing a target instance, clicking on any other target instance in the harvest history screen caused a change in the target instance originally being reviewed. In some cases users were endorsing the wrong target instance, believing that they were still reviewing the one they originally chose to review.

The target instance being reviewed now does not change unless the user decides to review the one selected in the target history, and a warning is displayed indicating this fact.

Target instances are now completely created for targets with repeating schedules A bug was introduced in WCT 1.6 that meant target instances were not created when adding a schedule to a target and saving. Any subsequent saves would create one target instance, but it could result in missing target instances. This has been fixed.

Max width of target, QA indicator screens has been limited to prevent scroll bars When using particularly long seeds or target names, a scroll bar on the target listing screen was necessary, similarly for the QA indicator listing. The table contents are now wrapped and sized appropriately.

The eSerial *next* function (used by NLNZ) has been included on the QA Target Instance Summary page Previously, the archive button would not show the *custom deposit form* for Rosetta. A *next* button now allows this function as per the Harvest Result screen.

Deletion of harvest resources fixed (requires that WCT database can create temporary tables) A potential problem with the deletion of harvest resources was fixed - a null pointer exception was possible, which meant that only one harvest was deleted per execution of the purge process. Additionally, the WCT database user needs authority to create temporary tables (e.g. for Oracle *GRANT CREATE TEMPORARY TABLE to usr_wct*)

2.8.3 Development related

Jetty/H2 database standalone development environment It is no longer necessary to install tomcat, a database etc to get a basic WCT environment set up and running.

See the *Developer Guide* for details.

Database upgrade script fixes Problems encountered by various users in the database upgrade scripts have been corrected. Upgrade scripts for 1.6.1 have been explicitly tested in all three databases.

Sourceforge tickets cleaned up and up-to-date Some sourceforge tickets had been fixed in the code, but not yet updated. Others were no longer necessary, or not possible to change as are not part of WCT. These have been investigated and resolved where applicable.

2.9 1.6.0

Release 1.6.0 greatly enhances the automated quality assurance (QA) features by providing a preview of each harvest and an automated recommendation. It contains a large number of updates summarised in the list below. Further details can be found in the release notes in the download and on the website.

2.9.1 Updates

FT001 Added config parameter to enable new QA module

FT002 Added new target instance summary screen (QA control and streamlines access to other WCT functions)

FT005 Added the QA Recommendation Service

FT006 Added website preview to target instances screen

FT007 Extended target instance flags (enables adhoc grouping)

FT008 Enhanced target instance search screen (sortable columns, filters and annotations as tooltips)

FT009 Integrated existing schedule service into new summary screen

FT011 Added 'Auto-prune' service

FT010 New Report: Heritrix Status Code Summary

FT003 New Report: Crawl differential comparison (New URIs + Matching URIs + Missing URIs)

FT012 New Report: URL count by Domain Summary

FT013 New Report: Off-scope URIs

FT014 New Report: Long URIs

FT015 New Report: Unknown MIME Types

FT016 New Report: robots.txt entries disallowed

FT017 New Report: Repeating patterns in URIs

2.9.2 SourceForge bug fixes

In addition, the following SourceForge bug fixes have been applied:

3434492 Warc write process with prune tool

2989826 Group schedule target to harvest agent errors

2870218 HibernateOptimisticLockingFailureException

2.9.3 Community and internal testing bug fixes

The following bugs have also been fixed as a result of user community and internal testing:

- Memory leak caused by target instances being pinned into memory by tag-pooling (also see 'WCT Configuration and Deployment Guide (WCT 1.6).pdf')
- Removed target instance from session after exiting edit mode
- Malformed initial <select> HTML tag within the <wct:list> tag caused options to disappear

2.10 1.5.2

Release 1.5.2 is primarily a maintenance update of the Web Curator Tool. It contains a number of bugfixes and enhancements. These are summarised in the list below:

Further details for each item can be found in the SourceForge Enhancement Tracker where relevant.

- SourceForge Ref: 3162584 - Fix bug where Targets with open schedules were erroneously going to completed status
- SourceForge Ref: 3162582 - Fix problem with Illegal group reference error on review
- SourceForge Ref: 3169679 - Add Rejection Reason functionality
- SourceForge Ref: 3162580 - Fix bug where errors thrown when Re-start indexing used
- SourceForge Ref: 3072266 - Implemented batch re-assignment of Target profiles to fix issues such as 'bad effect on Approved targets when profile is Disabled'.
- SourceForge Ref: 2941648 - Add facility to reject harvests stuck in 'restart indexing'
- SourceForge Ref: 2952587 - Approved targets should stay approved after profile edits - enhanced logic regarding Target state changes when editing profiles
- SourceForge Ref: 2940542 - Seed URL too long for database column - column width increased
- SourceForge Ref: 3162604 - By default checkProcessor bean will be disabled in harvest agent
- SourceForge Ref: 3162649 - Property file update - to fix typographical error in das.properties file
- SourceForge Ref: 3162600 - Date locale - bandwith settings bug
- SourceForge Ref: 3025576 - Fix bug introduced by Endorse/unendorse actions in WCT version 1.5
- SourceForge Ref: 3162610 - fix absolute paths coded into certain jsp and css files
- SourceForge Ref: 2943743 - Fix bug causing error after approving a target in some circumstances
- SourceForge Ref: 3006785 - Log IP address of harvested files to the crawl.log

- SourceForge Ref: 3162609 - Disk check bean not checking correct partition
- SourceForge Ref: 3162581 - Fix bug where URIListRegExpFilter not working in some profiles
- SourceForge Ref: 2983692 - Correct the permission checking logic where users are allowed to create permission request templates
- SourceForge Ref: 3162597 - Add 'view target instances' link to Target Action Column
- SourceForge Ref: 2983693 - Add new field to Harvest Auths permissions tab to allow saving of permission responses
- SourceForge Ref: 3016176 - Crawler Activity Report modifications - add filters
- SourceForge Ref: 2970877 - Link to associated target instance records from Harvest History table and navigate back again
- SourceForge Ref: 3097070 - Fix profile issues regarding redundant fields in DecideRules when setting scope
- SourceForge Ref: <none, BL specific enhancement> - Switch Automated QA on/off on a per target basis
- SourceForge Ref: <none, BL specific enhancement> - Allow the importing of missing content into a harvest via the wctaqa report
- SourceForge Ref: <none, BL specific enhancement> - SOAP API call enhancements which allow automation of certain setup tasks from external applications

2.11 1.5.1

Release 1.5.1 is primarily a maintenance update of the Web Curator Tool. It contains a number of bugfixes, enhancements and performance improvements. These are summarised in the lists below:

Further details for each item can be found in the SourceForge Tracker.

2.11.1 Enhancements

- SourceForge 2935731: Ability to add missing files to a harvest before submitting to archive
- SourceForge 1828045: Ability to set harvest alerts, triggered from Target record via alertable annotations
- SourceForge 2892942: Ability to export and import profiles to xml files
- SourceForge 2892356: Ability to sort the views of targets, target instances and harvest authorisations by name and date
- SourceForge 2934308: Ability to view hidden targets, target instances and groups (where display flag is false)
- SourceForge 2892361: Highlight the primary seed URL on target records
- SourceForge 2892965: Set Targets to Completed status when appropriate
- SourceForge 2932069: Ability to create Group sub-categories
- SourceForge 1828045: Display alert icon against scheduled TIs when associated Target has alertable annotations
- SourceForge 2932065: Add a 'Submit to Archive' icon to action column of endorsed Target Instances
- SourceForge 2931964: Facility to add Annotations on Access tab of Targets/Groups and the Display tab of TIs
- SourceForge 2892358: Capture WCT and Heritrix version numbers used when harvesting, display on Target Instance
- SourceForge 2892367: Distinguish between first time harvests and repeat harvests for a given Target

- SourceForge 2617632: Implement Tree Tool display enhancements
- SourceForge 2511377: Add ability to display crawl path (hop path) in tree tool view
- SourceForge 2892363: Display the date that permissions letter/email was sent on Harvest Authorisations view
- SourceForge 1774427: Selection Note and Evaluation Note fields on Target record annotations tab were too short

2.11.2 Bugfixes

- Fixed issue 2932075: Allow pre v1.5 harvests to be reviewed using wayback
- Fixed issue 2892945: Harvest profile description field length bug
- Fixed issue 2156378: Two default active profiles causes crash on new target screen
- Fixed issue 2931967: Submitting Target instance to archive now returns user to instances list screen
- Fixed issue ???????: WCT timeout was occuring when reviewing large harvests (raised directly via BL no Sourceforge ref)
- Fixed issue 3004090: Slow performance when accessing WCT targets/instances with group schedules
- Fixed issue 2931964: Annotations on Access tab of Targets and the Display tab of TIs
- Fixed issue 2928219: System Activity Report slow or unresponsive
- Fixed issue 1557611: Name duplication conflict between Target and Group names

2.12 1.5

Release 1.5 is the fourth major update of the Web Curator Tool. This release is mainly concerned with the optional integration of Wayback as an additional quality review tool, and the simplification of system configuration using properties files; but also contains a small number of additional enhancements and bugfixes summarised in the list below. Further details for each item can be found in the SourceForge Tracker.

2.12.1 Enhancements

- Quality Review Update to use wayback (2807159)
- Properties file for spring config (2807161)
- Autopopulate dublin core title field from target title (2815658)
- Target section multiple action on seeds (2811357)
- Add *Harvested* link to list of quick links (SourceForge# 2811364)
- Ability to submit to a Rosetta based archive

2.12.2 Bugfixes

- Fixed issue 2815654: Reindexing fails
- Fixed issue 2807164: MYSQL install file update
- Fixed issue 2810210: Sub-group deletion exception

- Fixed issue 2775423: Browse tool throwing exceptions with bad URI’s

2.13 Previous versions

This docuemnt does not include the *Release notes* for versions before 1.5.

CHAPTER 3

User Manual

3.1 Introduction

3.1.1 About the Web Curator Tool

The Web Curator Tool is a tool for managing the selective web harvesting process. It is typically used at national libraries and other collecting institutions to preserve online documentary heritage.

Unlike previous tools, it is enterprise-class software, and is designed for non-technical users like librarians. The software was developed jointly by the National Library of New Zealand and the British Library, and has been released as free software for the benefit of the international collecting community.

3.1.2 About this document

This document is the Web Curator Tool User Manual. It describes how to use the Web Curator Tool through its web browser interface. It assumes your system administrator has already set up the Web Curator Tool.

The manual is divided into chapters, each of which deals with a different aspect of the tool. The chapters generally correspond to the major Web Curator Tool modules.

System administrators will find an Administrators Guide and other technical documentation on the Web Curator Tool website (http://dia-nz.github.io/webcurator/).

3.1.3 Where to find more information

The primary source for information on the Web Curator Tool is the website:

http://dia-nz.github.io/webcurator/

The Github project pageThe includes links to download the tool, its corner that leads to the Github project page. Here you can navigate to the Web Curator Tool Wiki which is also hosted on Github.

Each page in the Web Curator Tool has a Help link in the top right corner that leads to the Github project page. Here you can navigate to the Web Curator Tool Wiki which is also hosted on Github.

3.2 System Overview

3.2.1 Background

More and more of our documentary heritage is only available online, but the impermanence and dynamic nature of this content poses significant challenges to any collecting institutions attempting to acquire it.

To solve these problems, the National Library of New Zealand and The British Library initiated a project to design and build a selective web harvesting tool, which has now been released to the collecting community as the Web Curator Tool.

3.2.2 Purpose and scope

The tool is designed to manage the selective web archiving process. It supports a harvesting workflow comprising a series of specialised tasks with the two main business processes supported being acquisition and description.

The Web Curator Tool supports:

- Harvest Authorisation: obtaining permission to harvest web material and make it publicly accessible;
- Selection, scoping and scheduling: deciding what to harvest, how, and when;
- Description: adding basic Dublin Core metadata;
- Harvesting: downloading the selected material from the internet;
- Quality Review: ensuring the harvested material is of sufficient quality for archival purposes; and
- Archiving: submitting the harvest results to a digital archive.

The scope of the tool is carefully defined to focus on web harvesting. It deliberately does not attempt to fulfil other enterprise functions:

- it is not a digital repository or archive (an external repository or archive is required for storage and preservation)
- it is not an access tool
- it is not a cataloguing system (though it does provide some support for simple Dublin Core metadata)
- it is not a document or records management system

Other, specialised tools can perform these functions more effectively and the Web Curator Tool has been designed to interoperate with such systems.

3.2.3 Essential terminology

Important terms used with the Web Curator Tool include:

- **Web Curator Tool** or **WCT** - a tool for managing the selective web harvesting process.
- **Target** - a portion of the web you want to harvest, such as a website or a set of web pages. Target information includes crawler configuration details and a schedule of harvest dates.
- **Target Instance** - a single harvest of a Target that is scheduled to occur (or which has already occurred) at a specific date and time.

- **harvest** or **crawl** - the process of exploring the internet and retrieving specific web pages.
- **harvest result** - the files that are retrieved during a **harvest**.
- **seed** or **seed url** - a starting URL for a harvest, usually the root address of a website. Most harvests start with a seed and include all pages "below" that seed.
- **harvest authorisation** - formal approval for you to harvest web material. You normally need permission to harvest the website, and also to store it and make it accessible.
- **permission record** - a specific record of a harvest authorisation, including the authorising agencies, the dates during which permissions apply and any restrictions on harvesting or access.
- **authorising agency** - a person or organisation who authorises a harvest; often a web site owner or copyright holder.
- **indicator** - a quality assurance metric used to quantify the success of a harvest (e.g. the amount of content downloaded)
- **recommendation** - the advice obtained by using one or more indicators to determine if a harvest successfully captured the content from a website
- **automated QA** - the automated quality assurance process that runs after a harvest completes that provides a recommendation
- **flag** - an arbitrary group created and assigned to one or more target instances
- **reference crawl** - a target instance that has been archived and marked as a baseline to which all future harvests will be compared for a specific target
- **harvest optimisation** - enables a harvest to run at the optimum time when there is available space in the schedule. The default is to look forward 12 hours (configurable).
- **heat map** - a calendar 'pop up' that indicates the spread of scheduled harvests over a period of time.

3.2.4 Impact of the tool

The Web Curator Tool is used at the National Library of New Zealand, and has had these impacts since it was introduced into the existing selective web archiving programme:

- Harvesting has become the responsibility of librarians and subject experts. These users control the software handling the technical details of web harvesting through their web browsers, and are much less reliant on technical support people.
- Many harvest activities previously performed manually are now automated, such as scheduling harvests, regulating bandwidth, generating preservation metadata.
- The institution's ability to harvest websites for archival purposes has been improved, and a more efficient and effective workflow is in place. The new workflow ensures material is safely managed from before it is harvested until the time it enters a digital archive.
- The harvested material is captured in ARC/WARC format which has strong storage and archiving characteristics.
- The system epitomises best practice through its use of auditing, permission management, and preservation metadata.

3.2.5 How Does it Work?

The Web Curator Tool has the following major components

The Control Centre

- The Control Centre includes an access-controlled web interface where users control the tool.
- It has a database of selected websites, with associated permission records and other settings, and maintains a harvest queue of scheduled harvests.

Harvest Agents

- When the Control Centre determines that a harvest is ready to start, it delegates it to one of its associated harvest agents.
- The harvest agent is responsible for crawling the website using the Heritrix web harvester, and downloading the required web content in accordance with the harvester settings and any bandwidth restrictions.
- Each installation can have more than one harvest agent, depending on the level of harvesting the organization undertakes.

Digital Asset Store

- When a harvest agent completes a harvest, the results are stored on the digital asset store.
- The Control Centre provides a set of quality review tools that allow users to assess the harvest results stored in the digital asset store.
- Successful harvests can then be submitted to a digital archive for long-term preservation.

3.3 Home Page

The **Web Curator Tool Home Page** is pictured below.

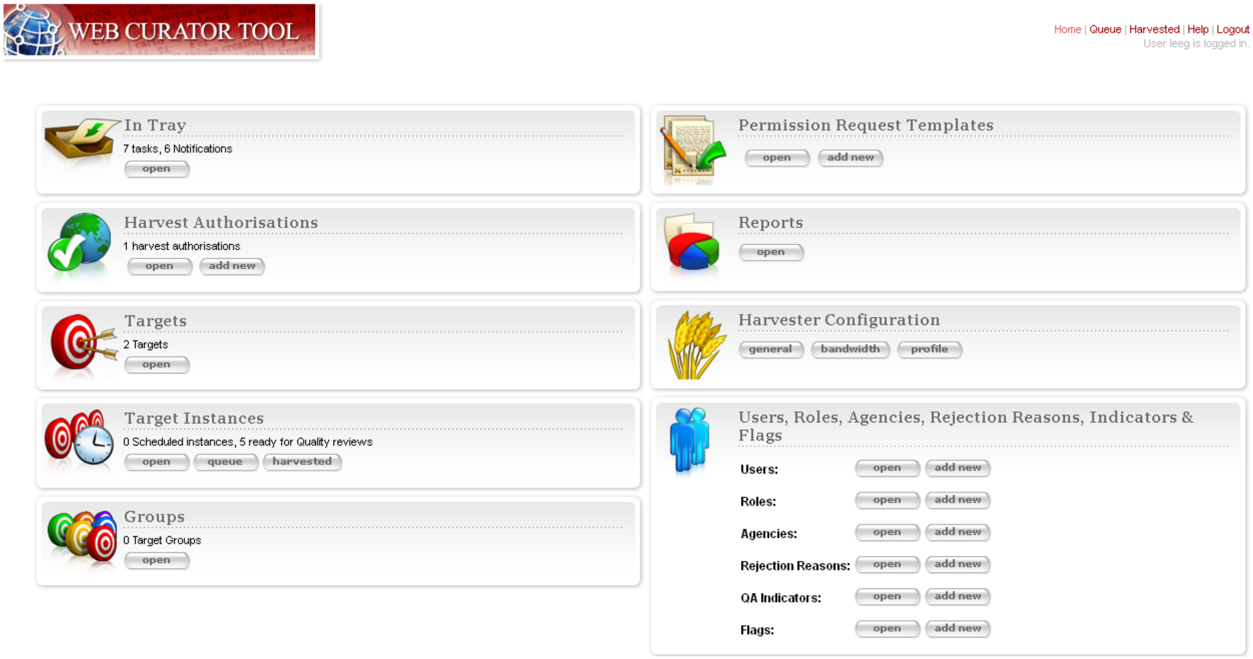


Fig. 1: Figure 1. Home Page

The left-hand side of the homepage gives access to the functionality used in the selection and harvest process:

In Tray - view tasks that require action and notifications that display information, specific to the user

Harvest Authorisations - create and manage harvest authorisation requests

Targets - create and manage Targets and their schedules

Target Instances - view the harvests scheduled in the future and review the harvests that are complete

Groups - create and manage collections of Targets, for collating meta-information or harvesting together

The right-hand side of the homepage gives access to administrative functions:

Permission Request Templates - create templates for permission request letters

Reports -generate reports on system activity

Harvest Configuration - view the harvester status, configure time-based bandwidth restrictions (how much content can be downloaded during different times of the day or week) and harvest profiles (such as how many documents to download, whether to compress them, delays to accommodate the hosting server, etc.)

Users, Roles, Agencies, Rejection Reasons, Indicators & flags - create and manage users, agencies, roles, privileges, rejection reasons, QA indicators and flags

The functions that display on the **Web Curator Tool Home Page** *depend on the user's privileges.*

3.4 Harvest Authorisations

3.4.1 Introduction

When you harvest a website, you are making a copy of a published document. This means you must consider copyright law when you harvest material, and also when you preserve it and when you make it accessible to users.

The Web Curator Tool has a sophisticated **harvest authorisation module** for recording your undertakings to copyright holders. Before you can harvest web pages, you must first confirm you are authorised to do so. The Web Curator Tool will record this information in its audit trail so that the person or agency that authorised a particular harvest can always be found. If you do not record who has authorised the harvest, the Web Curator Tool will defer the harvest until you confirm you are authorised.

In most cases, getting "harvest authorisation" means you must get permission from the website owner before you start the harvest. The Web Curator Tool lets you create harvest authorisation records that record what website or document you have requested permission for, who has authorised you to perform the crawl, whether you have been granted permission, and any special conditions.

Some institutions, such as national libraries, operate under special legislation and do not need to seek permission to harvest websites in their jurisdiction. The Web Curator Tool supports these organisations by allowing them to create a record that covers all such cases. See the section on **Legislative and other sources of information** below.

In other cases, your institution may decide to harvest a website before seeking permission, possibly because the target material is time-critical and it is in the public interest to capture it right away. In these cases, you must still record the entity who authorised the crawl, even if it is a person in your organisation, or even you yourself. This is also covered in the section on **Legislative and other sources of information** below.

Commercial search engines often harvest websites without seeking permission from the owners. Remember that these services do not attempt to preserve the websites, or to republish them, so have different legal obligations.

3.4.2 Terminology and status codes

Terminology

Important terms used with the Harvest Authorisation module include:

- **harvest authorisation** - formal approval for you to harvest web material. You normally need the copyright holder's permission to harvest the website, and also to store it and make it accessible.
- **authorising agency** - a person or organisation who authorises a harvest; often a website owner or copyright holder.
- **permission record** - a specific record of a harvest authorisation, including the authorising agencies, the dates during which permissions apply and any restrictions on harvesting or access.
- **url pattern** - a way of describing a URL or a set of URLs that a permission record applies to. For example, http://www.example.com/* is a pattern representing all the URLs on the website at www.example.com.

Permission record status codes

Each permission record has one of these status codes:

- **pending** - the permission record has been created, but permission has not yet been requested.
- **requested** - a request for permission has been sent to the authorising agency, but no response has been received.
- **approved** - the authorising agency has granted permission.
- **rejected** - the authorising agency has refused permission.

URL Patterns

URL Patterns are used to describe a portion of the internet that a harvest authorisation applies to.

In the simplest case, a URL can be used as a URL Pattern. In more complex cases, you can use the wildcard * at the start of the domain or end of the resource to match the permission to multiple URLs.

For example:

- **http://www.alphabetsoup.com/*** -include all resources within the Alphabet Soup site (a standard permission granted directly by a company)
- **http://www.alphabetsoup.com/resource/*** -include only the pages within the 'resource' section of the Alphabet Soup site
- **http://*.alphabetsoup.com/*** -include all resources on all sub sites of the specified domain.
- **http://www.govt.nz/*** -include all pages on the domain www.govt.nz
- **http://*.govt.nz/*** -include all NZ Government sites
- **http://*.nz/*** -include all sites in the *.nz domain space (this can be used to supports a national permission based on government legislation)

3.4.3 How harvest authorisations work

Each harvest authorisation contains four major components:

- A name and description for identifying the harvest authorisation, plus other **general information** such as an order number.
- One or more **authorising agencies**, being the person or organisation who authorises the harvest. This is often a website owner or copyright holder. Some authorising agencies may be associated with more than one harvest authorisation.
- A set of **url patterns** that describe the portion of the internet that the harvest authorisation applies to.

- One or more **permission records** that record a specific permission requested from an authorising agency, including
  - a set of URL patterns,
  - the state of the request (pending, requested, approved, rejected),
  - the time period the request applies to, and
  - any special conditions or access restrictions (such as 'only users in the Library can view the content').

In most cases, only users with specific roles will be allowed to manage harvest authorisations. Unlike some other Web Curator Tool objects, harvest authorisations do not have an "owner" who is responsible for them.

3.4.4 Sample harvest authorisation

For example, to harvest web pages from 'The Alphabet Soup Company', you might create a harvest authorisation record called 'Alphabet Soup'. This would include:

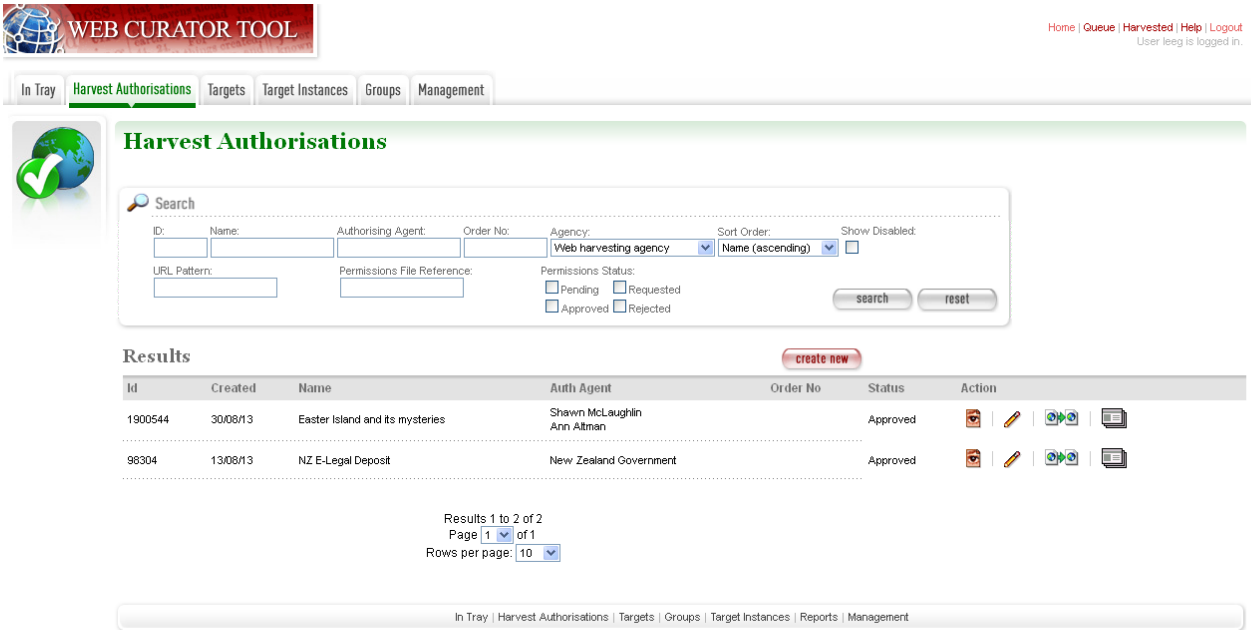
- **general information** recording the company name and the library order number for this request:
  - Name: 'Alphabet Soup'
  - Order Number: "AUTH 2007/03"
- **url patterns** to identify the company's three websites:
  - http://www.alphabsetsoup.com/*
  - http://www2.alphabsetsoup.com/*
  - http://extranet.alphabsetsoup.com/*
- **authorising agencies** for the two organisations responsible for the content on these sites:
  - The Alphabet Soup Company
  - Food Incorporated.
- **permission records**, linking each authorising agency with one or more URL patterns:
  - The Alphabet Soup Company to approve restriction-free access, on an open-ended basis, to http://www.alphabetsoup.com/* and http://www2.alphabetsoup.com/*
  - Food Incorporated to approve NZ-only access, for the period 1/1/2006 through 31/12/2006, to http://www.alphabetsoup.com/* and http://www2.alphabetsoup.com/*.

3.4.5 Harvest authorisation search page

The harvest authorisation search page lets you find and manage harvest authorisations.

At the top of the page are:

- Fields to enter search criteria for existing harvest authorisation records (**Identifier**, **Name**, **Authorising Agent**, **Order Number, Agency, URL Pattern, Permissions File Reference** and **Permissions Status**), and a search button for launching a search.
- There is also a drop down list that allows the user to define a sort order for the returned results **(name ascending, name descending, most recent record displayed first, oldest record displayed first)**
- A button to **create new** harvest authorisation requests.



Harvest Authorisations

Search

ID: | Name: | Authorising Agent: | Order No: | Agency: Web harvesting agency | Sort Order: Name (ascending) | Show Disabled: □

URL Pattern: | Permissions File Reference: | Permissions Status: □ Pending □ Requested □ Approved □ Rejected

Results

Id	Created	Name	Auth Agent	Order No	Status	Action
1900544	30/08/13	Easter Island and its mysteries	Shawn McLaughlin Ann Altman		Approved	
98304	13/08/13	NZ E-Legal Deposit	New Zealand Government		Approved	

Results 1 to 2 of 2
Page 1 of 1
Rows per page: 10

In Tray | Harvest Authorisations | Targets | Groups | Target Instances | Reports | Management

Fig. 2: Figure 2. Harvest Authorisations

Below that are search results. For each harvest authorisation record found, you can:

- **View** details

- **Edit** details

- **Copy** the harvest authorisation and make a new one.

- **Generate a permission request letter**.

The first time you visit this page, all the active harvest authorisations for the user's Agency are shown. You can then change the search parameters. On subsequent visits, the display is the same as the last harvest authorisation search.

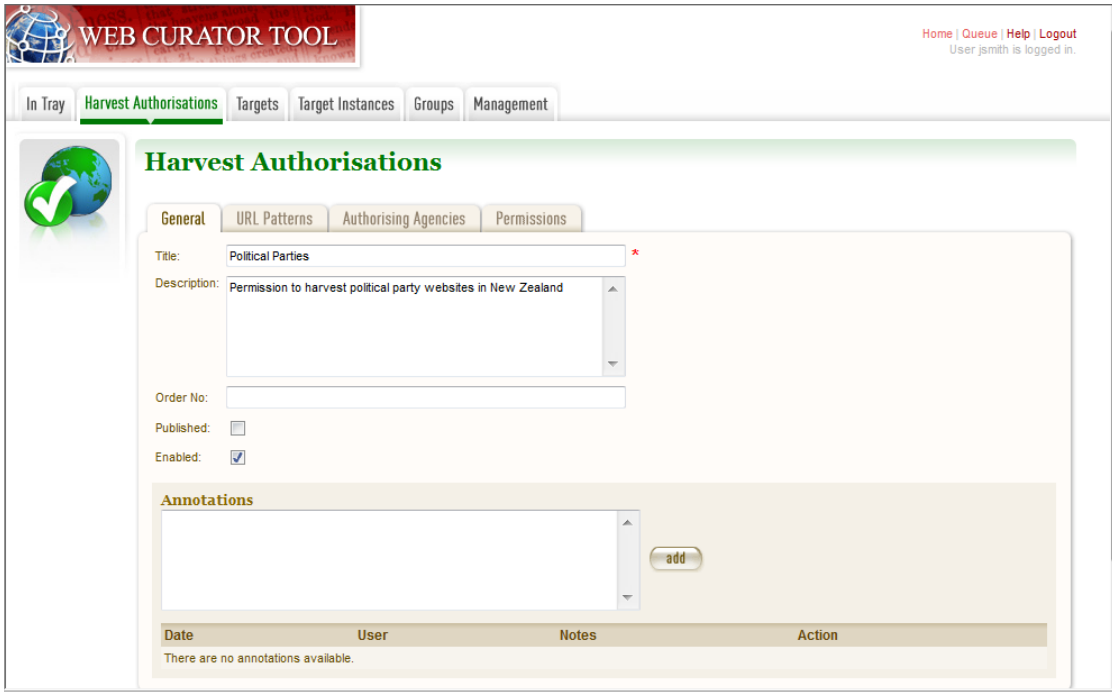
All search pages that present the search results in a 'page at a time' fashion have been modified so that the user can elect to change the default page size from 10 to 20, or 50 or even 100! The user's preference will be remembered across sessions in a cookie.

3.4.6 How to create a harvest authorisation

From the Harvest Authorisations search page:

1. Click **create new**.

The **Create/Edit Harvest Authorisations** page displays:



Title:	Political Parties *
Description:	Permission to harvest political party websites in New Zealand
Order No:	
Published:	□
Enabled:	☑

Annotations

Date	User	Notes	Action
There are no annotations available.			

Figure 3. Create/Edit Harvest Authorisations

The page includes four tabs for adding or editing information on a harvest authorisation record:

- **General** - general information about the request, such as a name, description and any notes
- **URLs** - patterns of URLs for which you are seeking authorisation
- **Authorising Agencies** - the persons and/or organisations from whom you are requesting authorisation
- **Permissions** - details of the authorisation, such as dates and status.

Enter general information about the request

2. On the **General** tab, enter basic information about the authorisation request.

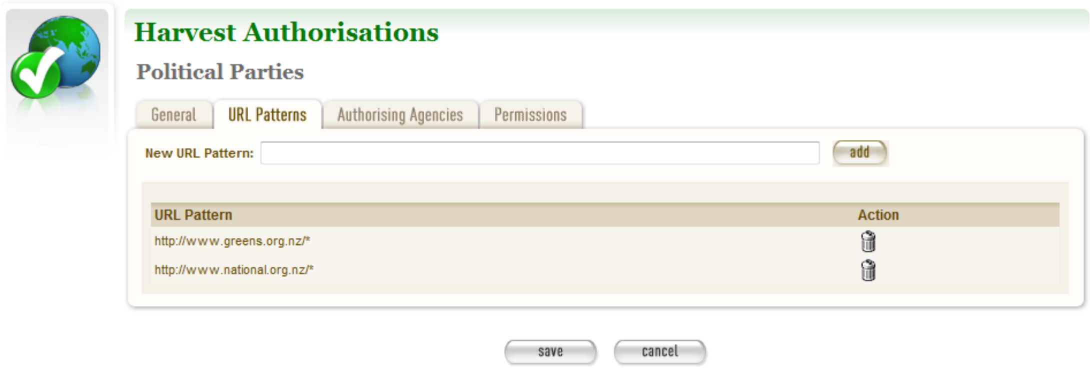
Required fields are marked with a red star. When the form is submitted, the system will validate your entries and let you know if you leave out any required information.

3. To add a note (annotation) to the record, type it in the Annotation text field and click **add**.

Enter URLs you want to harvest

4. Click the **URL Patterns** tab.

The **URL Patterns** tab includes a box for adding URL patterns and a list of added patterns.



New URL Pattern:

URL Pattern	Action
http://www.greens.org.nz/*	🗑
http://www.national.org.nz/*	🗑

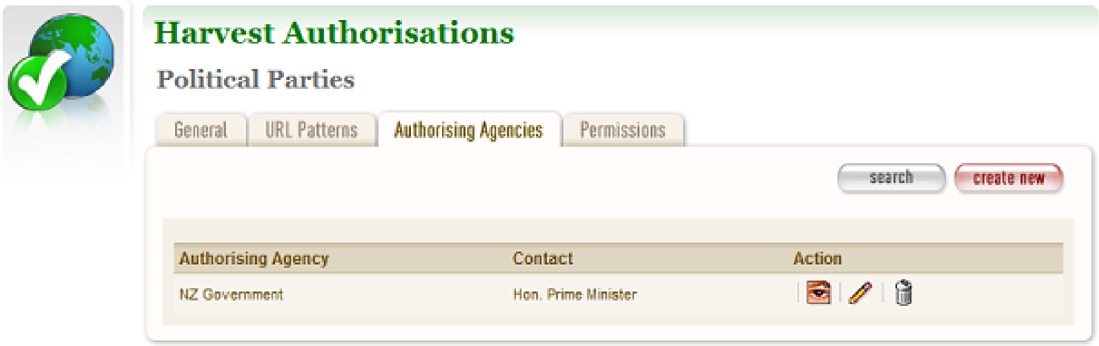
Figure 4. URL Patterns tab

5. Enter a pattern for the URLs you are seeking permission to harvest, and click **add**. Repeat for additional patterns.

Enter agencies who grant permission

6. Click the **Authorising Agencies** tab.

The **Authorising Agencies** *tab includes a list of authorising agencies and buttons to search for or create new agencies.*

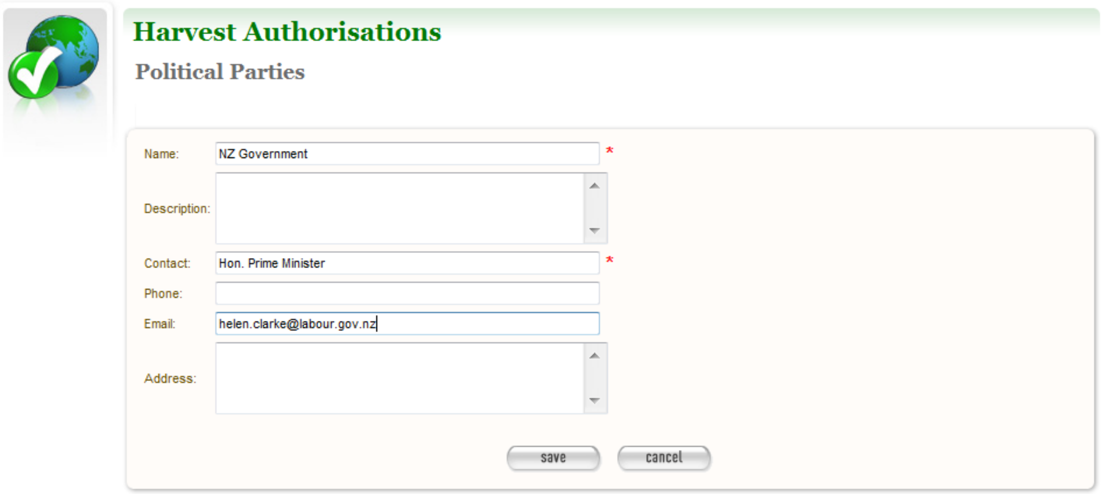


Authorising Agency	Contact	Action
NZ Government	Hon. Prime Minister	👁 ✎ 🗑

Figure 5. Authorising Agencies tab

7. To add a new agency, click **create new**.

The **Create/Edit Agency** page displays.



Name: NZ Government *

Description:

Contact: Hon. Prime Minister *

Phone:

Email: helen.clarke@labour.gov.nz

Address:

Figure 6. Create/Edit Agency

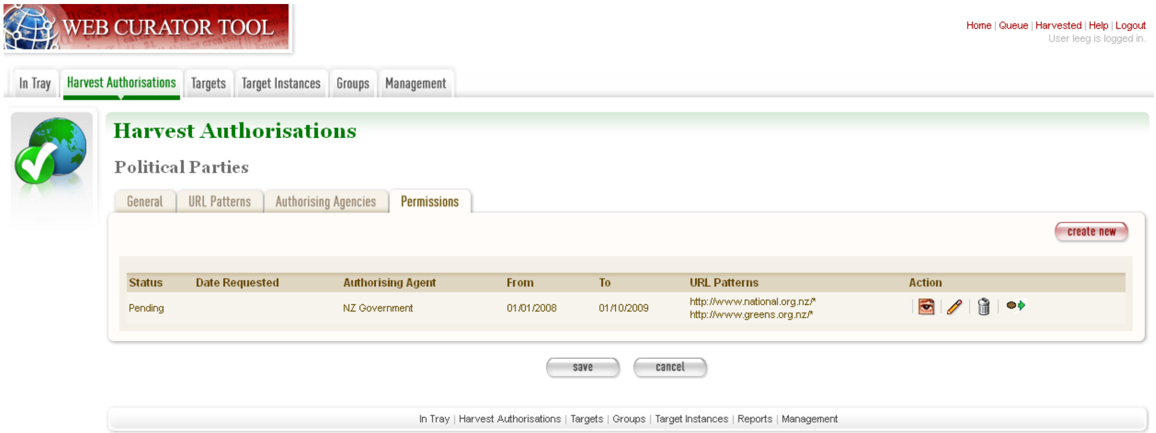
8. Enter the name, description, and contact information for the agency; and click **Save**.

The Authorising Agencies tab shows the added agency.

Create permissions record

9. Click the **Permissions** tab.

The **Permissions** *tab includes a list of permissions requested showing the status, agent, dates, and URL pattern for each.*



Harvest Authorisations

Political Parties

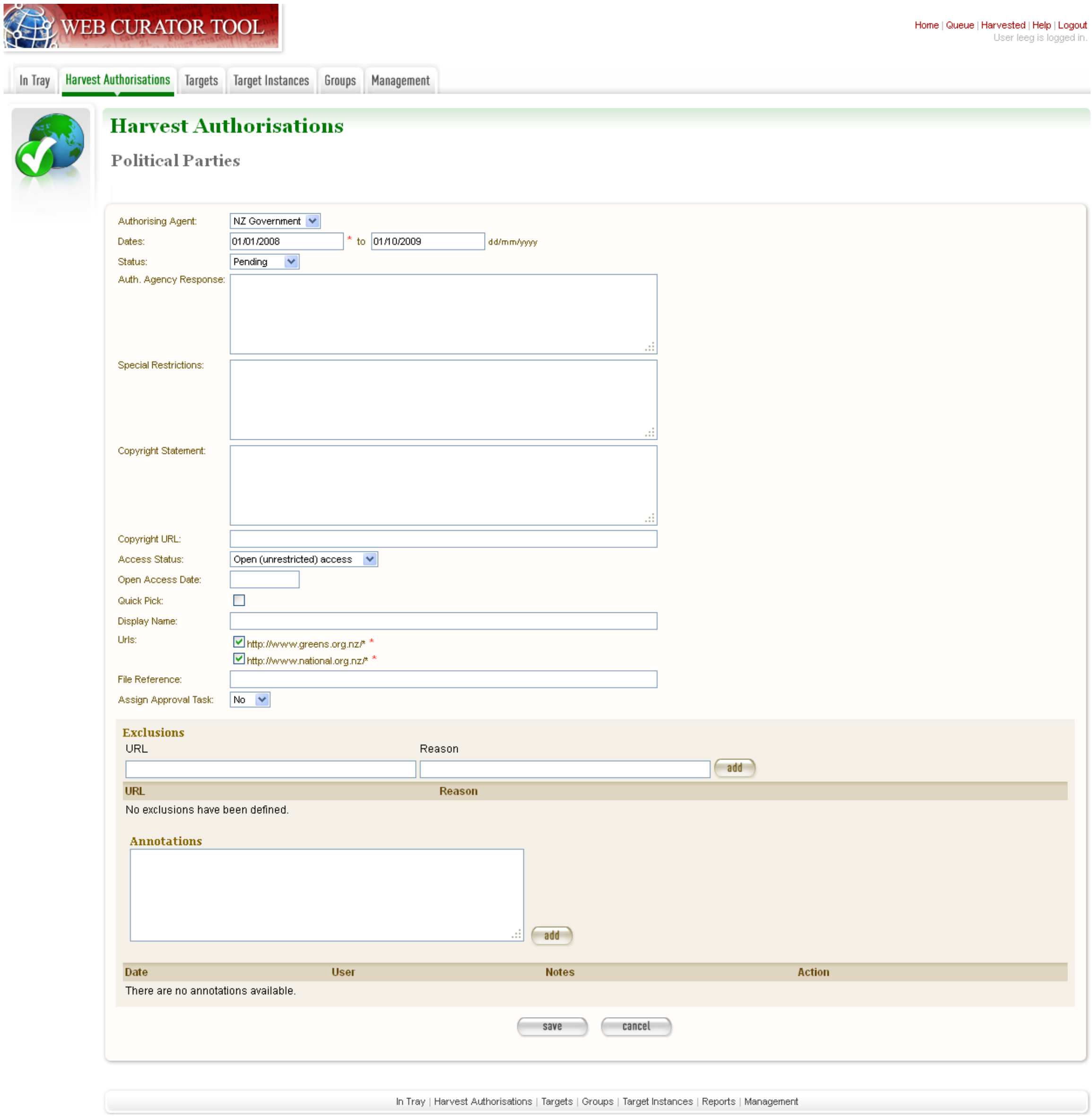
Status	Date Requested	Authorising Agent	From	To	URL Patterns	Action
Pending		NZ Government	01/01/2008	01/10/2009	http://www.national.org.nz/* http://www.greens.org.nz/*	

Figure 7. Permissions tab

10. The date requested column shows the date that a permission request (email or printed template) was generated.

11. To add a new permission, click **create new**.

The **Create/Edit Permission** *page displays.*


WEB CURATOR TOOL
Home | Queue | Harvested | Help | Logout
User leeg is logged in.
In Tray
Harvest Authorisations
Targets
Target Instances
Groups
Management
add
add
save
cancel
In Tray | Harvest Authorisations | Targets | Groups | Target Instances | Reports | Management

Harvest Authorisations

Political Parties

Authorising Agent:	NZ Government
Dates:	01/01/2008 * to 01/10/2009 dd/mm/yyyy
Status:	Pending
Auth. Agency Response:	
Special Restrictions:	
Copyright Statement:	
Copyright URL:	
Access Status:	Open (unrestricted) access
Open Access Date:	
Quick Pick:	□
Display Name:	
Urls:	☑ http://www.greens.org.nz/* * ☑ http://www.national.org.nz/* *
File Reference:	
Assign Approval Task:	No

Exclusions

URL Reason

URL	Reason

No exclusions have been defined.

Annotations

Date	User	Notes	Action

There are no annotations available.

Figure 8. Create/Edit Permission

12. Select an agent, enter the dates you want to harvest, tick the URL patterns you want to harvest, enter special restrictions, etc.;

    and click **Save**.

The **Permissions tab** *redisplays, showing the added permission.*

13. Click **Save** to save the harvest authorisation request.

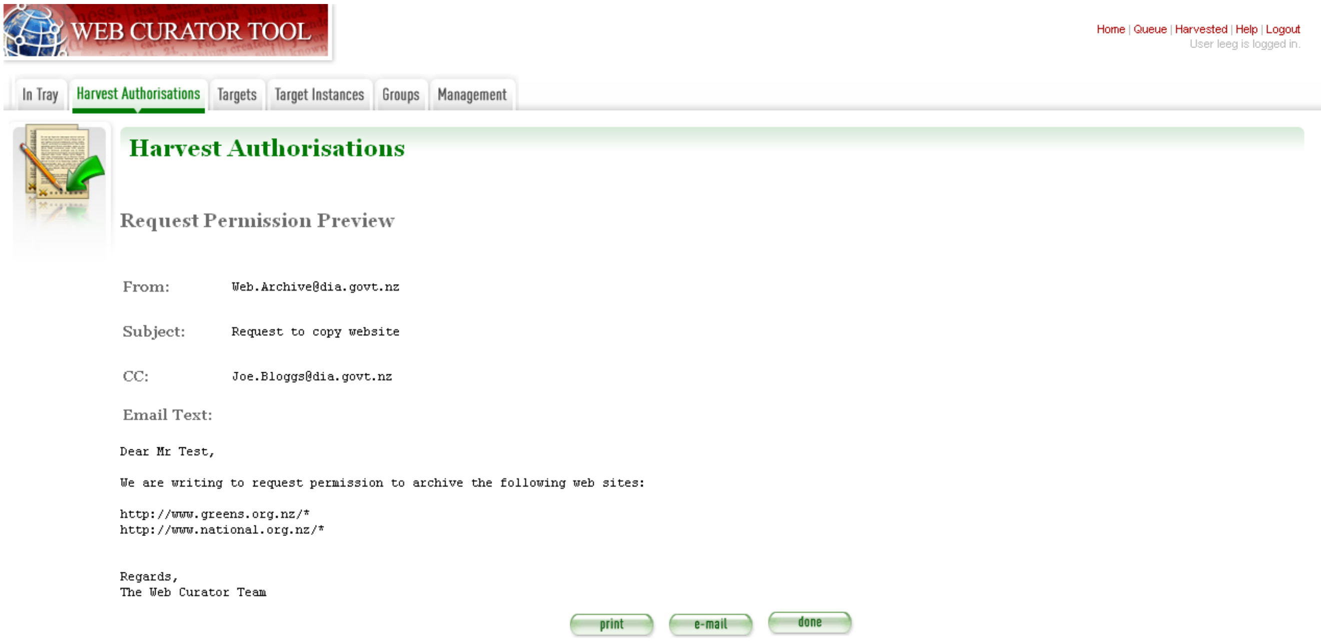
The harvest authorisation search page will be displayed.

After adding or editing a harvest authorisation record, you must save before clicking another main function tab (eg, Targets or Groups), or your changes will be lost.

3.4.7 How to send and/or print a permission request email

1. From the harvest authorisation search page, click ▤ next to the harvest authorisation request.
2. In the next screen choose the template from the dropdown list against the appropriate URL and click ▤

The system generates and displays the letter or Email template (depending on the template chosen)



From: Web.Archive@dia.govt.nz

Subject: Request to copy website

CC: Joe.Bloggs@dia.govt.nz

Email Text:

```
Dear Mr Test,

We are writing to request permission to archive the following web sites:

http://www.greens.org.nz/*
http://www.national.org.nz/*


Regards,
The Web Curator Team
```

Figure 9. Email Permission Request Letter

3. **Click to print or e-mail the letter to the agent.** (print-only templates will only allow you to print)

The system sends the letter and changes the permission status to **'requested'**.

4. Click **Done**.

The Harvest Authorisations search page redisplays.

3.4.8 How to view or update the status of a permission record

Once permission has been granted (or declined)

When you hear back from the authorising agent that you are authorised to harvest the website, follow steps 1 through 5 below to change the Status of the permission record to 'approved' (if permission is granted) or 'rejected' (if permission is declined).

The authorising agent may also specify special conditions, which should be recorded in the permission record at this point.

1. From the harvest authorisation search page, click ✏ next to the harvest authorisation request that includes the permission for which you sent the request letter.

The **General** *tab of the Create/Edit Harvest Authorisations page displays.*

16. Click the **Permissions** tab.

The Permissions tab displays.

17. Click 👁 (View) or ✎(Edit) next to the permission for which you sent the request letter.

The Create/Edit Permission page displays.

18. If editing, you can change the **Status** of the permission to 'approved' or 'rejected' as necessary, and click **Save**.
19. Click **Save** to close the Harvest Authorisation.

3.4.9 How to edit or view a harvest authorisation

Editing an existing authorisation is very similar to the process for creating a new record.

To start editing, go to the harvest authorisation search page, find the harvest authorisation you wish to edit, and click the

✎- **Edit** details

icon from the Actions column. This will load the harvest authorisation into the editor. Note that some users will not have access to edit some (or any) harvest authorisations.

An alternative to editing a harvest authorisation is to click the

👁 - **View** details

icon to open the harvest authorisation viewer. Data cannot be changed from within the viewer. Once in the harvest authorisation viewer you may also switch to the editor using the 'Edit' button

3.4.10 Legislative and other sources of authorisation

Some national libraries and other collecting institutions have a legislative mandate to harvest web material within their national jurisdiction, and do not need to request permission from individual copyright holders. In other cases, the library might rely on some other source of authority to harvest material, or may choose to harvest before permission is sought then seek permission retroactively.

The Web Curator Tool requires that every Seed URL be linked to a permission record. When a library is specifically authorised to perform harvests by legislation, this can seem like a source of inefficiency, as no "permission" is really required.

However, the Web Curator Tool still requires a harvest record, so that the ultimate source of harvest authority is always documented and auditable.

When the tool is configured correctly, there should be no overhead in most cases, and very little overhead in other cases.

This is possible through two mechanisms. First, the use of broad URL Patterns allows us to create a permission record that is almost always automatically assigned to Seed URLs without requiring any user action. Second, the "Quick Pick" option in permission records makes the permission record an option in the menu used to associate seeds with permission records.

In practical terms, this means institutions can set up a single harvest authorisation that applies to all their harvesting of their national internet. It should be set up as follows:

- **general information** should give the harvest authorisation a name that refers to the authorising legislation. For example:
  - Name: "NZ e-legal deposit"
  - Description: "All websites in the New Zealand domain acquired under legal deposit legislation"

- **url patterns** should identify as much of the national website as possible. For example:
  - http://*.nz/*
- **an authorising agency** should describe the government that provided the mandate to harvest. For example:
  - Name: "New Zealand Government"
  - Contact: "National Librarian"
  - Address: "National Library of New Zealand, Wellington"
- **a permission record** should link the authorising agency with the URL patterns, as for other permission records. Some points to note:
  - Dates: these fields should specify the date the legislation took (or takes) effect, and are typically open-ended.
  - Status: Approved.
  - Special restrictions / Access status: if your legislation places any restrictions on how the material may be harvested or access, record them here.
  - **Quick Pick**: Selected.
  - **Display Name**: The name used in the "Quick Pick" menu, such as "legal deposit legislation". The quick pick will show up in the seed tab of the Target record. See the Targets section for more information.

3.5 Targets

3.5.1 Introduction

In the Web Curator Tool, the portion of the web you have selected for harvesting is called a **Target**.

In the simplest cases, a Target is a website: a single conceptual entity that focuses on a particular topic or subject area, and which is hosted on a single internet address. However, many Targets are much more complicated (or much simpler) than this:

- A Target can be a single document, such as a PDF file
- A Target can be a part of a website, such as the Ministry of Education publications page, and all the PDF files it incorporates.
- A Target can be a website distributed across several different hosts, such as the Turbine website, whose front page is hosted at http://www.vuw.ac.nz/turbine, and whose content is hosted on www.nzetc.org.nz.
- A Target can be a collection of related websites, such as a set of political weblogs that provide discussion of a recent election.

  A Target can be an HTML serial issue located on a website
- A Target could be any combination of these.

A Target is often referred to as the **unit of selection**: if there is something desirable to harvest, archive, describe and make accessible, then it is a Target.

3.5.2 Terminology and status codes

Terminology

Important terms used with the Web Curator Tool include:

- **target** - a portion of the web you want to harvest, such as a web site or a set of web pages. Target includes crawler configuration details and a schedule of harvest dates.
- **seed** or **seed url** - a starting URL for a harvest, such as the root address of a website. A harvest usually starts with a seed and includes all pages “below” that seed.
- **approval** (of a target) - changing a Target into the **Approved** state. See the **How targets work** section below for an explanations of the implications of approval.
- **cancelled** (of a target) - changing a Target into the **Cancelled** state. This has the effect of deleting all scheduled Target Instances associated with the Target.

Target status

Each Target has a status:

- **pending** - a work in progress, not ready for approval
- **nominated** - completed and ready for approval
- **rejected** - rejected by the approver, usually because the Target was unsuitable or because it had an issue with permissions. You need to select a reason why a target was rejected.

  approved - complete and certified as ready for harvest
- **complete** -all scheduled harvests are complete
- **cancelled** - the Target was cancelled before all harvests were completed
- **reinstated** - the Target was reinstated from the complete, cancelled, or rejected state but is not yet ready for approval (equivalent to **pending**)

3.5.3 How targets work

Targets consist of several important elements, including a name and description for internal use; a set of Seed URLs, a web harvester profile that controls the behaviour of the web crawler during the harvest, one or more schedules that specify when the Target will be harvested, and (optionally) a set of descriptive metadata for the Target.

Seed URLS

The Seed URLs are a set of one or more URLs that form the starting point(s) for the harvest, and are used to define the scope of the harvest. For example, the Seed URL for the University of Canterbury website is http://www.canterbury.ac.nz/ and (by implication) the website includes all the other pages on that server.

Each Seed URL must be linked to at least one current, approved permission record before any harvests can proceed for the Target.

Schedules

A Schedule is added to a Target to specify when (and how often) the Target will be harvested. For example, you may want a Target to be harvested every Monday at midnight, or on the first of every month at 5AM, or every day at Noon for the next two weeks. Alternatively, you can request that a Target be harvested only once, as soon as possible. Multiple schedules can be added to each Target.

Nomination

After a Target has been created, has its Seed URLs added, has a schedule attached, and has all the other necessary information set, it is changed into the Nominated state. This indicates that the owner believes the Target is ready to be harvested.

Approval

A nominated Target must be **Approved** before any harvests will be performed.

Approving a Target is an action that is usually reserved for senior users, as it has several implications and consequences. First, approving a Target is a formal act of selection: the Approver is saying that the Target is a resource that the Library wishes to collect. Second, approving a Target is an act of verification: the Approver is confirming that the Target is correctly configured, that its schedule is appropriate, and that its permissions do authorise the scope and frequency of the scheduled harvests. Finally, approving a Target as a functional aspect: it tells the Web Curator Tool to add the scheduled harvests to the Harvest Queue.

Completion, Cancellation, and Reinstatement

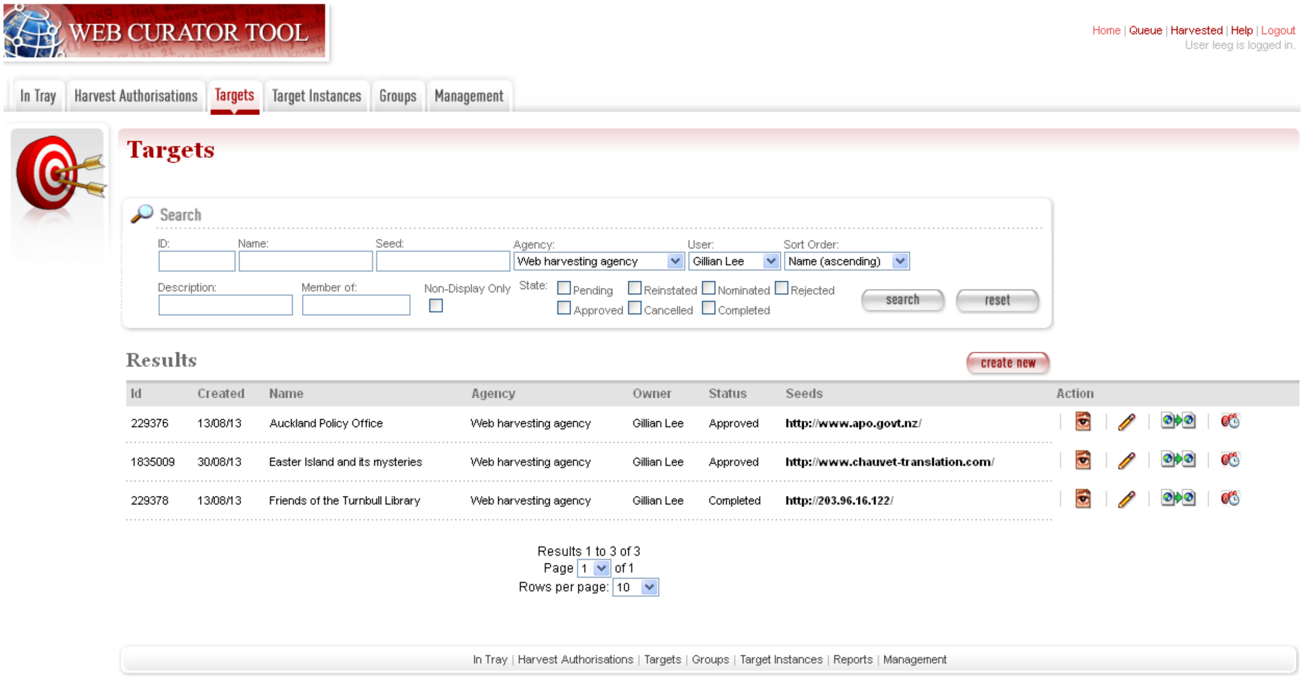
When all the harvests scheduled for a Target have finished, the Target automatically changes from the Approved state to the Completed state.

Sometime a user will change the state of an Approved Target to Cancelled before all the harvests are complete. This means that all scheduled harvests will be deleted.

Some users will have access to change a Completed or Cancelled Target to the Reinstated state, at which point they can edit the Target (for example, attaching a new schedule) and nominate it for harvest again.

3.5.4 Target search page

You manage Targets from the **Target search** page:



ID:
Name:
Seed:
Agency: Web harvesting agency
User: Gillian Lee
Sort Order: Name (ascending)
Description:
Member of:
Non-Display Only □
State: □ Pending □ Reinstated □ Nominated □ Rejected □ Approved □ Cancelled □ Completed

Id	Created	Name	Agency	Owner	Status	Seeds	Action
229376	13/08/13	Auckland Policy Office	Web harvesting agency	Gillian Lee	Approved	http://www.apo.govt.nz/	
1835009	30/08/13	Easter Island and its mysteries	Web harvesting agency	Gillian Lee	Approved	http://www.chauvet-translation.com/	
229378	13/08/13	Friends of the Turnbull Library	Web harvesting agency	Gillian Lee	Completed	http://203.96.16.122/	

Results 1 to 3 of 3
Page 1 of 1
Rows per page: 10

Figure 10. Target search page

At the top of the page are:

- fields to search for existing targets by **ID, Name**, **Seed URL**, **Agency**, **User**, **Sort Order, Description, Member of, Non-Display Only and State**
- The search panel contains a drop down list allowing the user to control the sort order of the search results. E.g. 'Most recent first' will display the targets with the most recently created target listed first.
- The Description field allows you to search for information found in the target description field
- The Member of field allows you to search for targets found in a particular Group.
- Non-Display allows you to search for targets that are ticked as non-display in the Target Access tab
- a button to **create new** Targets

You can enter search terms in any or all of the textboxes and menus, and select any number of states. All the text boxes contain simple text strings, except for Seed (URLs) and ID (Target ID numbers).

Search criteria will be combined as an AND query and the matching records retrieved. The default search is for Targets that you own.

Searches in text boxes are case-insensitive, and match against the prefix of the value. For example, a search for "computer" in the name field might return Targets named "Computer warehouse" and "Computerworld", but not "Fred's computer".

You can perform wildcard characters to perform more complex text matches. The percent (%) character can be used to match zero or more letters, and the underscore (_) to match one character. So, for example, a search for "%computer" would match "Computer warehouse" and "Computerworld" and "Fred's computer"

Below that are search results, with options to:

- **View** the Target

- **Edit** the Target

- **Copy** the Target and create a new one

- **View** the Target Instances derived from this Target

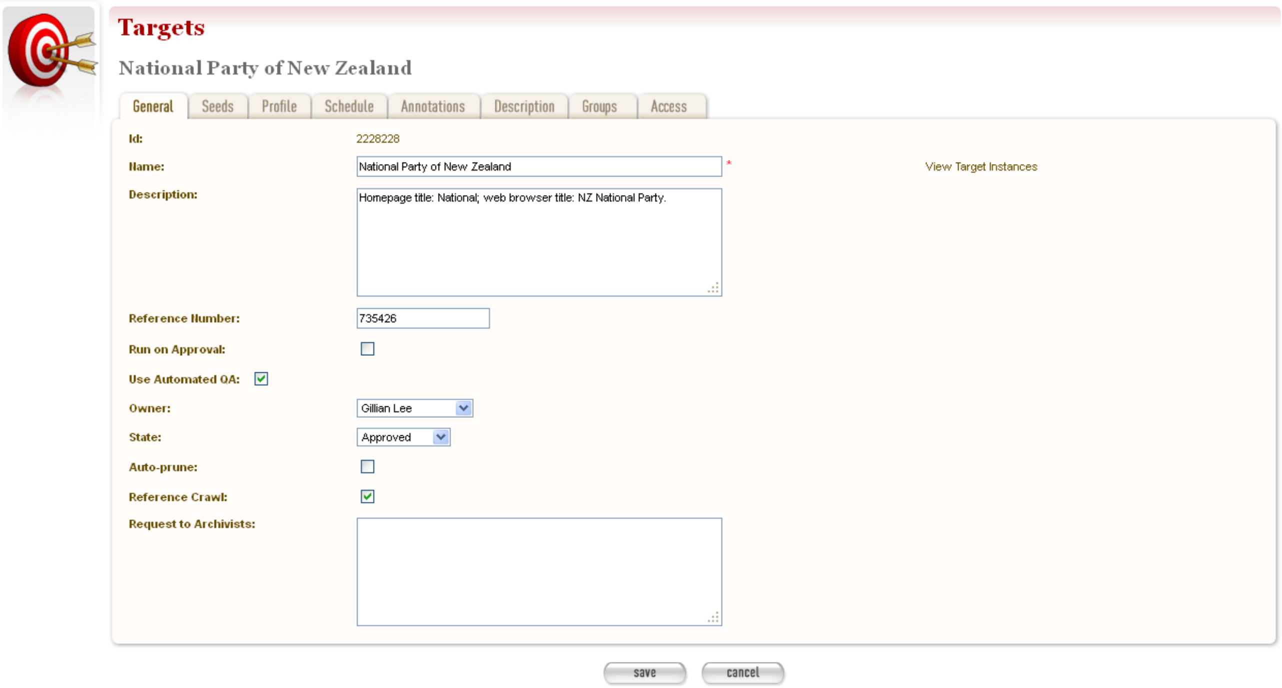
- **Delete** the Target. This action can only be done when the target is in the pending state

3.5.5 How to create a target

From the Targets page,

1. Click **create new**.

The **Create/Edit Targets** *page displays.*



Id: 2228228

Name: National Party of New Zealand *

Description: Homepage title: National; web browser title: NZ National Party.

Reference Number: 735426

Run on Approval: □

Use Automated QA: ☑

Owner: Gillian Lee

State: Approved

Auto-prune: □

Reference Crawl: ☑

Request to Archivists:

Figure 11. Create/Edit Targets

The **Create/Edit Targets** page includes several tabs for adding or editing information about Targets:

- **General** - general information about the Target, such as a name, description, owner, and status
- **Seeds** - base URLs for websites to harvest
- **Profile** - technical instructions on how to harvest the Target
- **Schedule** - dates and times to perform the harvest
- **Annotations** - notes about the Target
- **Description** - metadata about the Target
- **Access** - settings regarding access to the harvested Target

Enter general information about the target

2. On the **General** tab, enter basic information about the Target. When editing an existing Target, a ‘View Target Instances’ link is displayed to the right of the ‘Name’ field. Clicking this link displays the Target Instances screen with all Target Instances matching the Target name.
3. Reference number is optional. e.g. The National Library of New Zealand adds the catalogue record number here and their WCT system is configured so that no website can be archived into their National Digital Heritage Archive without this number being present in the target record.
4. ‘**Run on approval**’ If you check this box you can prepare the target record so that the harvest is ready to run once you set the Harvest Authorisation permissions form to “Approved”. To do this approve the target itself, add the seed URL and pending permission and schedule as instructed below.

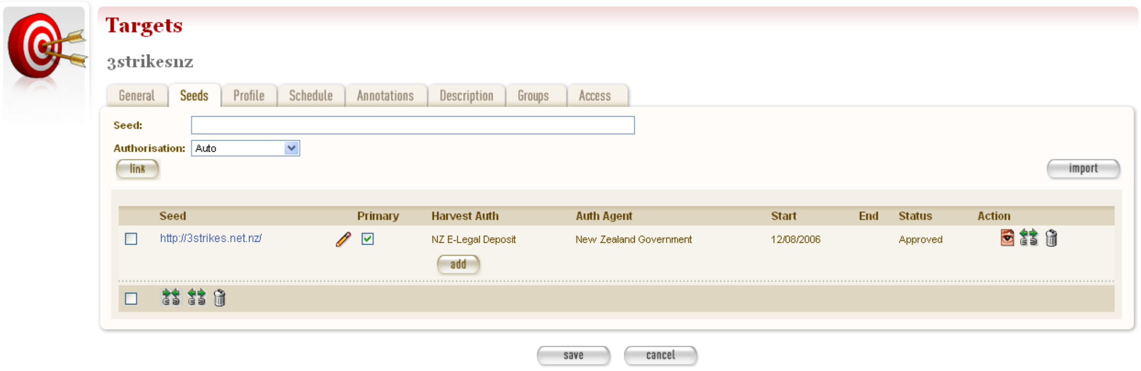
   NB. ‘Run on approval’ sets an immediate harvest one minute into the future, but until the harvest authorisation is approved the harvest itself will keep deferring 24 hours until the harvest authorisation is set to approved.

5. Enabling the **Auto-prune** checkbox causes WCT to identify pruned items from the last archived harvest and prunes those items from subsequent harvests.
6. **Note to Archivists** - An optional note.

The Required fields are marked with a red star. When the form is submitted, the system will validate your entries and let you know if you leave out any required information.

Enter the sites you want to harvest

7. Click the **Seeds** tab.
8. The **Seeds** tab includes a box for adding the base URL of each web site you want to harvest and list of previously added seeds.



Seed:

Authorisation: Auto

	Seed		Primary	Harvest Auth	Auth Agent	Start	End	Status	Action
□	http://3strikes.net.nz/	✎	☑	NZ E-Legal Deposit	New Zealand Government	12/08/2006		Approved	

Figure 12. Seeds tab

9. Enter the root URL of a website for this Target.
10. Select a permission record (or records) that **authorise** you to harvest the seed:

- **Auto** will automatically find all permission records whose URL Patterns match the seed.
- **Add Later** enters the seed without to any permissions (the Target cannot be Approved until a permission is added).
- **Quick Picks**. See the harvest authorisation section for directions on how to create these.
- **NB.** If your seed URL doesn't match the seed URL pattern in the permission record you want to use (e.g. a '.com' site that is in scope for Legal Deposit) it will still run when you link it to the approved Harvest Authorisation.

11. Click **link**. Repeat for additional sites.

The seed displays in the list below.

You can also use the **Import** *button to import a precompiled list of seeds from a text file. The text file should have one URL per line.*

The multiple selection bar at the bottom of the list allows you to link, unlink and delete multiple selected seeds.

You can edit the seed URL after it has been linked. Click on the edit icon ✎ , make the changes, and then click on the save icon 💾.

Select a profile and any overrides

12. Click the **Profile** tab.

The Profile tab includes a list of harvest profiles, and a series of options to override them. Generally, the default settings are fine. There are two types of harvest profiles to choose from:

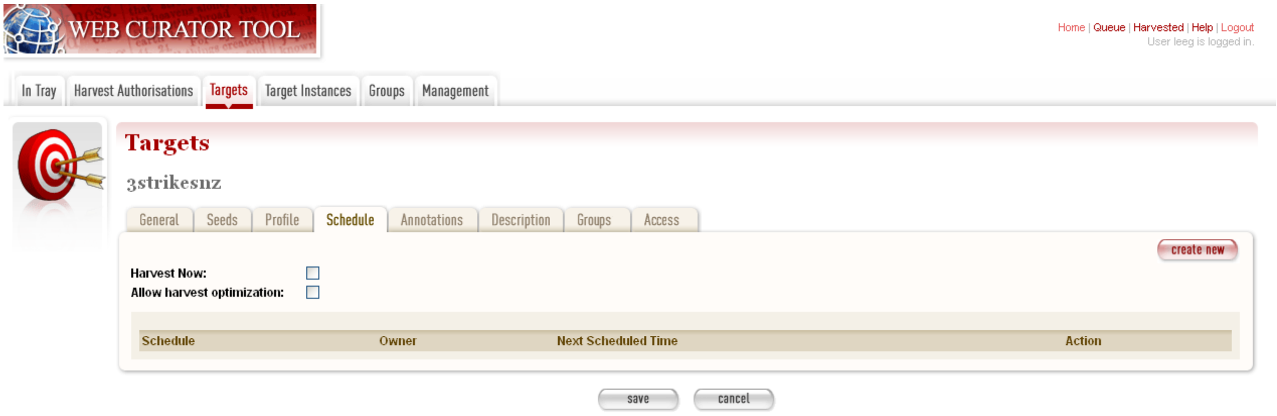
- Heritrix 1
- Heritrix 3

See the Target Instance Quality Review section for further information about overriding profiles.

Enter a schedule for the target

13. Click the **Schedule** tab.

The **Schedule** *tab includes a list of schedules and a button to create a new schedule.*



Harvest Now: □

Allow harvest optimization: □

Schedule	Owner	Next Scheduled Time	Action

Figure 13. Schedule tab

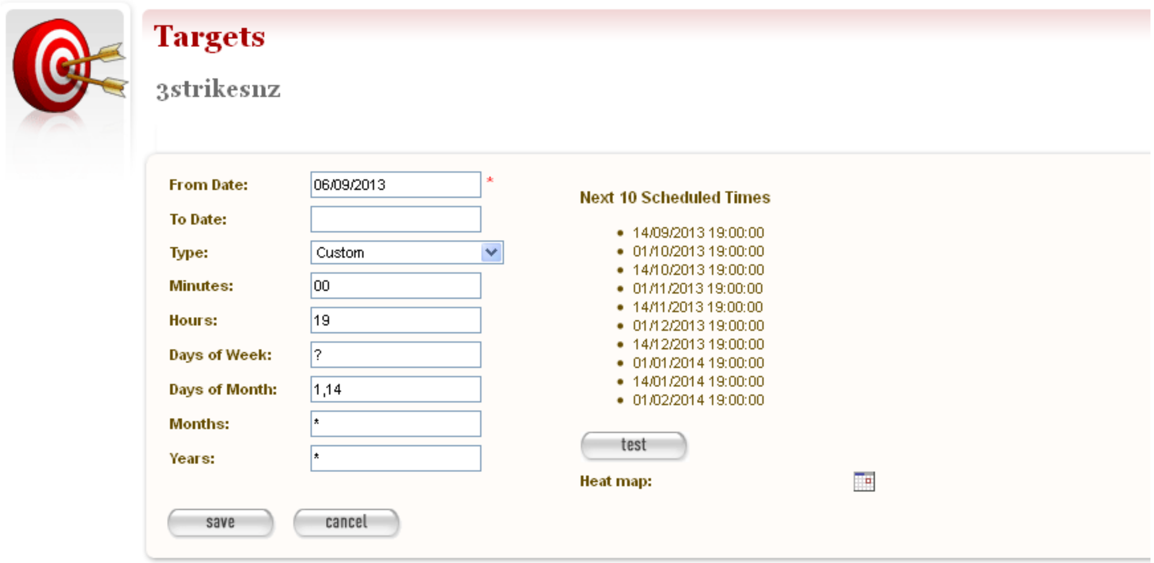
14. **Harvest now** - ticking this box will schedule a one off harvest 5 minutes after saving the record.

    NB: If you click on 'harvest now' and the target is in the completed state you will now a prompt to inform you that it's possible if you have the authority to do so. The National Library of New Zealand also uses WCT to harvest HTML serials (as a separate agency). They don't use schedules and they don't want to reinstate a target in the completed state and have to approve the target every time a new serial issue is harvested.

15. **Harvest optimization**. See the Management section for information about setting this up.

16. Click **create new**.

The **Create/Edit Schedule** *page displays fields for entering a schedule.*



Targets

3strikesnz

From Date:	06/09/2013 *
To Date:	
Type:	Custom
Minutes:	00
Hours:	19
Days of Week:	?
Days of Month:	1,14
Months:	*
Years:	*

Next 10 Scheduled Times

- 14/09/2013 19:00:00
- 01/10/2013 19:00:00
- 14/10/2013 19:00:00
- 01/11/2013 19:00:00
- 14/11/2013 19:00:00
- 01/12/2013 19:00:00
- 14/12/2013 19:00:00
- 01/01/2014 19:00:00
- 14/01/2014 19:00:00
- 01/02/2014 19:00:00

Figure 14. Create/Edit Schedule

17. Enter **From** and **To** dates for when the harvest will run; select a **Type** of schedule, e.g. 'Every Monday at 9:00pm' or 'Custom'
18. If you select 'Custom', enter details of the schedule; and click **Save**. Figure 14 shows a fortnightly schedule. A two-yearly schedule can be set up in **Years** e.g. 2013/2 means the next scheduled harvest would be 2015.

    The scheduling uses Cron expressions. For more information about how to use these expressions go to: http://en.wikipedia.org/wiki/Cron
19. The **Heat map** pop up displays a calendar indicating the level of harvesting scheduled for each day, so you can schedule harvests on less busy days if required. The thresholds and colour coding can be set in the Harvester Configuration under the Management section.

Annotations

20. Click the **Annotations** tab.
21. The **Annotations** tab allows you to record internal and selection information about the Target. The Annotations are intended for internal use, but are included in submissions to archives.
22. Annotations can be modified or deleted after creation by the user who created them. When an annotation is modified, the annotation date is automatically updated to the time of modification.

Description

23. Click the **Description** tab.

The **Description** *tab includes a set of fields for storing Dublin Core metadata. This not used in the Web Curator Tool, but is included when any harvests are submitted to a digital archive.*

Groups

24. Click the **Groups** tab.

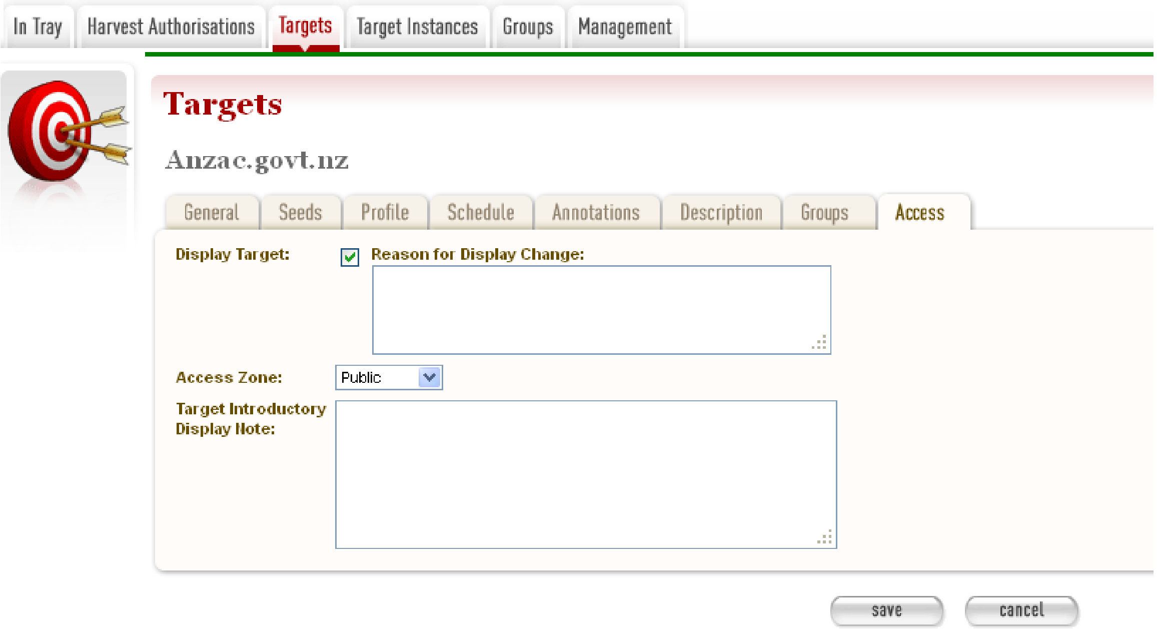
The **Groups** *tab allows you to add Targets to Web Curator Tool groups, such as collections, events or subjects. See the chapter on Groups for more information.*

Access

25. Click the **Access** tab.

The **Access** *tab allows you to specify a Display Target flag, Display Notes and an Access Zone from*

- *Public(default)*
- *Onsite*
- *Restricted*



Display Target: ☑ Reason for Display Change:

Access Zone: Public

Target Introductory Display Note:

Figure 15. Access Tab

The ‘Reason for Display Change’ text field allows the user to record why the Display Target flag was set or unset.

Save the completed target

26. Click **save** at the bottom of the page to save the target.

   You should pay close attention to the State the Target is saved in. When you are creating a new record, it will be saved in the ‘Pending’ state.

3.5.6 How to edit or view a target

Editing an existing target is very similar to the process for creating a new record.

To start editing, go to the Target search page, and click the

✏ - **Edit** details

icon from the Actions column. This will load the relevant Target editor. Note that some users will not have access to edit some (or any) Targets.

An alternative to editing a Target is to click the

- **View** details

icon to open the Target viewer. Targets cannot be changed from within the viewer. Once in the Target viewer you may also switch to the editor using the 'Edit' button

3.5.7 How to nominate and approve a target

When you are creating a new record, it will be saved in the 'Pending' state. This means that the Target is a work in progress, and not ready for harvesting.

When the record is complete, you should **nominate** it for harvesting. This signals to the other editors that your target is ready for Approval.

An editor who has permission to approve targets will then review the Target and make sure it is entirely correct, that it has the right Seed URLs, that its permissions are present and correct, and that its schedule is appropriately configured. They will then **approve** the Target (which means that Target Instances will be created and harvests will proceed).

Nominating

1. Open the Target in Edit mode.

   The **General** *tab will be displayed, and the* **State** *of the Target will be set to* **Pending**.

2. Change the state to **Nominated**.
3. Click **save** at the bottom of the page to save the Target.

Approval

4. Open the Target in Edit mode.

   The **General** *tab will be displayed, and the* **state** *of the Target will be set to* **Nominated**.

5. Change the state to **Approved**.
6. Click **save** at the bottom of the page to save the Target.

   A set of Target Instances representing harvests of the Target will be created.

   Users with permission to Approve Targets will be able to set the state of a new target to Approved without going through the Nominated state.

3.5.8 How to delete or cancel a target

Targets can be deleted, but only if they have no attached Target Instances.

However, once a Target Instance enters the Running (or Queued) state, it can no longer be deleted from the system. In other words, a Target cannot be deleted if it has been harvested (even if that harvest was unsuccessful). This restriction is necessary so that the Web Curator Tool retains a record of all the harvests attempted in the tool in case it is needed later for audit purposes.

Targets that are no longer required should be left in the **Cancelled** state. Targets whose scheduled harvests have all been completed will be changed to the **Completed** state. Both cancelled and completed targets can be changed to the **Reinstated** state and re-used.

Targets can be set to a **Rejected** state and in this case the tool allows the user to nominate a reason for the rejection from a drop down list whose contents are defined by system administrators using the administration screen for Rejection Reasons.

3.6 Target Instances and Scheduling

3.6.1 Introduction

Target Instances are individual harvests that are scheduled to happen, or that are currently in progress, or that have already finished. They are created automatically when a Target is **Approved**.

For example, a target might specify that a particular website should be harvested every Monday at 9pm. When the target is Approved, a Target Instance is created representing the harvest run at 9pm on Monday 24 July 2006, and other Target Instances are created for each subsequent Monday.

3.6.2 Terminology and status codes

Terminology

Important terms used with the Web Curator Tool include:

- **target instance** - a single harvest of a Target that is scheduled to occur (or which has already occurred) at a specific date and time.
- **Queue or harvest queue** - the sequence of future harvests that are scheduled to be performed.
- **harvest** - the process of crawling the web and retrieving specific web pages.
- **harvest result** - the files that are retrieved during a **harvest**.
- **quality review** - the process of manually checking a **harvest result** to se if it is of sufficient quality to archive.

Target instance status

Each Target Instance has a status:

- **scheduled** - waiting for the scheduled harvest date and time.
- **queued** - the scheduled start time has passed, but the harvest cannot be run immediately because there are no slots available on the harvest agents, or there is not enough bandwidth available.
- **running** - in the process of harvesting.
- **stopping** - harvesting is finished and the harvest result is being copied to the digital asset store (this is a sub-state of **running**).
- **paused** - paused during harvesting.
- **aborted** - the harvest was manually aborted, deleting any collected data.
- **harvested** - completed or stopped; data collected is available for review
- **endorsed** - harvested data reviewed and deemed suitable for archiving

- **rejected** - harvested data reviewed and found not suitable for archiving (ie, content is incomplete or not required)
- **archiving** - in the process of submitting a harvest to the archive (this is a sub-state of **archived**).
- **archived** - harvested content submitted to the archive.

3.6.3 How target instances work

Target Instances are created when a Target is approved.

Scheduling and Harvesting

Target Instances are always created in the **scheduled** state, and always have a Scheduled Harvest Date.

The scheduled Target Instances are kept in the Harvest Queue. Examining this queue (by clicking on the **queue** button on the homepage) gives you a good overview of the current state of the system and what scheduled harvests are coming up next.

When the scheduled start time arrives for a scheduled Target Instance, the Web Curator Tool makes a final check that the permission records for this harvest are valid. If the Target Instance is appropriately authorised, the harvest is started and the state of the Target Instance changes to **Running**.

When the harvest is complete, the Harvest Result is ready for quality review, and the Target Instance state is changed to **Harvested**.

Quality Review

When a harvest finishes, the Web Curator Tool notifies its owner, who has to Quality Review the harvest result to verify that the harvest was successful and that it downloaded all the necessary parts of the website.

Several tools are provided for supporting the quality review function, these are described in detail in the next chapter.

When the Target Instance owner has finished reviewing a harvest result, they must decide whether it is of acceptable quality for the digital archive. If it fails this test, the user marks the Target Instance as **rejected**, and the harvest result is deleted. No further action can be performed on the Target Instance, though the user can attempt to make adjustments to the scope of the Target in order to get a better result the next item it is harvested.

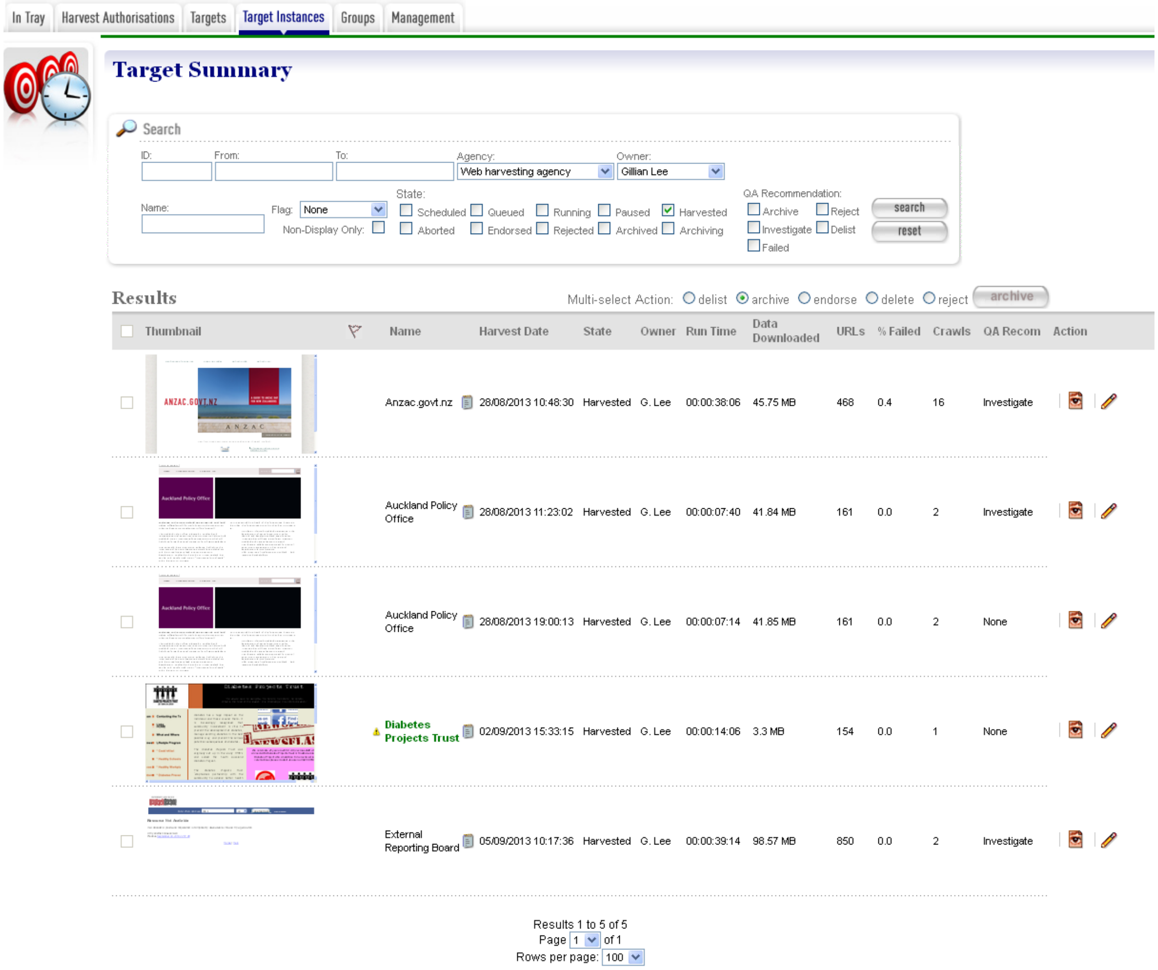
If the harvest result is successful, the user can **endorse** it to indicate that it is ready for inclusion in the digital archive.

Submitting a Harvest to the Digital Archive

Once a Target Instance has been Endorsed, it can be **submitted** to the archive for long-term storage and subsequent access by users. At this point, the harvest result leaves the control of the Web Curator Tool, and becomes the responsibility of the archive. The harvest result will eventually be deleted from the Web Curator Tool, but metadata about the Target Instance will be permanently retained.

3.6.4 Target instance page

You manage Target Instances from the **Target Instance page**:



Target Summary

Search

ID: From: To: Agency: Web harvesting agency Owner: Gillian Lee

Name: Flag: None Non-Display Only: □

State: □ Scheduled □ Queued □ Running □ Paused ☑ Harvested □ Aborted □ Endorsed □ Rejected □ Archived □ Archiving

QA Recommendation: □ Archive □ Reject □ Investigate □ Delist □ Failed

search reset

Results

Multi-select Action: ○ delist ● archive ○ endorse ○ delete ○ reject archive

□	Thumbnail		Name		Harvest Date	State	Owner	Run Time	Data Downloaded	URLs	% Failed	Crawls	QA Recom	Action
□			Anzac.govt.nz		28/08/2013 10:48:30	Harvested	G. Lee	00:00:38:06	45.75 MB	468	0.4	16	Investigate	
□			Auckland Policy Office		28/08/2013 11:23:02	Harvested	G. Lee	00:00:07:40	41.84 MB	161	0.0	2	Investigate	
□			Auckland Policy Office		28/08/2013 19:00:13	Harvested	G. Lee	00:00:07:14	41.85 MB	161	0.0	2	None	
□		⚠	Diabetes Projects Trust		02/09/2013 15:33:15	Harvested	G. Lee	00:00:14:06	3.3 MB	154	0.0	1	None	
□			External Reporting Board		05/09/2013 10:17:36	Harvested	G. Lee	00:00:39:14	98.57 MB	850	0.0	2	Investigate	

Results 1 to 5 of 5
Page 1 of 1
Rows per page: 100

Figure 16. Target Instances

NB: the homepage images are pointing to the live site. WCT is configured so that you can switch off this functionality if this slows your system's performance.

At the top of the page are fields to search for existing target instances by **ID, start date** (**From**, **To**), **Agency**, **Owner**, Target **Name, Flagged** Target Instances and **State** and **QA Recommendation**.

> The search page remembers your last search and repeats it as the default search, with two exceptions. If you navigate to the Target Instance search page by clicking the "open" button on the homepage, it will show all the Target Instances that you own. And if you navigate to the page by clicking the "Queue" button on the homepage, or the "Queue" link at the top right of any page, it will show the Target Instances that make up the current harvest queue. If you navigate to the Target Instance search page by clicking the "harvested" button on the homepage, it will show all the Target Instances that you own that are in the 'Harvested' state, and if you navigate to the Target Instance search page from the Target General tab by clicking the "View Target Instances" link, it will show all the Target Instances that match the Target name. Once in the Target Instance viewer you may also switch to the editor using the 'Edit' button

The search results are listed at the bottom of the page. For each, you may have these options, depending on its state and your permissions:

- **View** the Target Instance

- **Edit** the Target Instance

🗑 - **Delete** a scheduled or queued Target Instance

- **Harvest** a scheduled Target Instance immediately

- **Pause** a running Target Instance

- **Stop** a running Target Instance and save its patrial harvest result

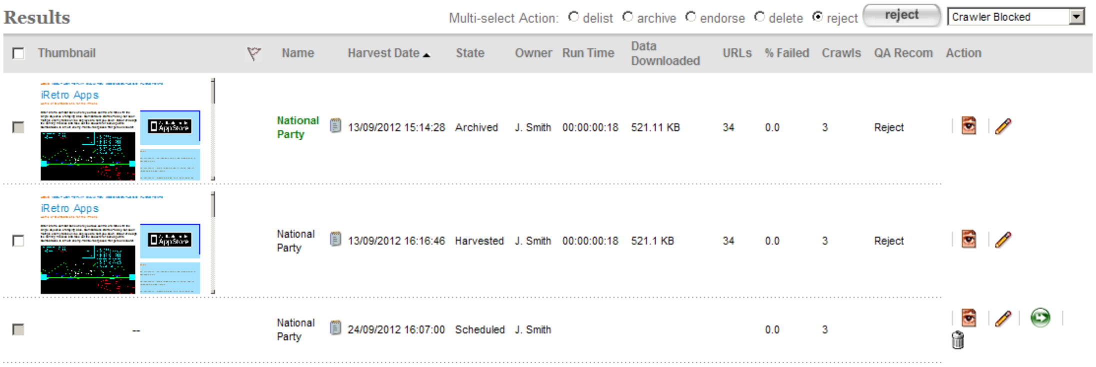
- **Abort** a running Target Instance and delete its harvest result

H3 - **H3 Script Console** for executing scripts against Heritrix 3 Target Instances

- **Target Annotation**: displays any annotations defined for this target instance's target.

Operations on multiple target instances can be performed using the **Multi-select Action** radio button. Note that the target instance checkbox will be enabled only for those target instances in a valid state for the selected multi-select action:

- **delist**: cancels all future schedules for the selected target instances.
- **endorse**: endorses the selected target instances.
- **archive**: archives the selected target instances.
- **delete**: deletes all selected target instances in a valid state (eg: scheduled target instances).
- **reject**: when selected, a rejection reason drop-down box is displayed and clicking the action button will reject the selected target instances with the selected rejection reason:



Results

Multi-select Action: ○ delist ○ archive ○ endorse ○ delete ● reject [reject] [Crawler Blocked ▾]

☐	Thumbnail		Name	Harvest Date ▲	State	Owner	Run Time	Data Downloaded	URLs	% Failed	Crawls	QA Recom	Action
☐	iRetro Apps		National Party	13/09/2012 15:14:28	Archived	J. Smith	00:00:00:18	521.11 KB	34	0.0	3	Reject	
☐	iRetro Apps		National Party	13/09/2012 16:16:46	Harvested	J. Smith	00:00:00:18	521.1 KB	34	0.0	3	Reject	
☐	--		National Party	24/09/2012 16:07:00	Scheduled	J. Smith				0.0	3		🗑

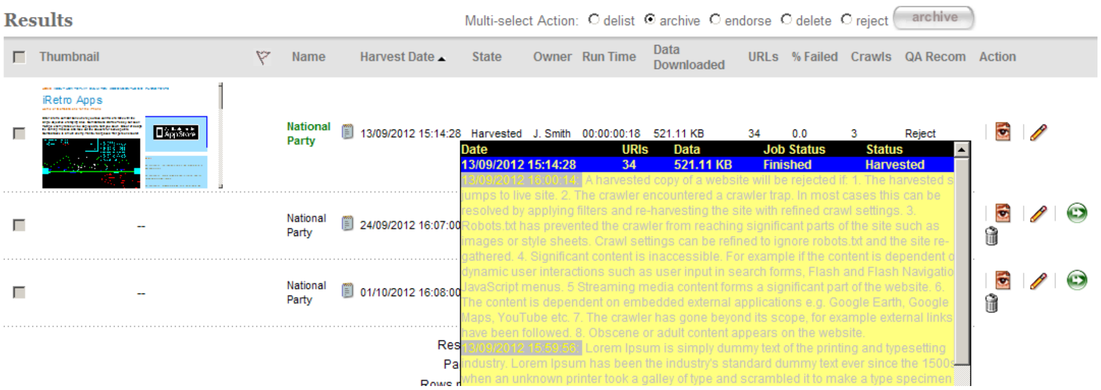
Figure 17. Rejecting a target instance

Sortable fields:

Harvest Date ▼ Clicking on the **Name**, **Harvest Date**, **State**, **Run Time**, **URLs**, **% Failed** or **Crawls** columns will sort the search results by that column.

Harvest Date ▲ Clicking the same column again will perform a reverse sort of the column

QA Recom Hovering over the QA Recommendation will display a list of the three most recent harvest status and any annotations for the target instance:



Results

Multi-select Action: ○ delist ● archive ○ endorse ○ delete ○ reject [archive]

□	Thumbnail	Name	Harvest Date ▲	State	Owner	Run Time	Data Downloaded	URLs	% Failed	Crawls	QA Recom	Action
□	iRetro Apps	National Party	13/09/2012 15:14:28	Harvested	J. Smith	00:00:00:18	521.11 KB	34	0.0	3	Reject	
□	--	National Party	24/09/2012 16:07:00									
□	--	National Party	01/10/2012 16:08:00									

Date	URIs	Data	Job Status	Status
13/09/2012 15:14:28	34	521.11 KB	Finished	Harvested

13/09/2012 16:00:14: A harvested copy of a website will be rejected if: 1. The harvested s
jumps to live site. 2. The crawler encountered a crawler trap. In most cases this can be resolved by applying filters and re-harvesting the site with refined crawl settings. 3. Robots.txt has prevented the crawler from reaching significant parts of the site such as images or style sheets. Crawl settings can be refined to ignore robots.txt and the site re-gathered. 4. Significant content is inaccessible. For example if the content is dependent o
dynamic user interactions such as user input in search forms, Flash and Flash Navigatio
JavaScript menus. 5 Streaming media content forms a significant part of the website. 6. The content is dependent on embedded external applications e.g. Google Earth, Google Maps, YouTube etc. 7. The crawler has gone beyond its scope, for example external links have been followed. 8. Obscene or adult content appears on the website.

13/09/2012 15:59:56: Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s
when an unknown printer took a galley of type and scrambled it to make a type specimen

Figure 18. Sortable fields

3.6.5 Scheduling and the harvest queue

Target Instance Creation

Target Instances are created when a Target is **approved**. They are always created in the **scheduled** state, and always have a Scheduled Harvest Date (which is actually a date and time).

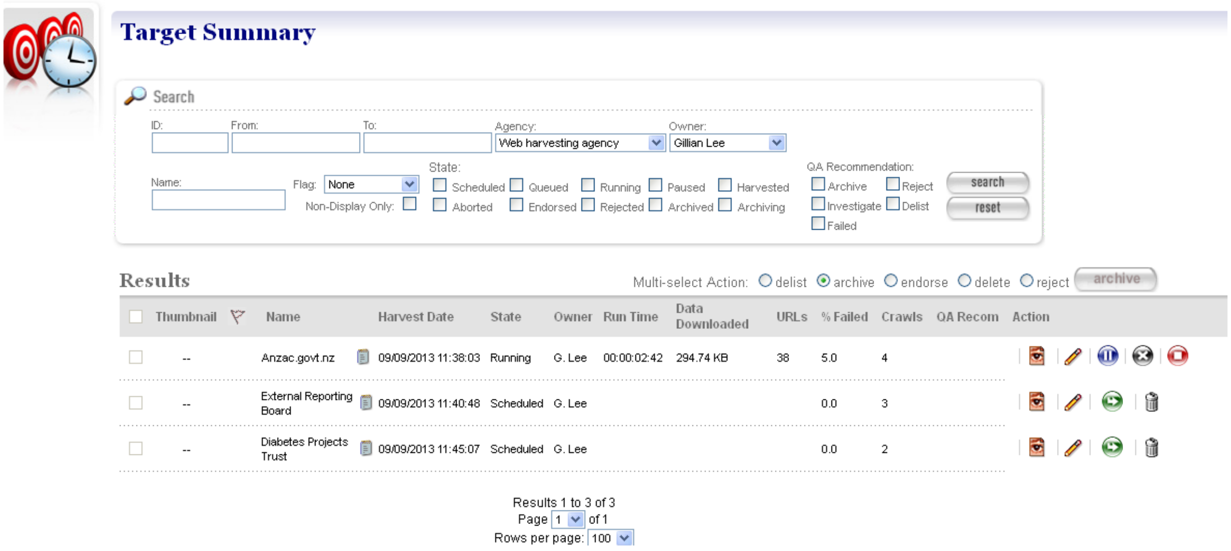
The Target Instances are created in accordance with the Target's Schedule (or Schedules). Target Instances will be created three months in advance of their scheduled harvest date (this period is configurable), and the first Target Instance is always scheduled (even if it is outside the three month window).

If the **Run on Approval** box is checked on the General Tab of the Target, then an additional Target Instance will be created with a Scheduled Harvest Date one minute in the future.

Examining the Harvest Queue

The Scheduled Target Instances are kept in the Harvest Queue. You can view the queue by clicking on the **queue** button on the homepage. It gives you a good overview of the current state of the system and what scheduled harvests are coming up next.

The queue view is shown in the figure below.



Target Summary

Search

ID: [ ] From: [ ] To: [ ] Agency: [Web harvesting agency] Owner: [Gillian Lee]

Name: [ ] Flag: [None] Non-Display Only: □

State: □ Scheduled □ Queued □ Running □ Paused □ Harvested □ Aborted □ Endorsed □ Rejected □ Archived □ Archiving

QA Recommendation: □ Archive □ Reject □ Investigate □ Delist □ Failed

[search] [reset]

Results

Multi-select Action: ○ delist ● archive ○ endorse ○ delete ○ reject [archive]

□	Thumbnail	Name	Harvest Date	State	Owner	Run Time	Data Downloaded	URLs	% Failed	Crawls	QA Recom	Action
□	--	Anzac.govt.nz	09/09/2013 11:38:03	Running	G. Lee	00:00:02:42	294.74 KB	38	5.0	4		
□	--	External Reporting Board	09/09/2013 11:40:48	Scheduled	G. Lee				0.0	3		
□	--	Diabetes Projects Trust	09/09/2013 11:45:07	Scheduled	G. Lee				0.0	2		

Results 1 to 3 of 3
Page [1] of 1
Rows per page: [100]

Figure 19. Harvest queue

The queue view is actually just a predefined search for all the Target Instances that are Running, Paused, Queued (i.e. postponed), or Scheduled.

Running a Harvest

When the scheduled start time arrives for a Scheduled Target Instance, the Web Curator Tool makes final checks that the permission records for this harvest are valid. If the harvest is appropriately authorised, then the Web Curator Tool will normally allocate it to one of the Harvest Agents, which invokes the Heritrix web crawler to harvest the site (as directed by the profile tab in the Target). For example, if a Target Instance is assigned to a Heritrix 3 profile, then it will be allocated to a Heritrix 3 Harvest Agent. The Target Instance State will be updated to **running**.

Some users may have the option of using the ▶ - '**Harvest** a Scheduled Target Instance immediately' icon to launch the harvest before its Scheduled Start Date arrives.

Queued Target Instances

Sometimes a harvest cannot be run because there is no capacity on the system: either the maximum number of harvests are already running, or there is no spare bandwidth available for an additional harvest.

In these cases, the Target Instance cannot be sent to the Harvest Agents. Instead, their state is updated to **queued**, and they remain in the Harvest Queue. The harvest is run as soon as capacity becomes available on a Harvest Agent.

Deferring Target Instances

Sometimes a Target Instance is scheduled to run, but the Target it is based on has one or more permission records attached that are still in the pending state. In other words, permission has not (yet) been granted for this harvest.

In this situation, the Scheduled Start Date of the Target instance is moved forward by 24 hours (its state remains scheduled). At the same time, a notification is sent to the Target Instance owner to tell them the harvest has been **deferred**.

Deleting Target Instances

Only Target Instances in the Scheduled or Queued states can be deleted. A Target Instance in the Queued state may only be deleted if it has not yet begun to harvest. Queued Target Instances that have previously begun to harvest but have returned to the Queued state may not be deleted.

Once a Target Instances enters the Running state, it can no longer be removed from the system. This means we retain information about every crawl attempted by the Web Curator Tool in case we need it later for audit purposes.

A Scheduled Target Instance that is deleted will not be run.

> *When the state of a Target changes from Approved to any other state, then all its Scheduled Target Instances will be immediately deleted.*

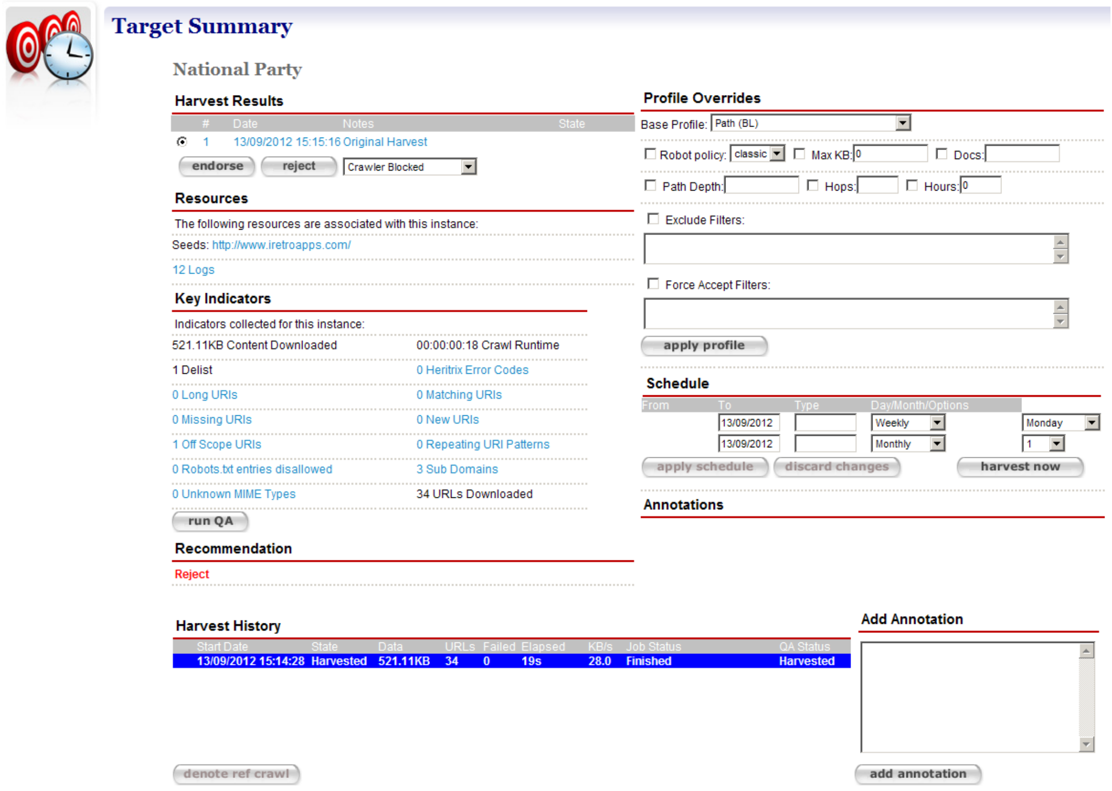
Harvested Target Instances

When the harvest is complete, the Harvest Result is transferred to the digital asset store, and the Target Instance state is changed to **Harvested**. At this point, it is no longer part of the Harvest Queue.

3.6.6 To review target instances:

1. Click the name of the target instance to view the target instance summary page.

   The summary page is composed of panels that provide access to the QA Indicators and Recommendation, and draws together existing functionality into a single location.



Target Summary

National Party

Harvest Results

	#	Date	Notes	State
●	1	13/09/2012 15:15:16	Original Harvest	

Resources

The following resources are associated with this instance:

Seeds: http://www.iretroapps.com/

12 Logs

Key Indicators

Indicators collected for this instance:

521.11KB Content Downloaded	00:00:00:18 Crawl Runtime
1 Delist	0 Heritrix Error Codes
0 Long URIs	0 Matching URIs
0 Missing URIs	0 New URIs
1 Off Scope URIs	0 Repeating URI Patterns
0 Robots.txt entries disallowed	3 Sub Domains
0 Unknown MIME Types	34 URLs Downloaded

Recommendation

Reject

Profile Overrides

Base Profile: Path (BL)

□ Robot policy: classic □ Max KB: 0 □ Docs:

□ Path Depth: □ Hops: □ Hours: 0

□ Exclude Filters:

□ Force Accept Filters:

Schedule

From	To	Type	Day/Month/Options	
	13/09/2012		Weekly	Monday
	13/09/2012		Monthly	1

Annotations

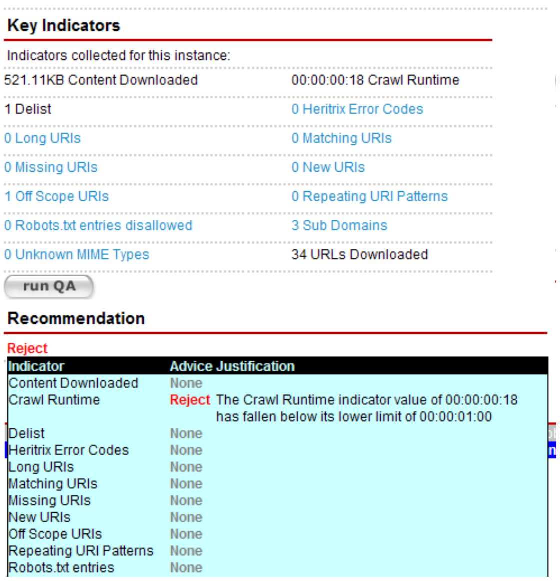
Harvest History

Start Date	State	Data	URLs	Failed	Elapsed	KB/s	Job Status	QA Status
13/09/2012 15:14:28	Harvested	521.11KB	34	0	19s	28.0	Finished	Harvested

Add Annotation

Figure 20. Target instance summary page

- **Harvest Results** - display the harvest results for the target instance; clicking the results displays the Harvest Results tab for the target instance
- **Profile Overrides** - access to the base profile for the target instance
- **Resources** - displays the seeds for the target instance; clicking a seed displays the Seeds tab for the target
- **Schedule** - enables modification of existing schedules
- **Key Indicators** - results of applying the Indicators defined in the System Administration Page for QA Indicators to the target instance; clicking a hyperlinked Indicator will display a generic report to explain the figure displayed. In the event that a target instance has been manually pruned, the **runQA** button is provided to re-compute the Indicator values and recommendation for the target instance.
- **Annotations** - lists the notes about the target instance.
- **Recommendation** - displays the final advice assigned to the target instance by considering all Indicator values. Hovering the mouse over the recommendation will display the advice for each indicator



Key Indicators

Indicators collected for this instance:

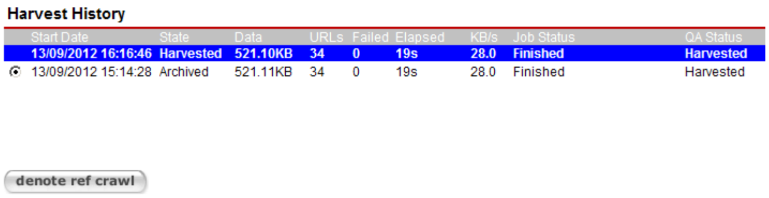
521.11KB Content Downloaded	00:00:00:18 Crawl Runtime
1 Delist	0 Heritrix Error Codes
0 Long URIs	0 Matching URIs
0 Missing URIs	0 New URIs
1 Off Scope URIs	0 Repeating URI Patterns
0 Robots.txt entries disallowed	3 Sub Domains
0 Unknown MIME Types	34 URLs Downloaded

Recommendation

Reject

Indicator	Advice	Justification
Content Downloaded	None	
Crawl Runtime	Reject	The Crawl Runtime indicator value of 00:00:00:18 has fallen below its lower limit of 00:00:01:00
Delist	None	
Heritrix Error Codes	None	
Long URIs	None	
Matching URIs	None	
Missing URIs	None	
New URIs	None	
Off Scope URIs	None	
Repeating URI Patterns	None	
Robots.txt entries	None	

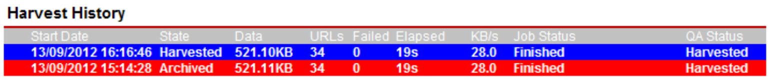
- **Add Annotation** - enables notes for the target instance to be added.
- **Harvest History** - displays all harvest history for the target instance's target. The current harvest is highlighted in blue. The harvest history for an archived target instance will be displayed with a radio option and clicking **denote ref crawl** will mark the selected archived target instance as the reference crawl for future crawls



Harvest History

	Start Date	State	Data	URLs	Failed	Elapsed	KB/s	Job Status	QA Status
	13/09/2012 16:16:46	Harvested	521.10KB	34	0	19s	28.0	Finished	Harvested
◉	13/09/2012 15:14:28	Archived	521.11KB	34	0	19s	28.0	Finished	Harvested

When an archived target instance is denoted as a reference crawl, it is used as a baseline to compare the indicators for future crawls and is highlighted in red

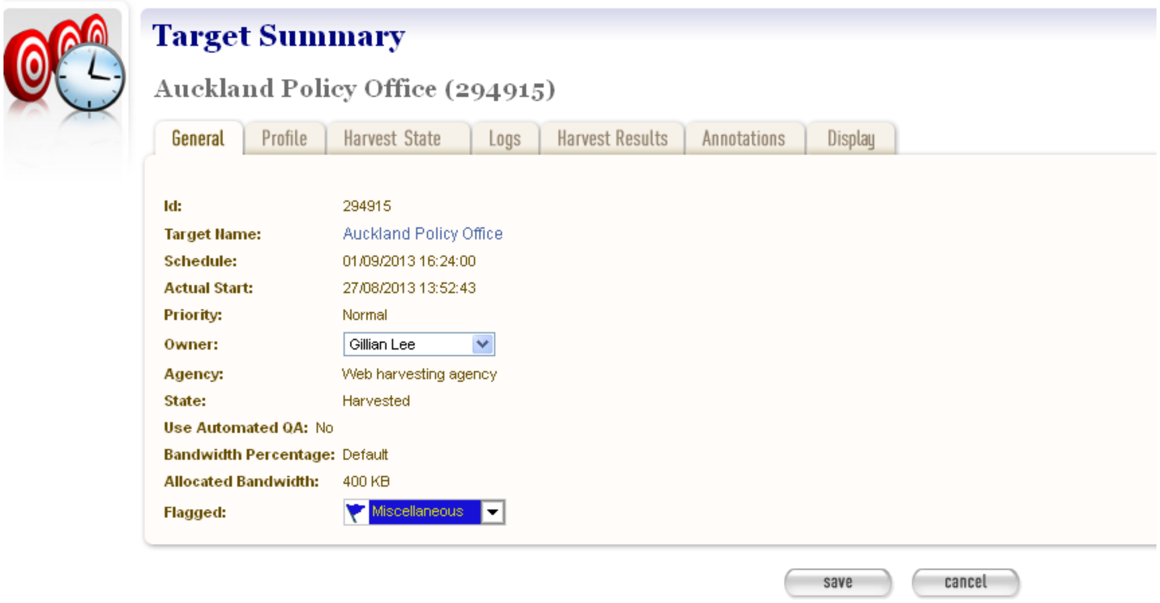


Harvest History

Start Date	State	Data	URLs	Failed	Elapsed	KB/s	Job Status	QA Status
13/09/2012 16:16:46	Harvested	521.10KB	34	0	19s	28.0	Finished	Harvested
13/09/2012 15:14:28	Archived	521.11KB	34	0	19s	28.0	Finished	Harvested

2. From the Target summary page. click 👁 to view a Target Instance, or ✎ to edit a Target Instance.

   The **View/Edit Target Instance** *page displays.*



Target Summary

Auckland Policy Office (294915)

Id: 294915

Target Name: Auckland Policy Office

Schedule: 01/09/2013 16:24:00

Actual Start: 27/08/2013 13:52:43

Priority: Normal

Owner: Gillian Lee

Agency: Web harvesting agency

State: Harvested

Use Automated QA: No

Bandwidth Percentage: Default

Allocated Bandwidth: 400 KB

Flagged: Miscellaneous

Figure 21. View/Edit Target Instance

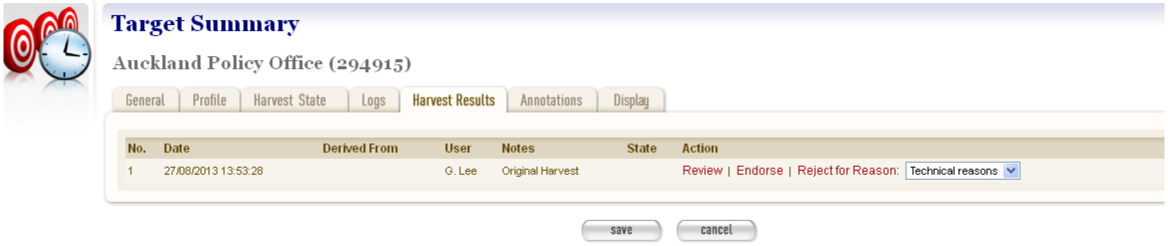
The **View/Edit Target Instance** page includes six tabs for viewing, running, or editing information about a target instance:

- **General** - general information about the Target Instance, such the Target it belongs to, schedule, owner, agency, etc.
- **Profile** - technical instructions on how to harvest the Target.
- **Harvest State** - details of the harvest, for example total bandwidth and amount of data downloaded.
- **Logs** - access to log files recording technical details of the harvest.
- **Harvest Results** - access to harvested content with options to review, endorse, reject, and archive harvest results.
- **Annotations** - notes about the Target Instance.
- **Display** - settings regarding the eventual display of the Target Instance in a browsing tool.

3.6.7 How to review, endorse or submit a target instance

3. Open the Target Instance in Edit mode, and click the **Harvest Results** tab.

   A list of target results displays.



Target Summary

Auckland Policy Office (294915)

No.	Date	Derived From	User	Notes	State	Action
1	27/08/2013 13:53:28		G. Lee	Original Harvest		Review | Endorse | Reject for Reason: Technical reasons

Figure 22. Harvest Results tab

Quality Review

4. To review a result, click **Review.**

   Quality Review is a complex task, and is covered separately in the next chapter.

Endorse or Reject harvest results

When you have finished reviewing a Target Instance, the **Done** button will return you to the harvest results page. At this point, you should know whether the harvest was successful, and should be **Endorsed**, or was unsuccessful, and should be **Rejected**.

5. To endorse the results, click **Endorse**.
6. To reject the results, click **Reject** and the reason for rejecting the TI.

Submit harvest results to an archive

Once you have endorsed a Target Instance, two new buttons appear that read '**Submit to Archive'** and **'Un-Endorse'**.

7. To archive an endorsed result, click **Submit to Archive**.
8. To un-endorse an erroneously endorsed instance, click **Un-Endorse**, this will set the target instance back to the **harvested** state.

   > *The Reject, Endorse, Un-Endorse and Submit to Archive links will automatically Save the Target Instance for you. You do not need to click on the* **save** *button after these operations (it won't hurt if you do).*

3.7 Target Instance Quality Review

3.7.1 Introduction

Target Instances are individual harvests that are scheduled to happen, or that are currently in progress, or that have already finished. See the previous chapter for an overview.

When a harvest is complete, the harvest result is saved in the digital asset store, and the Target Instance is saved in the Harvested state. The next step is for the Target Instance Owner to Quality Review the harvest result.

The first half of this chapter describes the quality review tools available when reviewing harvest results. The second half describes some problems that you may encounter when quality-reviewing harvest results in the Web Curator Tool, and how to diagnose and solve them. This includes detailed instructions and is intended for advanced users.

3.7.2 Terminology and status codes

Terminology

Important terms used with the Web Curator Tool include:

- **Target Instance** - a single harvest of a Target that is scheduled to occur (or which has already occurred) at a specific date and time.
- **harvest** - the process of crawling the web and retrieving specific web pages.
- **harvest result** - the files that are retrieved during a **harvest**.

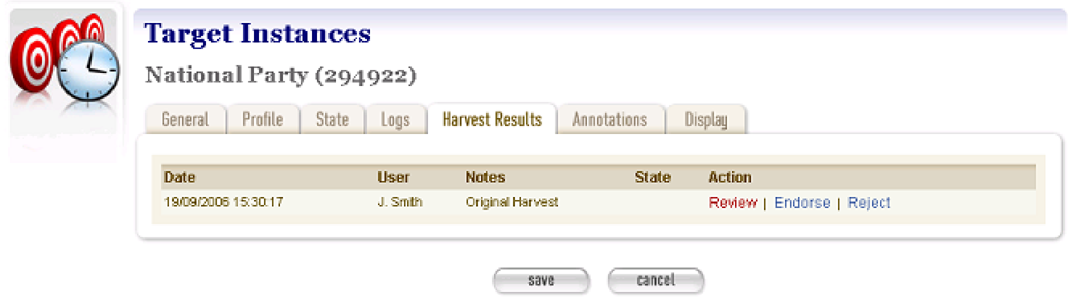
- **quality review** - the process of manually checking a **harvest result** to se if it is of sufficient quality to archive.
- **live url** - the real version of a URL that is used by the original website on the internet.
- **browse tool url** - the URL of a page in the **browse tool** (the browse tool URL is different for different harvest results).

  The browse tool URL is constructed as follows: http://wct.natlib.govt.nz/wct/curator/tools/browse/{[}Identifier{]}/{[}Live_URL] where [Identifier] is usually the Target Instance identifier, but may be an internal harvest result identifier.

3.7.3 Opening quality review tools

To review a harvested Target Instance, open it in edit mode, then select the Harvest Results tab.

A list of Target results displays. If this is the first time you have reviewed this Target Instance, a single Harvest Result will be displayed.



Target Instances

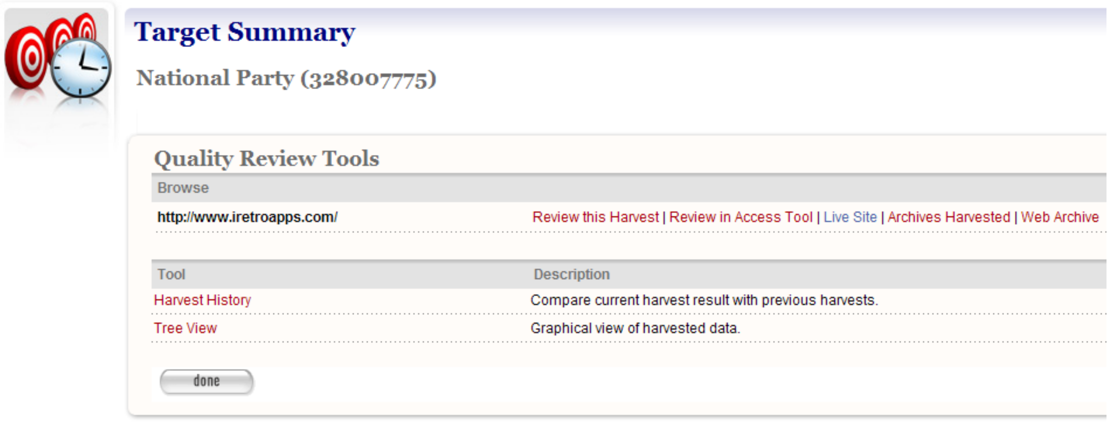
National Party (294922)

Date	User	Notes	State	Action
19/09/2006 15:30:17	J. Smith	Original Harvest		Review | Endorse | Reject

Figure 23. Harvest Results tab

To review a result, click Review. The next screen shows the available quality review tools.

Options for reviewing display.



Target Summary

National Party (328007775)

Quality Review Tools

Browse	
http://www.iretroapps.com/	Review this Harvest | Review in Access Tool | Live Site | Archives Harvested | Web Archive

Tool	Description
Harvest History	Compare current harvest result with previous harvests.
Tree View	Graphical view of harvested data.

Figure 24. Review Options

3.7.4 Quality review with the browse tool

The **Browse Tool** lets the user interact with a version of the harvest result with their web browser. It is designed to simulate the experience the user would have if they visited the original website. If the harvest is successful, the harvested material offers a comparable user experience to the original material.

The tool is controlled with a set of options in the Browse section of the Quality Review Tools screen. The Seed URLs for the harvest are listed at left, with three possible actions on the right:

- **Review this Harvest** - Open a view of the harvested Seed URL in a new window of your web browser. If this option is enabled it uses the internal WCT Browse Tool to generate the page.
- **Review in Access Tool** - Open a view of the harvested Seed URL in a new window of your web browser. If this option is enabled it uses an external Access Tool[1] to generate the page. This is the preferred browse tool.
- **Live Site** - Open the original web page in a new window
- **Archives Harvested** - Open any known archived versions of the site in a new window.
- **Web Archive** - Open the site entry page in the public archive (eg: http://www.webarchive.org.uk or http://archive.org/web/web.php).

The **Review this harvest (WCT browse tool)** is no longer being updated, which means some pages may not render properly. It is useful as a backup browser if the Access Tool goes down. It is also useful if you have several TI's of the same website harvested, as it only displays the TI requested.

The Review in Access Tool (OpenWayback) is the preferred browser as it is being maintained.

The **Live Site** link is provided so you can quickly open the original site for a side-by-side comparison with the harvested version.

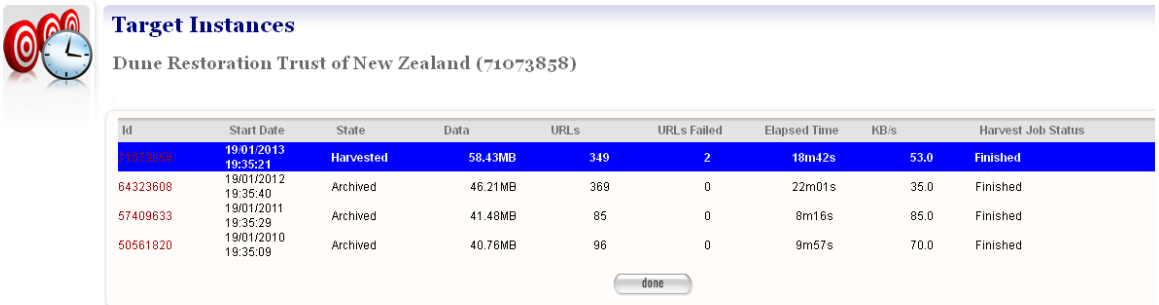
The **Archived Harvests** link lets you compare your harvest with previous harvests of the website.

Web Archive By default, the Web Curator Tool will open a list pages stored in the digital archive maintained by the Internet Archive, but your administrator can configure the tool to use your local archive instead.

3.7.5 Quality review with the harvest history tool

The **Harvest History Tool** can be used to quickly compare the harvest result of the current harvest to the result of previous harvests of the same Target.

The harvest history tool showing a history of the harvest results for a website that has been harvested every year.



Target Instances

Dune Restoration Trust of New Zealand (71073858)

Id	Start Date	State	Data	URLs	URLs Failed	Elapsed Time	KB/s	Harvest Job Status
71073858	19/01/2013 19:35:21	Harvested	58.43MB	349	2	18m42s	53.0	Finished
64323608	19/01/2012 19:35:40	Archived	46.21MB	369	0	22m01s	35.0	Finished
57409633	19/01/2011 19:35:29	Archived	41.48MB	85	0	8m16s	85.0	Finished
50561820	19/01/2010 19:35:09	Archived	40.76MB	96	0	9m57s	70.0	Finished

Figure 25. Harvest History.

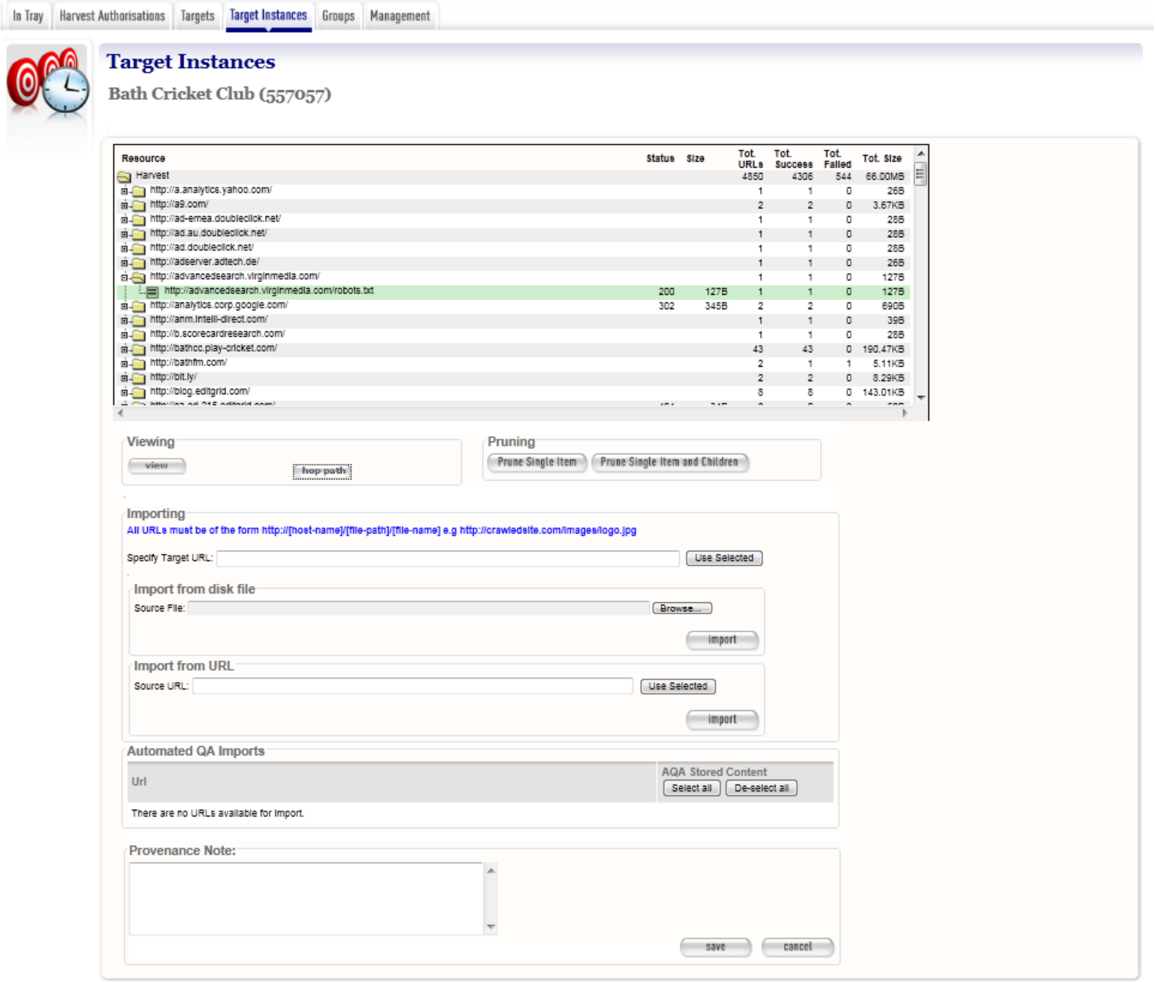
The tool shows all the harvests, with the most recent first. This allows the user to compare current and previous statistics for the number of pages downloaded, the number of download errors, the amount of data, and other statistics. If the user clicks on the link they are taken to the Target Instance view page corresponding to that particular harvest which in turn has a link back to the back to the Harvest History page from which they came.

[1] The use of the IIPC's redevelopment of the Java Wayback Machine - OpenWayback - as an access tool is described as in the Web Curator Wiki https://github.com/DIA-NZ/webcurator/wiki/ and is available to download from the OpenWayback project on Github https://github.com/iipc/openwayback.

3.7.6 Quality review with the prune tool

The **Tree Tool** gives you a graphical, tree-like view of the harvested data. It is a visualisation tool, but can also be used to delete unwanted material from the harvest or add new material.

A summary of the harvested web pages displayed in the tree tool.



Target Instances

Bath Cricket Club (557057)

Resource	Status	Size	Tot. URLs	Tot. Success	Tot. Failed	Tot. Size
Harvest			4850	4306	544	66.00MB
http://a.analytics.yahoo.com/			1	1	0	26B
http://a9.com/			2	2	0	3.67KB
http://ad-emea.doubleclick.net/			1	1	0	28B
http://ad.au.doubleclick.net/			1	1	0	28B
http://ad.doubleclick.net/			1	1	0	28B
http://adserver.adtech.de/			1	1	0	26B
http://advancedsearch.virginmedia.com/			1	1	0	127B
http://advancedsearch.virginmedia.com/robots.txt	200	127B	1	1	0	127B
http://analytics.corp.google.com/	302	345B	2	2	0	690B
http://anm.intelli-direct.com/			1	1	0	39B
http://b.scorecardresearch.com/			1	1	0	28B
http://bathcc.play-cricket.com/			43	43	0	190.47KB
http://bathfm.com/			2	1	1	5.11KB
http://bit.ly/			2	2	0	8.29KB
http://blog.editgrid.com/			8	8	0	143.01KB

Viewing

view hop path

Pruning

Prune Single Item Prune Single Item and Children

Importing

All URLs must be of the form http://[host-name]/[file-path]/[file-name] e.g http://crawledsite.com/images/logo.jpg

Specify Target URL: Use Selected

Import from disk file

Source File: Browse...

import

Import from URL

Source URL: Use Selected

import

Automated QA Imports

Url | AQA Stored Content Select all De-select all

There are no URLs available for import.

Provenance Note:

save cancel

Figure 26. Tree Tool

When the tool is opened, a series of rows is presented. The first row represents the complete harvest, and several additional columns are provided with additional data about the harvest.

Subsequent rows contain summary information about each of the websites visited during the crawl. These can be expanded to show the directories and files that were harvested from within the website. Note that each row may represent a page that was downloaded, or may represent a summary statistic, or may fulfil both roles.

On each row, the following statistics are presented:

- **Status** - The HTTP status for an entry that was downloaded.
- **Size** - The size (in bytes) of an entry that was downloaded.
- **Total URLs** - The number of attempts to download documents from “within” this site or folder.
- **Total Success** - The number of documents successfully downloaded from “within” this site or folder.
- **Total Failed** - The number of documents unsuccessfully downloaded from “within” this site or folder.

- **Total Size** - The number of bytes downloaded from "within" this site or folder.

Users can browse the tree structure and then view, prune or insert specific pages or files.

To view a page, select it in the display, and press the **view** button - it is also possible to see the hop-path for a specific item by clicking on the hop-path button.

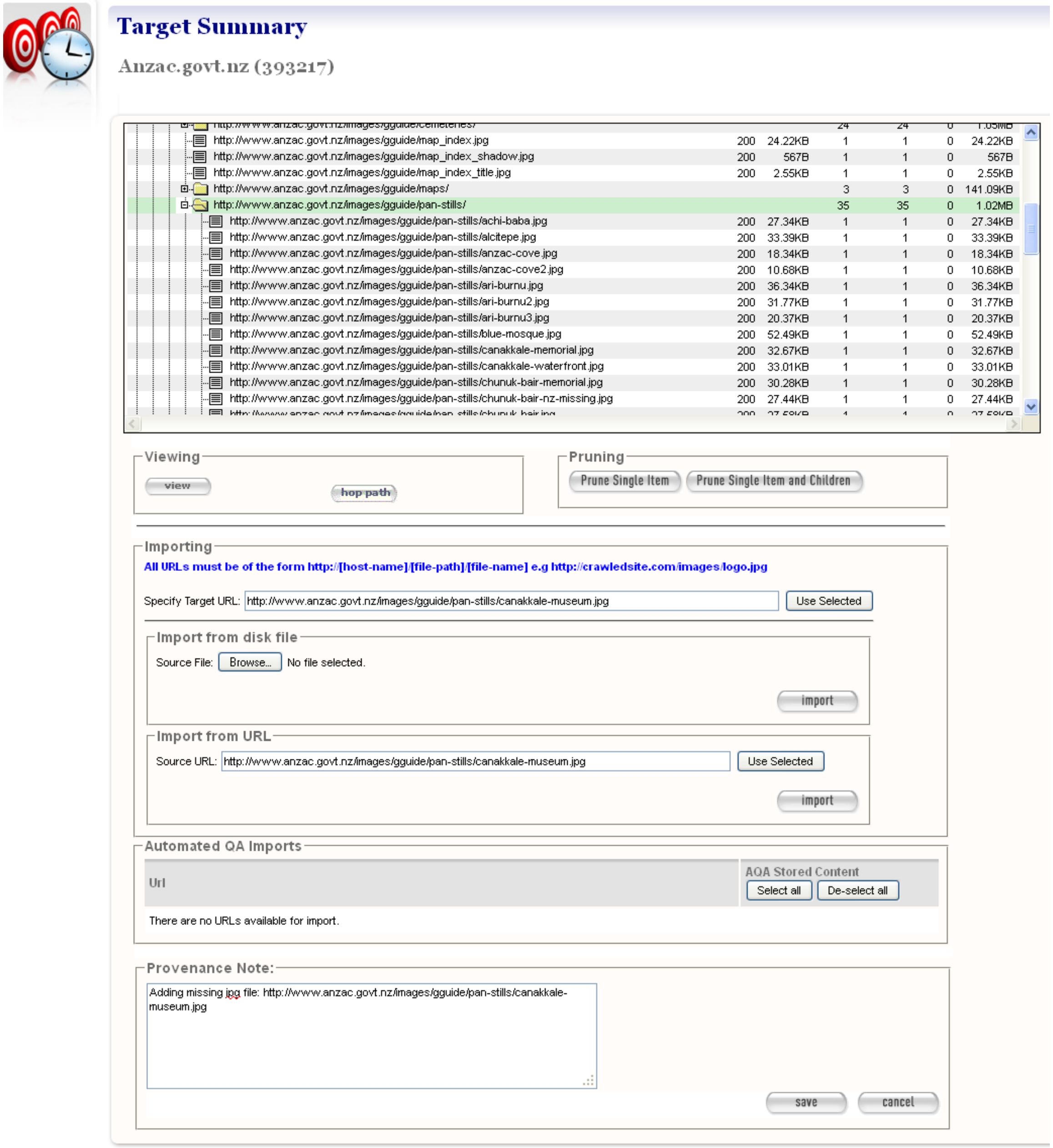
To prune a page, or a set of pages:

- Select the site, folder, or page that you want to prune
- click Prune Single Item to remove just the highlighted page; or Prune Item and Children to remove the page and all the pages "within" it

To insert a new page or missing item (such as a graphics file):

- Click on the folder in the Tree View where the item should appear (see Figure 23 below)
- Specify the full URL of the item as it should appear within the site harvest in **Specify Target URL**
- Specify the appropriate file location on disk or the appropriate external URL for the new item which is to be added and click on the appropriate Import button.
- The new item will be inserted at the appropriate place in the tree view hierarchy.

  Then after either type of action;
- Add a description of why you have pruned or inserted content to the provenance note textbox (required).
- Click Save. Note that for best efficiency it is best to combine multiple prune and import operations before saving - as a new Harvest Result is created after each operation which can be a very resource intensive operation on the server.



Target Summary

Anzac.govt.nz (393217)

http://www.anzac.govt.nz/images/gguide/cemeteries/			24	24	0	1.05MB
http://www.anzac.govt.nz/images/gguide/map_index.jpg	200	24.22KB	1	1	0	24.22KB
http://www.anzac.govt.nz/images/gguide/map_index_shadow.jpg	200	567B	1	1	0	567B
http://www.anzac.govt.nz/images/gguide/map_index_title.jpg	200	2.55KB	1	1	0	2.55KB
http://www.anzac.govt.nz/images/gguide/maps/			3	3	0	141.09KB
http://www.anzac.govt.nz/images/gguide/pan-stills/			35	35	0	1.02MB
http://www.anzac.govt.nz/images/gguide/pan-stills/achi-baba.jpg	200	27.34KB	1	1	0	27.34KB
http://www.anzac.govt.nz/images/gguide/pan-stills/alcitepe.jpg	200	33.39KB	1	1	0	33.39KB
http://www.anzac.govt.nz/images/gguide/pan-stills/anzac-cove.jpg	200	18.34KB	1	1	0	18.34KB
http://www.anzac.govt.nz/images/gguide/pan-stills/anzac-cove2.jpg	200	10.68KB	1	1	0	10.68KB
http://www.anzac.govt.nz/images/gguide/pan-stills/ari-burnu.jpg	200	36.34KB	1	1	0	36.34KB
http://www.anzac.govt.nz/images/gguide/pan-stills/ari-burnu2.jpg	200	31.77KB	1	1	0	31.77KB
http://www.anzac.govt.nz/images/gguide/pan-stills/ari-burnu3.jpg	200	20.37KB	1	1	0	20.37KB
http://www.anzac.govt.nz/images/gguide/pan-stills/blue-mosque.jpg	200	52.49KB	1	1	0	52.49KB
http://www.anzac.govt.nz/images/gguide/pan-stills/canakkale-memorial.jpg	200	32.67KB	1	1	0	32.67KB
http://www.anzac.govt.nz/images/gguide/pan-stills/canakkale-waterfront.jpg	200	33.01KB	1	1	0	33.01KB
http://www.anzac.govt.nz/images/gguide/pan-stills/chunuk-bair-memorial.jpg	200	30.28KB	1	1	0	30.28KB
http://www.anzac.govt.nz/images/gguide/pan-stills/chunuk-bair-nz-missing.jpg	200	27.44KB	1	1	0	27.44KB
http://www.anzac.govt.nz/images/gguide/pan-stills/chunuk-bair.jpg	200	27.58KB	1	1	0	27.58KB

Importing

All URLs must be of the form http://[host-name]/[file-path]/[file-name] e.g http://crawledsite.com/images/logo.jpg

Specify Target URL: http://www.anzac.govt.nz/images/gguide/pan-stills/canakkale-museum.jpg

Import from disk file

Source File: No file selected.

Import from URL

Source URL: http://www.anzac.govt.nz/images/gguide/pan-stills/canakkale-museum.jpg

Automated QA Imports

Url	AQA Stored Content
There are no URLs available for import.	

Provenance Note:

Adding missing jpg file: http://www.anzac.govt.nz/images/gguide/pan-stills/canakkale-museum.jpg

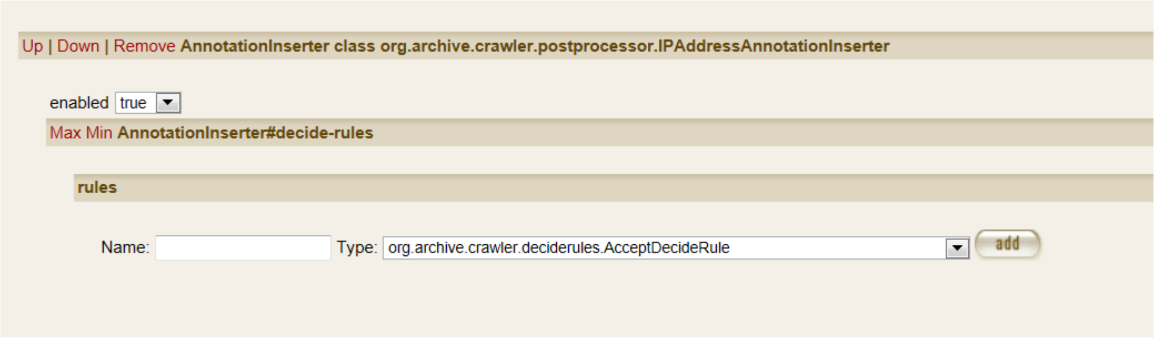
Figure 27. Adding a missing jpg file

The display returns to the Harvest Results tab.

3.7.7 The log file viewer

Although it is not a quality review tool, the Web Curator Tool log file viewer can assist with quality review by letting you examine the log files for Target Instances that that are running or harvested.

If you want the IP address associated with a harvested item to be captured at the end of each line in the crawl.log file the profile being used by the Heritrix Crawler for that harvest must contain a post-processor class called IPAddressAnnotationInserter (see screen shot of the relevant section of the post-processors tab in the profile editor).



Up | Down | Remove AnnotationInserter class org.archive.crawler.postprocessor.IPAddressAnnotationInserter

enabled true

Max Min AnnotationInserter#decide-rules

rules

Name: Type: org.archive.crawler.deciderules.AcceptDecideRule

The log file viewer is launched from the Logs tab of the Target Instance edit pages, and by default the final 700 lines of the log are displayed. However, there are several advanced features.

View the entire file

Open a log in the Log File Viewer, then set the *Number of lines to display* field to 99999 and click the update button. This will show the entire log file (unless the harvest had more than 100,000 URLs).

View only the lines that contain a specified substring

The *regular expression filter* box can be used to restrict the lines that are displayed to only those that match a pattern (or “regular expression”).

For example:

- **To show only lines that include report.pdf**: Set the regular expression filter to **.*report.pdf.*** and press update.

  In the regular expression language, the dot (“.”) means “any character” and the star (asterisk, or “*”) means “repeated zero or more times. So “.*” (which is often pronounced “dot-star”) means any character repeated zero or more times, and the regular expression above means “show all the lines that have any sequence of characters, followed by “report.pdf”, followed by any other sequence of characters.

- **To find out whether a specific URL is in the crawl.log**: Suppose you want to see if http://www.example.com/some/file.html was downloaded. Open the crawl.log file in the Log File Viewer, enter the regular expression .*http://www.example.com/some/file.html.* and press update.

3.7.8 Diagnosing problems with completed harvests

Many harvest problems only become evident once a harvest is complete and loaded in the browse tool. For example, some images may not display properly, or some stylesheets may not be loaded, or some links may not work.

Diagnosis

In these cases, the general procedure is to

1. Determine the URL (or URLs) that are not working. Some good techniques are:

   - Go to the live site, and find the page that the missing URL is linked from. Find out the missing URL by

     – opening the document in the browser (applies to links, images) and reading the URL from the Location bar, or

     – by right-clicking on the missing object (images and links), or

   - by using view source to see the HTML (stylesheets), or
   - by using the Web Developer Toolbar to view CSS information (Stylesheets-see Tools section below).

2. Determine whether the harvester downloaded the URL successfully. Here are some of the ways you might do this (from simplest to most complex):
   - Open the Prune Tool and see if the URL is displayed. If the URL is not present, then it was **not downloaded** during the crawl.
   - Calculate the browse tool URL, and see if it can be loaded in the Browse Tool. If so, the URL was **downloaded successfully**.
   - Examine the crawl.log file in the Log File Viewer to see if the URL was harvested and what its status code was.
     - If the URL is not in the crawl.log file, the URL was **not downloaded**.
     - If the URL is in the crawl.log file with a status code indicating a successful download (such as 200, or some other code of the form 2XX) then the URL was **downloaded successfully**.
     - If the URL is in the crawl.log file with a status code indicating a failed download (such as -1) then there was a **download error**. Check the Heritrix status codes are described in Section 4 below for information about what went wrong.
3. If the URL was **downloaded successfully** by the harvester but is not displaying, then there is a problem with the browse tool that needs to be fixed by an administrator or developer. The good news is that your harvest was (probably) successful-you just can't see the results.
   - Some common cases in Web Curator Tool version 1.1 (which are fixed in later versions) include:
     - web pages with empty anchor tags (SourceForge bug 1541022),
     - paths that contain spaces (bug 1692829),
     - some Javascript links (bug 1666472),
     - some background images will not render (bug 1702552), and
     - CSS files with import statements (bug 1701162).
   - You should probably endorse the site if:
     - there are relatively few URLs affected by the problem, or
     - the information on the site is time critical and may not be available by the time Web Curator Tool 1.2 is installed.
4. If the URL was **not downloaded** by the harvester, determine why:
   - It is possible that the crawl finished before the URL could be downloaded. Check to see if the state of the crawl (in the "Harvest State" tab of the Target Instance) says something like "Finished - Maximum document limit reached". To fix:
     - Increase the relevant limit for the Target using the Profile Overrides tab.
     - If this is a common problem, you may want to ask an administrator to increase the default limit set in the harvester profile.
   - It is possible that the URL is out of scope for the crawl. The most obvious case is where the URL has a different host. It is also possible that the harvester is configured to only crawl the website to a certain depth, or to a certain number of hops (i.e. links from the homepage). To fix:
     - For resources on different hosts, you can adjust the scope for the crawl by adding a new (secondary) seed URL.

  - For path depth or hops issues, you can add a new secondary seed to extend the scope, or you can increase the relevant limit for the Target using the Profile Overrides tab.
- It is possible that the URL appears on a page that the Heritrix harvester cannot understand.
  - URLs that appear in CSS, Shockwave Flash Javascript and other files will not be installed unless the harvest profile includes the correct "Extactor" plugin: ExtractorCSS, ExtractorSWF, ExtractorJS, etc. These will not be part of your profile (in WCT 1.1) unless your administrator adds them.
  - URLs that appear in new or rare page types may not be parsed.
- It is possible that the URL does not appear explicitly on the page. For example, instead of linking to a URL directly, a Javascript function may be used to construct the URL out of several bits and pieces. To fix:
  - There may be no easy way to fix this problem, since it is extremely hard for the harvester to interpret every single piece of Javascript it encounters (though it does try).
  - If there are only one or two affected files, or if the affected files are very important, you can add the affected files as secondary seeds.
  - If you are very lucky, all the affected files might be stored in the same location, such as a single directory, which can be crawled directly with a single additional seed.

5. If the URL was not retrieved because of a **download error** then the Heritrix status code can be used to diagnose the problem.
   - See https://github.com/internetarchive/heritrix3/wiki/Status-Codes for a list of Heritrix status codes.
   - A 500 (or other 5XX) status code indicates an internal server error. If you see 500 status codes when you download with Heritrix, but are able to browse successfully in your web browser, it may be that the website is recognising the web curator tool and sending you errors (to prevent you from crawling the website). See the section on the Firefox User Agent Switcher below for information on diagnosing this problem. To resolve it, you can either negotiate with the web site administrator to allow you to harvest, or set up a profile that gives a false user agent string.

Common problems

Here are some common problems, and their solutions:

- **Formatting not showing up in the browse tool**. We most often see this when a CSS file has not been downloaded (due to an oversight by the crawler). To see if this is the real problem, use "View Source" in your browser to identify the missing CSS file (or files-some pages have several), then check whether it was really downloaded. If not, try adding the CSS file as a secondary seed URL in the target and re-harvesting.

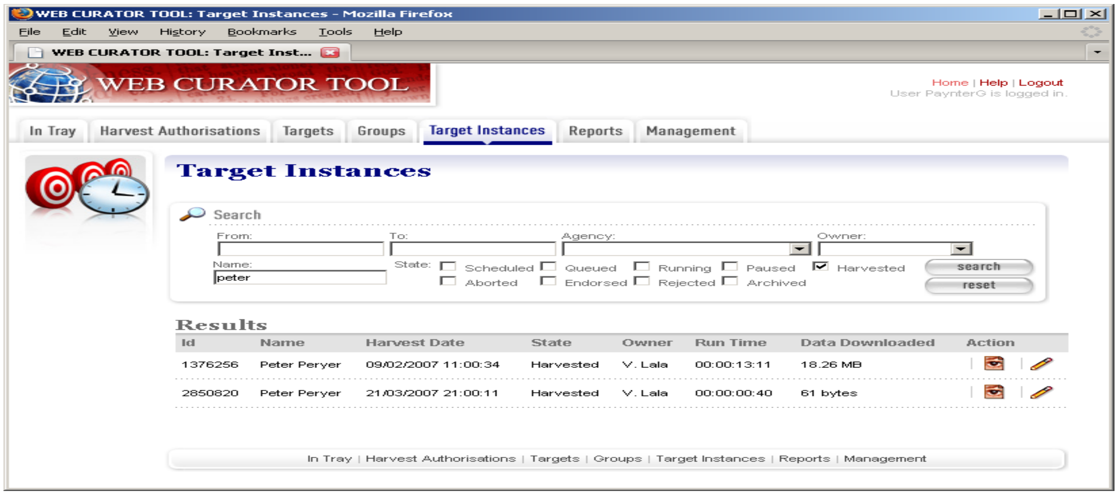
3.7.9 Diagnosing when too little material is harvested

Sometimes a harvest fails to complete, or does not harvest as much material as you expected. This section describes some common causes of this problem.

When no material is downloaded (the "61 bytes" result)

In the screenshot below, the same website was harvested twice, and the quantity of data harvested fell from 18 MB to 61 bytes. This tells us that the second harvest has effectively failed.

Two harvests of the same website, undertaken a month apart, showing a dramatic change in the size of the harvest result.


WEB CURATOR TOOL: Target Instances - Mozilla Firefox
File Edit View History Bookmarks Tools Help
WEB CURATOR TOOL: Target Inst...
WEB CURATOR TOOL
Home | Help | Logout
User PaynterG is logged in.
In Tray | Harvest Authorisations | Targets | Groups | Target Instances | Reports | Management
search
reset
In Tray | Harvest Authorisations | Targets | Groups | Target Instances | Reports | Management

Target Instances

Search

From: To: Agency: Owner:

Name: peter

State: □ Scheduled □ Queued □ Running □ Paused ☑ Harvested □ Aborted □ Endorsed □ Rejected □ Archived

Results

Id	Name	Harvest Date	State	Owner	Run Time	Data Downloaded	Action
1376256	Peter Peryer	09/02/2007 11:00:34	Harvested	V. Lala	00:00:13:11	18.26 MB	
2850820	Peter Peryer	21/03/2007 21:00:11	Harvested	V. Lala	00:00:00:40	61 bytes	

Figure 28: Target Instance that failed to complete.

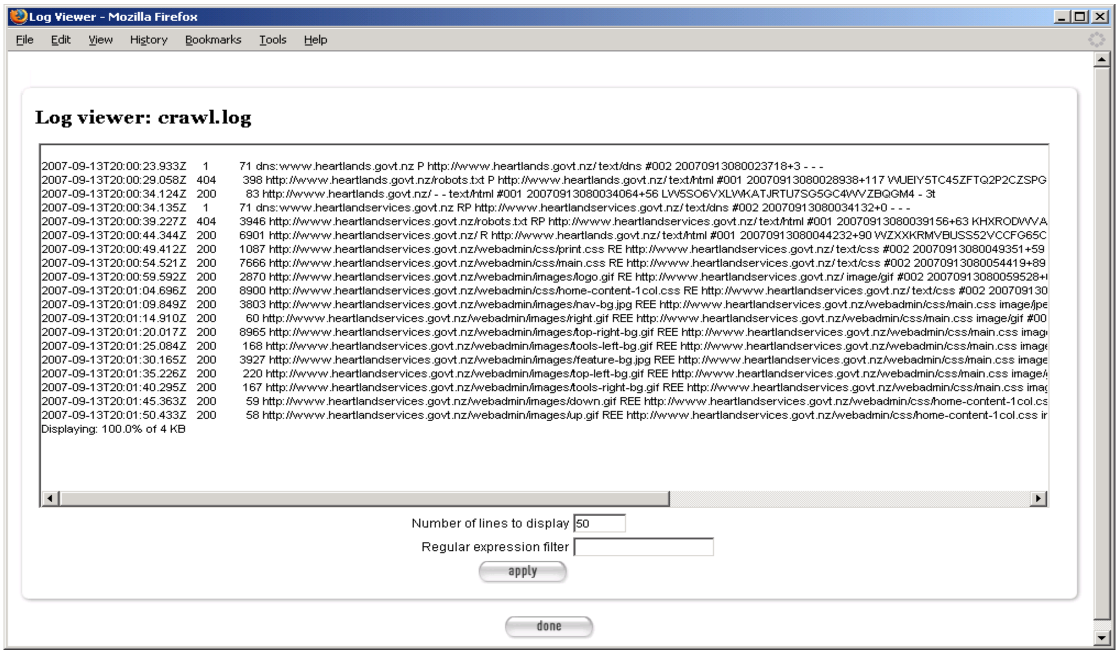
In these cases, the general procedure is to

1. Open the Target Instance (in either mode) and check the Harvest State tab to verify that the crawl is in the "Finished" state.
2. If the Target Instance Harvest State tab does not show the Finished state, then a message will usually explain the problem.
3. Open the Logs tab and check whether any error logs have been created.
    - If there is a local-errors.log file, open it in the Log file viewer, and see what kind of errors are shown. Some examples:
        - Errors that include "*Failed to get host [hostname] address from ServerCache*" indicate that the harvester was unable to look up the hostname in DNS, which probably means there was an error connecting to the internet (it may also mean you entered the URL incorrectly in the Target seed URLs).

When only the homepage is downloaded

In some cases a harvest may appear to work, but will result in only the homepage being visible in the browse tool. This can be because the seed URL you have entered is an alias to the "real" URL for the website.

For example, the screenshot below shows the crawl.log file for a harvest of the seed URL www.heartlands.govt.nz, which is successfully downloaded (third line) but contains only a redirect to the "real" version of the site at www.heartlandservices.govt.nz. This new web page is successfully downloaded (line 6), and all its embedded images and stylesheets are also downloaded (lines 7-19), but no further pages on www.heartlandservices.govt.nz are harvested because the site is out-of-scope relative to the seed URL.


Log Viewer - Mozilla Firefox
File Edit View History Bookmarks Tools Help

Log viewer: crawl.log

```
2007-09-13T20:00:23.933Z   1      71 dns:www.heartlands.govt.nz P http://www.heartlands.govt.nz/ text/dns #002 20070913080023718+3 - - -
2007-09-13T20:00:29.058Z 404     398 http://www.heartlands.govt.nz/robots.txt P http://www.heartlands.govt.nz/ text/html #001 20070913080028938+117 WUEIY5TC45ZFTQ2P2CZSPG
2007-09-13T20:00:34.124Z 200      83 http://www.heartlands.govt.nz/ - - text/html #001 20070913080034064+56 LW5SO6VXLWKATJRTU7SG5GC4WVZBQGM4 - 3t
2007-09-13T20:00:34.135Z   1      71 dns:www.heartlandservices.govt.nz RP http://www.heartlandservices.govt.nz/ text/dns #002 20070913080034132+0 - - -
2007-09-13T20:00:39.227Z 404    3946 http://www.heartlandservices.govt.nz/robots.txt RP http://www.heartlandservices.govt.nz/ text/html #001 20070913080039156+63 KHXRODWVA
2007-09-13T20:00:44.344Z 200    6901 http://www.heartlandservices.govt.nz/ R http://www.heartlands.govt.nz/ text/html #001 20070913080044232+90 WZXXKRMVBUSS52VCCFG65C
2007-09-13T20:00:49.412Z 200    1087 http://www.heartlandservices.govt.nz/webadmin/css/print.css RE http://www.heartlandservices.govt.nz/ text/css #002 20070913080049351+59
2007-09-13T20:00:54.521Z 200    7666 http://www.heartlandservices.govt.nz/webadmin/css/main.css RE http://www.heartlandservices.govt.nz/ text/css #002 20070913080054419+89
2007-09-13T20:00:59.592Z 200    2870 http://www.heartlandservices.govt.nz/webadmin/images/logo.gif RE http://www.heartlandservices.govt.nz/ image/gif #002 20070913080059528+
2007-09-13T20:01:04.696Z 200    8900 http://www.heartlandservices.govt.nz/webadmin/css/home-content-1col.css RE http://www.heartlandservices.govt.nz/ text/css #002 200709130
2007-09-13T20:01:09.849Z 200    3803 http://www.heartlandservices.govt.nz/webadmin/images/nav-bg.jpg REE http://www.heartlandservices.govt.nz/webadmin/css/main.css image/jpe
2007-09-13T20:01:14.910Z 200      60 http://www.heartlandservices.govt.nz/webadmin/images/right.gif REE http://www.heartlandservices.govt.nz/webadmin/css/main.css image/gif #00
2007-09-13T20:01:20.017Z 200    8965 http://www.heartlandservices.govt.nz/webadmin/images/top-right-bg.gif REE http://www.heartlandservices.govt.nz/webadmin/css/main.css imag
2007-09-13T20:01:25.084Z 200     168 http://www.heartlandservices.govt.nz/webadmin/images/tools-left-bg.gif REE http://www.heartlandservices.govt.nz/webadmin/css/main.css image
2007-09-13T20:01:30.165Z 200    3927 http://www.heartlandservices.govt.nz/webadmin/images/feature-bg.jpg REE http://www.heartlandservices.govt.nz/webadmin/css/main.css image
2007-09-13T20:01:35.226Z 200     220 http://www.heartlandservices.govt.nz/webadmin/images/top-left-bg.gif REE http://www.heartlandservices.govt.nz/webadmin/css/main.css image/
2007-09-13T20:01:40.295Z 200     167 http://www.heartlandservices.govt.nz/webadmin/images/tools-right-bg.gif REE http://www.heartlandservices.govt.nz/webadmin/css/main.css imag
2007-09-13T20:01:45.363Z 200      59 http://www.heartlandservices.govt.nz/webadmin/images/down.gif REE http://www.heartlandservices.govt.nz/webadmin/css/home-content-1col.cs
2007-09-13T20:01:50.433Z 200      58 http://www.heartlandservices.govt.nz/webadmin/images/up.gif REE http://www.heartlandservices.govt.nz/webadmin/css/home-content-1col.css i
Displaying: 100.0% of 4 KB
```

Number of lines to display 50

Regular expression filter

Figure 29. Crawl log

The solution to this problem is to add the "real" site as a primary or secondary seed URL.

3.7.10 Diagnosing when too much material is harvested

Sometimes a harvest will complete, and will look right in the browse tool, but will appear to be far too large: either too many URLs were downloaded, or you harvested more data than you expected.

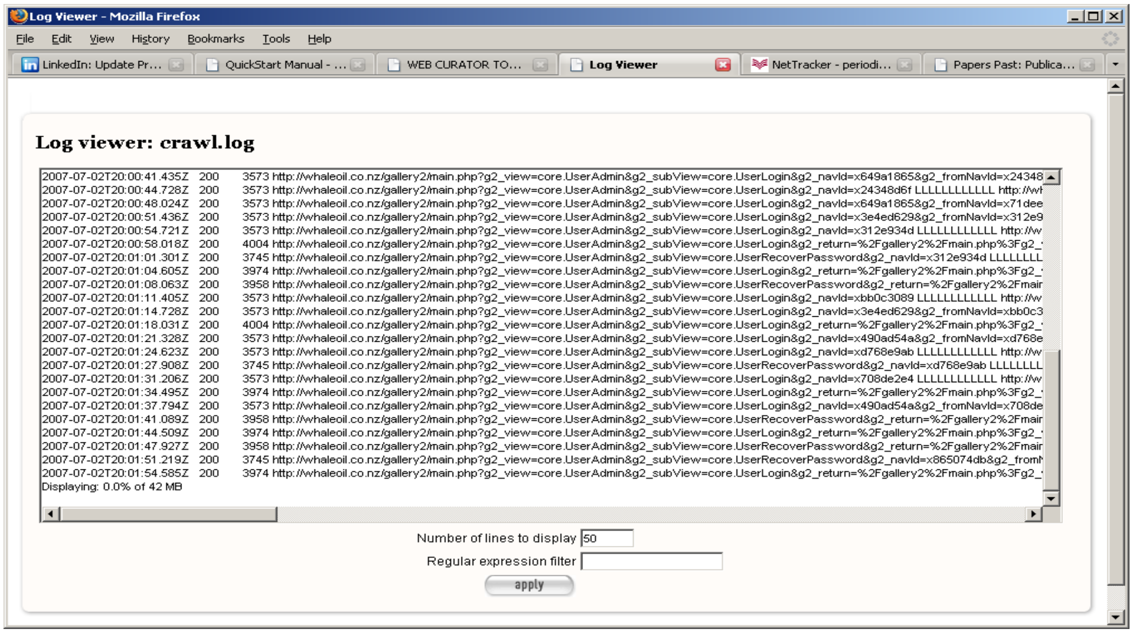
Too many URLs downloaded

Sometimes a harvest will be larger than expected, and will involve a large number of URLs. The harvest will often show the following status value in the Harvest Status tab of the Target Instance:

> **Finished - Maximum number of documents limit hit**

It is possible that the harvester has become caught in a "spider trap" or some other unintended loop. The best way to investigate this problem is to go to the Target Instance Logs tab, and to view the crawl.log file. By default, this shows you the last 50 lines of the log file, and this is where the problem is most likely to be.

For example, one recent harvest downloaded 100,000 documents, and finished with the requests shown in this log file viewer window.



Log viewer: crawl.log

```
2007-07-02T20:00:41.435Z  200   3573 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserLogin&g2_navId=x649a1865&g2_fromNavId=x24348
2007-07-02T20:00:44.728Z  200   3573 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserLogin&g2_navId=x24348d6f LLLLLLLLLLLL http://wh
2007-07-02T20:00:48.024Z  200   3573 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserLogin&g2_navId=x649a1865&g2_fromNavId=x71dee
2007-07-02T20:00:51.436Z  200   3573 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserLogin&g2_navId=x3e4ed629&g2_fromNavId=x312e9
2007-07-02T20:00:54.721Z  200   3573 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserLogin&g2_navId=x312e934d LLLLLLLLLLLL http://w
2007-07-02T20:00:58.018Z  200   4004 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserLogin&g2_return=%2Fgallery2%2Fmain.php%3Fg2_
2007-07-02T20:01:01.301Z  200   3745 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserRecoverPassword&g2_navId=x312e934d LLLLLLLL
2007-07-02T20:01:04.605Z  200   3974 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserLogin&g2_return=%2Fgallery2%2Fmain.php%3Fg2_
2007-07-02T20:01:08.063Z  200   3958 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserRecoverPassword&g2_return=%2Fgallery2%2Fmair
2007-07-02T20:01:11.405Z  200   3573 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserLogin&g2_navId=xbb0c3089 LLLLLLLLLLLL http://w
2007-07-02T20:01:14.728Z  200   3573 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserLogin&g2_navId=x3e4ed629&g2_fromNavId=xbb0c3
2007-07-02T20:01:18.031Z  200   4004 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserLogin&g2_return=%2Fgallery2%2Fmain.php%3Fg2_
2007-07-02T20:01:21.328Z  200   3573 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserLogin&g2_navId=x490ad54a&g2_fromNavId=xd768e
2007-07-02T20:01:24.623Z  200   3573 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserLogin&g2_navId=xd768e9ab LLLLLLLLLLLL http://w
2007-07-02T20:01:27.908Z  200   3745 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserRecoverPassword&g2_navId=xd768e9ab LLLLLLLL
2007-07-02T20:01:31.206Z  200   3573 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserLogin&g2_navId=x708de2e4 LLLLLLLLLLLL http://w
2007-07-02T20:01:34.495Z  200   3974 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserLogin&g2_return=%2Fgallery2%2Fmain.php%3Fg2_
2007-07-02T20:01:37.794Z  200   3573 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserLogin&g2_navId=x490ad54a&g2_fromNavId=x708de
2007-07-02T20:01:41.089Z  200   3958 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserRecoverPassword&g2_return=%2Fgallery2%2Fmair
2007-07-02T20:01:44.509Z  200   3974 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserLogin&g2_return=%2Fgallery2%2Fmain.php%3Fg2_
2007-07-02T20:01:47.927Z  200   3958 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserRecoverPassword&g2_return=%2Fgallery2%2Fmair
2007-07-02T20:01:51.219Z  200   3745 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserRecoverPassword&g2_navId=x865074db&g2_fromN
2007-07-02T20:01:54.585Z  200   3974 http://whaleoil.co.nz/gallery2/main.php?g2_view=core.UserAdmin&g2_subView=core.UserLogin&g2_return=%2Fgallery2%2Fmain.php%3Fg2_
Displaying: 0.0% of 42 MB
```

Number of lines to display: 50

Regular expression filter:

Figure 30: the log file viewer showing the crawl log.

Note that many of the requests are repeated calls to the CGI script http://whaleoil.co.nz/gallery2/main.php that include the parameters:

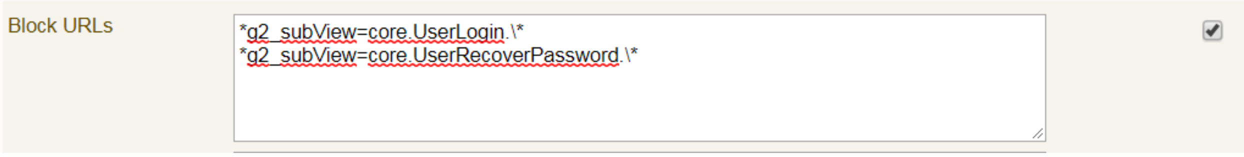
- g2_view=core.UserAdmin&g2_subView=core.UserLogin, or
- g2_view=core.UserAdmin&g2_subView=core.UserRecoverPassword

and that resolve to similar pages which have no real value to the harvest. These URLs are spurious and should not be harvested (and there are tens of thousands of them).

You can filter these URLs out of future harvests by going to the associated Target and opening the Profile tab and adding the following two lines to the "Exclude Filters" box for Heritrix 1 or "Block URLs" box for Heritrix 3:

- .*g2_subView=core.UserLogin.*
- .*g2_subView=core.UserRecoverPassword.*

The first line will ensure that all URLs that match include the substring 'g2_subView=core.UserLogin' will be excluded from future harvests, and the second line will do the same for the "Recover Password" URLs.


Block URLs

```
*g2_subView=core.UserLogin.\*
*g2_subView=core.UserRecoverPassword.\*
```

☑

3.7.11 Third-party quality review tools

The main tools used to diagnose harvest errors are your web browser, and the WCT Quality Review Tools: the Browse Tool and the Prune Tool. However, other tools that may be useful.

Web Developer Toolbar for Firefox and Chrome

The Web Developer Toolbars provide a toolbar in the Firefox and Chrome web browsers with numerous features for diagnosing problems with websites.

The full set of functionality is quite daunting, but these features can be very useful:

- **View the CSS information about a page**: Open the page in Firefox, then choose *View CSS* from the *CSS* menu. A new window (or tab) will be opened that lists all the stylesheets that were loaded in order to display the page, and which also show the contents of each of the stylesheets.
- **View the URL Path of each image in a page**: Open the page in Firefox, then choose *Display Image Paths* from the *Image* menu. Each image will have its URL path superimposed over the image. (Use the same menu to turn it off again.)
- **Get a list of all the links out of a page**: Open a page in Firefox, then choose *View Link Information* from the *Information* menu. A new window (or tab) will be opened that lists all the URLs that the page links to.

There are numerous other functions in the Web Developer Toolbar.

The Heritrix User Manual

The Heritrix User Manual includes a section that explains how to interpret Heritrix Log files-these are the same log files you see in the Web Curator Tool.

Useful sections include:

- **Interpreting crawl.log**:
  https://github.com/internetarchive/heritrix3/wiki/Logs#crawllog
- **Status code definitions:** This explains the status codes that
  appear in the crawl log:
  https://github.com/internetarchive/heritrix3/wiki/Status-Codes
- **Interpreting progress-statistics.log**:
  https://github.com/internetarchive/heritrix3/wiki/Logs#progress-statisticslog
- **Interpreting Reports: See Section 8.3:** http://crawler.archive.org/articles/user_manual/analysis.html#logs

User Agent Switcher for Firefox

The User Agent Switcher addon for Firefox (https://addons.mozilla.org/en-US/firefox/addon/59) provides a menu in the Firefox web browser that lets you tell Firefox to request a page but to identify itself as a different User Agent.

This is useful to identify those (thankfully rare) websites that give one sort of content to some web agents (such as web browsers like Firefox, Internet Explorer, and Safari), and other content to different web browsers (such as Heritrix, Googlebot, etc).

To test whether this is happening to you, switch the user agent Firefox is using to the one used in the Web Curator Tool, and then attempt to browse the relevant site.

- Default Heritrix 1 string `Mozilla/5.0 (compatible; heritrix/1.14.1 +http://dia-nz.github.io/webcurator/)`
- Default Heritrix 3 string `Mozilla/5.0 (compatible; heritrix/3.3.0 +http://dia-nz.github.io/webcurator/`

3.8 Groups

3.8.1 Introduction

Groups are a mechanism for associating two or more Targets that are related in some way. For example, a Group might be used to associate all the Targets that belong to a particular collection, subject, or event.

It is possible to create nested groups, where a specialised group (like Hurricanes) is itself a member of a more general group, (such as Natural Disasters).

Groups may have a start and end date. This can be used to define groups that are based on events, such as elections.

In many ways, Groups behave in a very similar way to Targets. They can have a name, a description, an owner, and can be searched for and edited. Groups can also be used to synchronise the harvest of multiple related Targets by attaching a schedule to the Group.

Target Instances inherit their group membership from Targets. When a Target Instance is submitted to an archive, its Target metadata is included in the SIP, including all Group information.

Terminology

Important terms used with the Web Curator Tool include:

- **group** - a set of targets (or other groups) that are related in some way.
- **member** - a group member is a target or group that belongs to the group.
- **expired** - a group is said to have expired when its end date has passed.

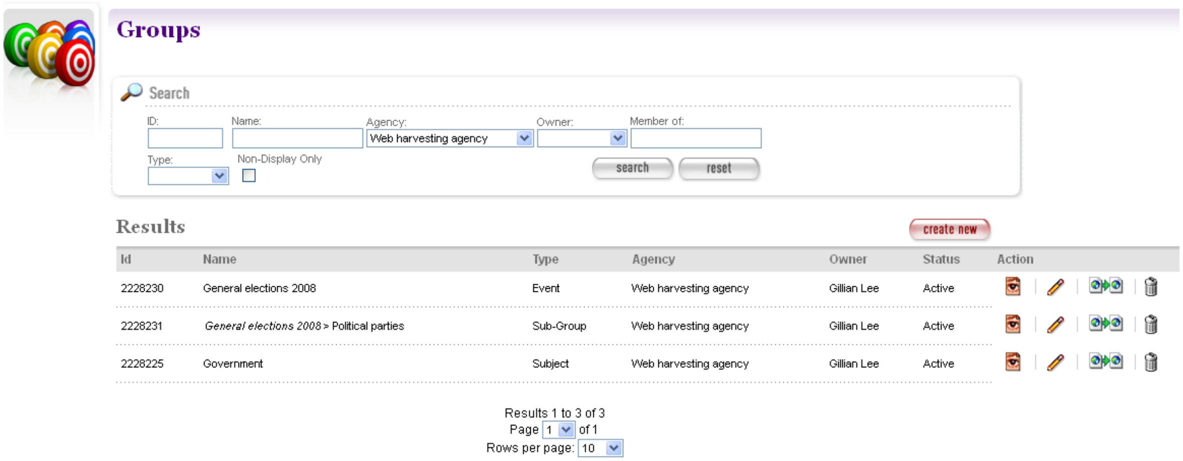
Target status

Each group has a status that is automatically calculated by the system:

- **schedulable** - at least one of its members are approved, and therefore a schedule can be attached to this group.
- **unschedulable** - no members of the group are approved, and therefore no schedule can be attached to this group.

3.8.2 Group search page

You manage Groups from the **Group search page**:



ID:
Name:
Agency: Web harvesting agency
Owner:
Member of:
Type:
Non-Display Only □

Id	Name	Type	Agency	Owner	Status	Action
2228230	General elections 2008	Event	Web harvesting agency	Gillian Lee	Active	
2228231	*General elections 2008* > Political parties	Sub-Group	Web harvesting agency	Gillian Lee	Active	
2228225	Government	Subject	Web harvesting agency	Gillian Lee	Active	

Results 1 to 3 of 3
Page 1 of 1
Rows per page: 10

Figure 31. Group search page

At the top of the page are fields to search for existing groups by **ID**, **Name**, **Agency**, **Owner**, **Member Of**, and **Group Type**.

Non-Display Only allows users to see Groups which have been flagged as hidden.

> *The search page remembers your last search and repeats it as the default search, initially defaulting to search based on your Agency only.*

The search results are listed at the bottom of the page. For each, you may have these options, depending on its state and your permissions:

👁 - **View** the Group

✏ - **Edit** the Group

⇄ - **Copy** the Group and create a new one

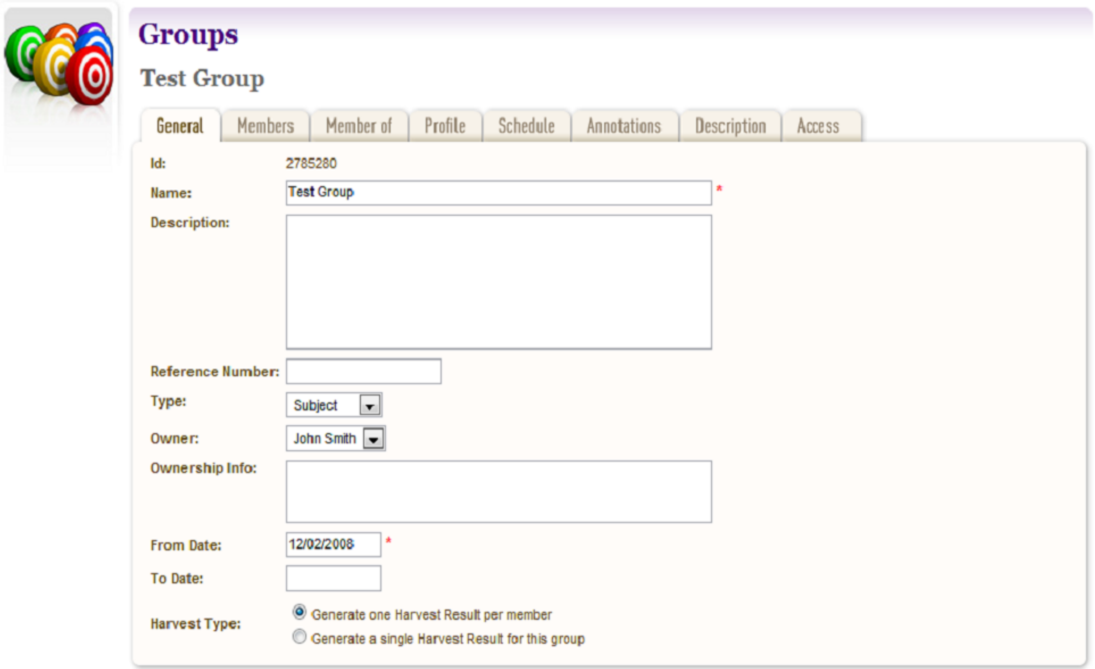
🗑 - **Delete** the Group

3.8.3 How to create a group

From the *Groups* page,

1. Click **create new**.

   The **Create/Edit Groups** *page displays.*



Id:	2785280
Name:	Test Group *
Description:	
Reference Number:	
Type:	Subject
Owner:	John Smith
Ownership Info:	
From Date:	12/02/2008 *
To Date:	
Harvest Type:	● Generate one Harvest Result per member ○ Generate a single Harvest Result for this group

Figure 32. Create/Edit Groups

The **Create/Edit Groups** page includes several tabs for adding or editing information about Groups:

- **General** - general information about the Group, such as a name, description, owner, and type
- **Members** - Targets and Groups which are members of this Group

- **Member Of** - Groups which this Group is a member of
- **Profile** - technical instructions on how to harvest the Group
- **Schedule** - dates and times to perform the harvest
- **Annotations** - notes about the Group
- **Description** - metadata about the Group
- **Access** - settings regarding access to the harvested Group

Groups may have a start and end date. This can be used to define groups that are based on events, such as elections. This is particularly relevant to Target Instances, as some harvests of a given Target might belong to a group, while others may not, depending upon the date of the harvest and the interval of the Group.

> *When a start or end date is set, members are only considered part of the Group during that interval. Once the end date has passed, members are not considered to belong to the Group.*

Enter general information about the target

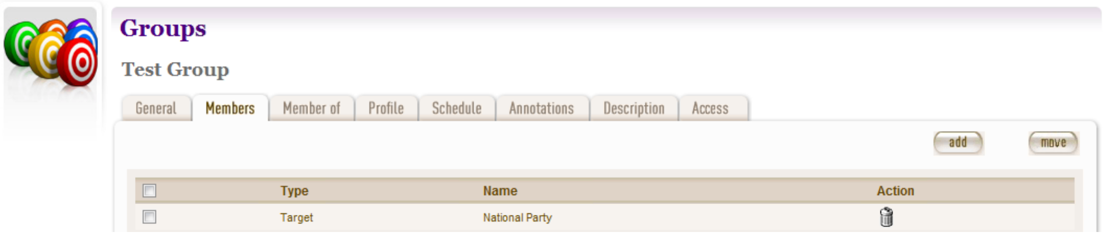
2. On the **General** tab, enter basic information about the Group.
3. If the 'Sub-Group' type is selected in the 'Type' field, a 'Parent Group' field is displayed above the 'Name' field requiring selection of a parent group. Click the add button to add a parent Group.

   > *The Required fields are marked with a red star. When the form is submitted, the system will validate your entries and let you know if you leave out any required information.*

Add the members of the Group

4. Click the **Members** tab.

   The **Members** *tab includes a list of member Targets and Groups and a button to add new members*



Groups

Test Group

□	Type	Name	Action
□	Target	National Party	🗑

   Figure 33. Members tab

5. Click the add button to search for previously created Targets and Groups by name to add to this Group.
6. Select one or more Targets and click the move button to move them to a different Group.

Select a profile and any overrides

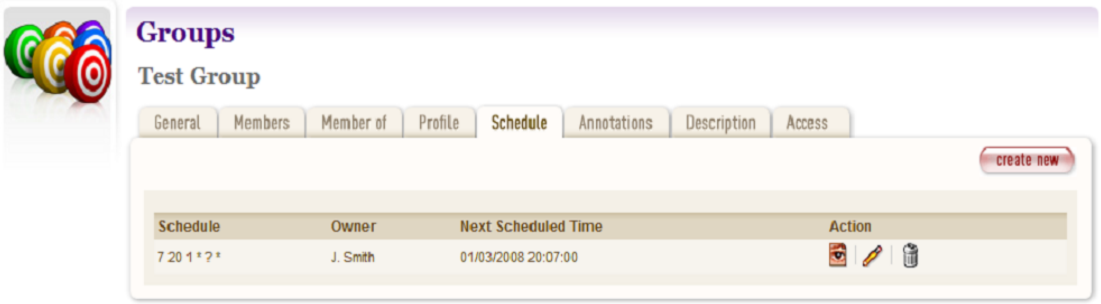
7. Click the **Profile** tab.

   The Profile tab includes a list of harvest profiles, and a series of options to override them. Generally, the default settings are fine.

Enter a schedule for the group

8. Click the **Schedule** tab.

   The **Schedule** *tab includes a list of schedules and a button to create a new schedule.*



Groups

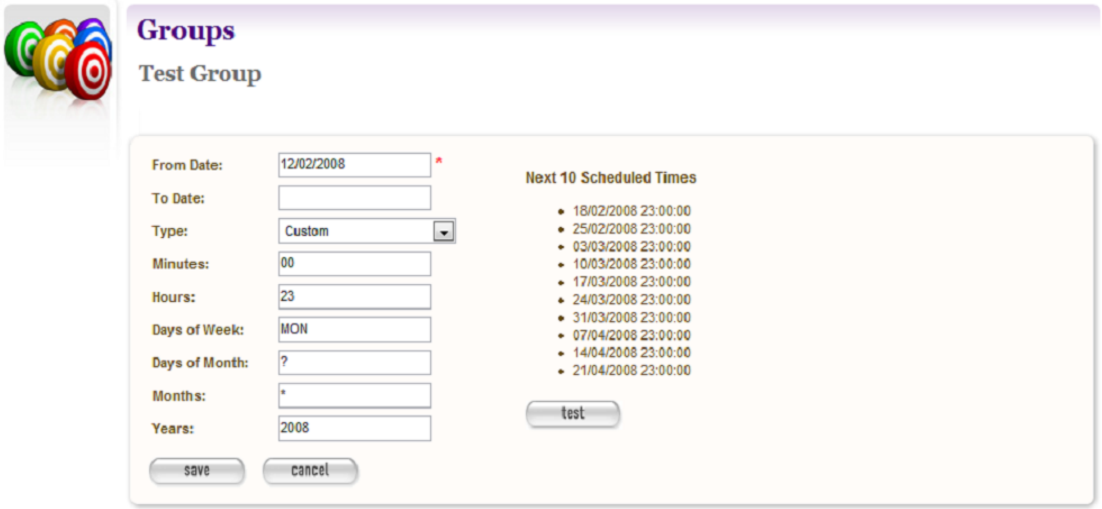
Test Group

Schedule	Owner	Next Scheduled Time	Action
7 20 1 * ? *	J. Smith	01/03/2008 20:07:00	

Figure 34. Schedule tab

9. Click **create new**.

   The **Create/Edit Schedule** *page displays fields for entering a schedule.*



Groups

Test Group

From Date:	12/02/2008 *
To Date:	
Type:	Custom
Minutes:	00
Hours:	23
Days of Week:	MON
Days of Month:	?
Months:	*
Years:	2008

Next 10 Scheduled Times

- 18/02/2008 23:00:00
- 25/02/2008 23:00:00
- 03/03/2008 23:00:00
- 10/03/2008 23:00:00
- 17/03/2008 23:00:00
- 24/03/2008 23:00:00
- 31/03/2008 23:00:00
- 07/04/2008 23:00:00
- 14/04/2008 23:00:00
- 21/04/2008 23:00:00

Figure 35. Create/Edit Schedule

10. Enter **From** and **To** dates for when the harvest will run; select a **Type** of schedule, eg ‘Every Monday at 9:00pm’ or ‘Custom’ - if you select ‘Custom’, enter details of the schedule; and click **Save**.

Annotations

11. Click the **Annotations** tab.

    The **Annotations** *tab allows you to record internal and selection information about the Target. The Annotations are intended for internal use, but are included in submissions to archives.*

    Annotations can be modified or deleted after creation by the user who created them. When an annotation is modified, the annotation date is automatically updated to the time of modification.

Description

12. Click the **Description** tab.

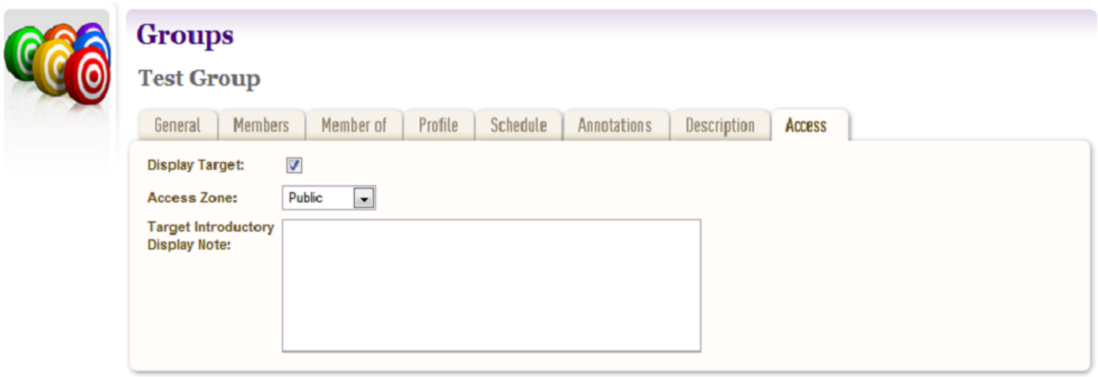
    The **Description** *tab includes a set of fields for storing Dublin Core metadata. This not used in the Web Curator Tool, but is included when any harvests are submitted to a digital archive.*

Access

13. Click the **Access** tab.

    The **Access** *tab allows you to specify a Display Group flag, Display Notes and an Access Zone from*

    - Public(default)
    - Onsite
    - Restricted



Display Target: ☑

Access Zone: Public

Target Introductory Display Note:

Figure 36. Access Tab

Save the completed group

14. Click **save** at the bottom of the page to save the group.

3.8.4 How to edit or view a Group

Editing an existing group is very similar to the process for creating a new record.

To start editing, go to the Group search page, and click the

- **Edit** details

icon from the Actions column. This will load the relevant Group editor. Note that some users will not have access to edit some (or any) Groups.

An alternative to editing a Group is to click the

- **View** details

icon to open the Group viewer. Groups cannot be changed from within the viewer. Once in the Group viewer you may also switch to the editor using the ‘Edit’ button

3.8.5 Harvesting a group

Groups can also be used to synchronise the harvest of multiple related Targets by attaching a schedule to a Group.

Group harvests can be performed in two different ways:

- **Multiple SIP** - Each of the Targets in the Group have multiple Target Instances scheduled with the same harvest start date.
- **Single SIP** - The seed URLs from all the Targets in the Group are combined into a single Target Instance, and are harvested in one operation, quality reviewed in one operation, and submitted to the archive in one operation.

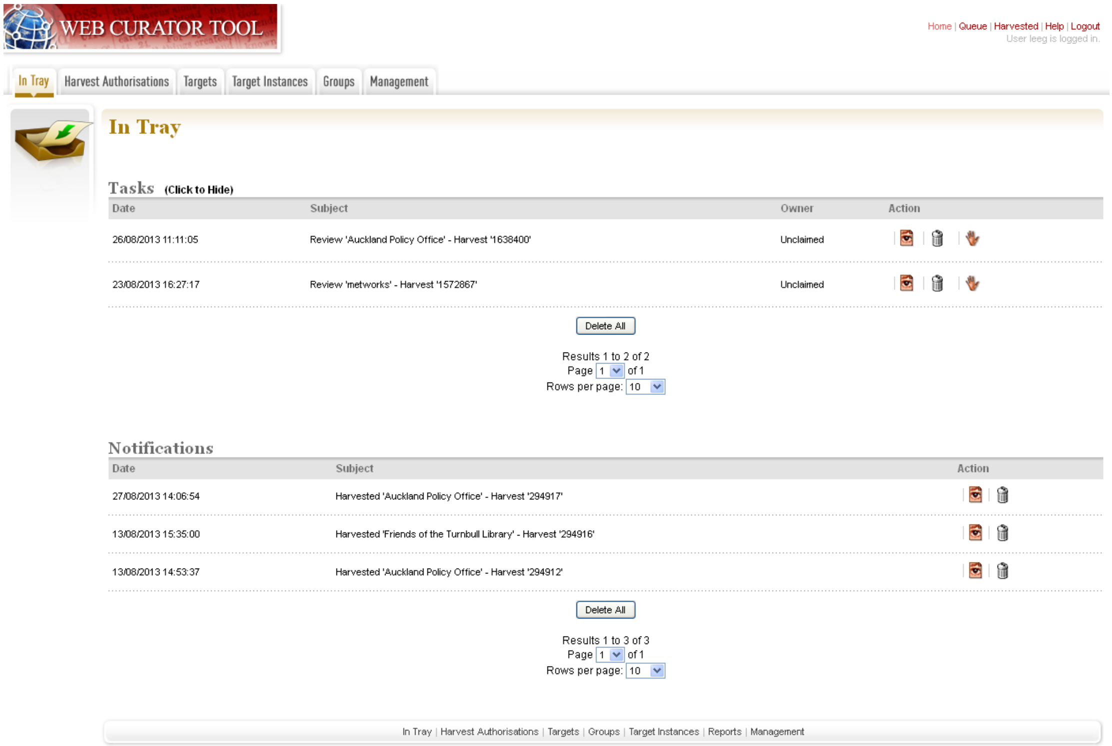
Single SIP harvests are performed using the profile settings and profile override settings for the Group (not the individual Targets).

3.9 The In Tray

3.9.1 Introduction

The **In Tray** is a place where the Web Curator Tool sends you notices and tracks any tasks that have been assigned to you.

The display below shows the *Tasks* and *Notifications* specific to your login. These can also (at your option) be emailed to you.



User leeg is logged in.

In Tray

Tasks (Click to Hide)

Date	Subject	Owner	Action
26/08/2013 11:11:05	Review 'Auckland Policy Office' - Harvest '1638400'	Unclaimed	
23/08/2013 16:27:17	Review 'metworks' - Harvest '1572867'	Unclaimed	

Delete All

Results 1 to 2 of 2
Page 1 of 1
Rows per page: 10

Notifications

Date	Subject	Action
27/08/2013 14:06:54	Harvested 'Auckland Policy Office' - Harvest '294917'	
13/08/2013 15:35:00	Harvested 'Friends of the Turnbull Library' - Harvest '294916'	
13/08/2013 14:53:37	Harvested 'Auckland Policy Office' - Harvest '294912'	

Delete All

Results 1 to 3 of 3
Page 1 of 1
Rows per page: 10

In Tray | Harvest Authorisations | Targets | Groups | Target Instances | Reports | Management

Figure 37. In Tray

> *Note that the* **In Tray** *- and each Web Curator Tool page - has tabs across the top to access the main system functions, which match the icons on the Home Page.*

3.9.2 Tasks

Tasks are events that require action from you (or from someone else with your privileges).

They support workflows where different people are involved at different steps in the harvesting process. For example, the person creating a Target may not be the same as the person who endorses a Target.

For each Task, you can:

👁 - **View** details of the task

🗑 - **Delete** the task

✋ - **Claim** the task (for example, if you are among those who can endorse a harvest, you can claim the task so that you can then perform the endorsement).

☒ - **Un-claim** the task (for example, if you have accidentally claimed a task that is more appropriately carried out by someone else then you can release the task back to the pool of un-claimed tasks for someone else to claim).

Tasks are automatically created, and get automatically deleted once they have been finished (and will then disappear from the In Tray).

There is an option to 'Delete All' if the Tasks list is getting long, but this should only be used if no one in the agency is using the Tasks functionality as part of their workflow, otherwise use the option 'Click to hide' instead.

The different types of Task are outlined below.

Type	Reason	Recipient
Seek Approval	A user has requested someone seek approval for a permission record.	Users with the Confirm Permission privilege.
Endorse Target	A Target Instance needs to be endorsed	Users with the Endorse privilege.
Archive Target	A Target Instance needs to be archived	Users with the Archive privilege.
Approve Target	A Target has been nominated and needs to be approved.	Users with the Approve Target privilege.

3.9.3 Notifications

Notifications are messages generated by the system to tell you about the state of your data. Administrators may also receive notifications about the state of the harvesters.

For each Notification, you can:

👁 - **View** details of the notification

🗑 - **Delete** the notification

The different types of notification are outlined below.

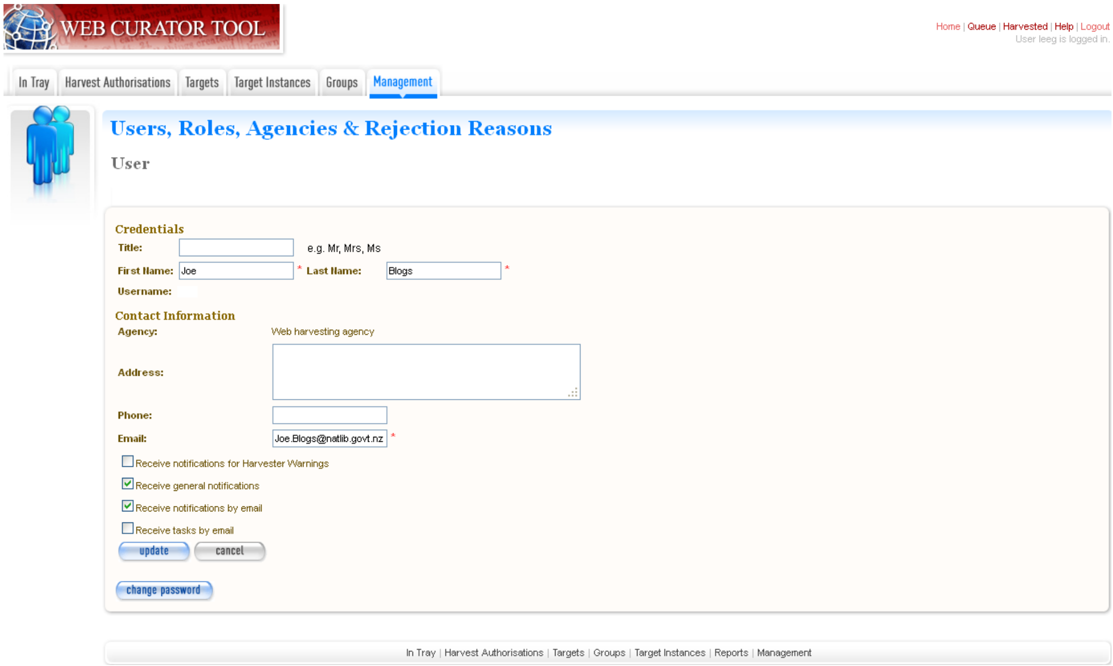
Type	Trigger	Recipient
Harvest Complete	Target Instance has been harvested.	Target Instance Owner
Target Instance Queued	Target Instance has been queued because there is no capacity available.	Target Instance Owner
Target Instance Rescheduled	Target Instance has been delayed 24hrs because the permissions are not approved.	Target Instance Owner
Target Instance Failed	The Target Instance failed to complete	Target Instance Owner
Target Delegated	The ownership of a Target has been delegated.	The new Target Owner
Schedule Added	Someone other than the owner of the Target has added a schedule to it.	Target Owner
Permission Approved	A permission record has been approved.	Owners of Targets associated with the permission.
Permission Rejected	A permission record has been rejected.	Owners of Targets associated with the permission.
Group Changed	A new member has been added to a subgroup.	Owner of the Group
Disk Warning	The disk usage threshold/limit has been reached	Users with Manage Web Harvester privilege
Memory Warning	The memory threshold/limit has been reached.	Users with Manage Web Harvester privilege
Processor Warning	The processor threshold/limit has been reached.	Users with Manage Web Harvester privilege
Bandwidth Warning	The bandwidth limit has been exceeded reached.	Users with Manage Web Harvester privilege

Most notifications are sent only to people within the same Agency. The exception is the system usage warnings that are sent to all users with Manage Web Harvester privilege.

3.9.4 Receive Tasks and Notifications via Email

In your user settings page, the “Receive task notifications by email” setting controls whether notifications and tasks in your In Tray are also emailed to you.

This is useful if, for example, you want to receive an email notification when a harvest finishes.



Users, Roles, Agencies & Rejection Reasons

User

Credentials

Title: e.g. Mr, Mrs, Ms

First Name: Joe * Last Name: Blogs *

Username:

Contact Information

Agency: Web harvesting agency

Address:

Phone:

Email: Joe.Blogs@natlib.govt.nz *

□ Receive notifications for Harvester Warnings

☑ Receive general notifications

☑ Receive notifications by email

□ Receive tasks by email

Figure 38. User settings

3.10 User, Roles, Agencies, Rejection Reasons & QA Indicators

3.10.1 Introduction

The Web Curator Tool has a flexible system of users, permissions, roles and agencies. Each user belongs to an agency, and has a number of roles that define the access individual users have to Web Curator Tool functionality.

In this chapter we refer to administrative users, who are those users that can register other users, manage user accounts, assign roles to users, and adjust the system's configuration. However, in the Web Curator Tool, an administrative user is simply a user who has been assigned a role like "System Administrator" or "Agency Administrator", and the exact responsibilities of these roles (and even their names) will likely vary between institutions.

3.10.2 Users

Each user has a Web Curator Tool account, which includes some basic identifying information and some preferences.

Each user is also assigned one or more roles. Roles are sets of Web Curator Tool privileges that restrict the access individual users have to Web Curator Tool functionality.

3.10.3 Roles

A role is a way of capturing a set of privileges and responsibilities that can be assigned to sets of Web Curator Tool Users. Each role has a set of privileges attached. Users who are assigned the role will be given permission to perform operations.

Most privileges can be adjusted to three levels of scope: **All**, **Agency**, or **Owner**. If the scope of an active permission is set to **All** then the permission applies to all objects; if it is set to **Agency** then it applies only to those objects that belong to the same agency as the user; if it is set to **Owner** it applies only to those owned by that user.

3.10.4 Agencies

An agency is an organisation who is involved in harvesting websites using the tool. Users and roles are defined for an agency scope and Targets, Groups and Harvest Authorisations are also owned at Agency level. This provides a convenient way of managing access to the tool for multiple organisations.

3.10.5 Harvest authorisation privileges

The permissions that control access to the harvest authorisation module are listed in the Role editing page in the **Manage Copying Permissions and Access Rights** section.

They are:

- Create Harvest Authorisations
- Modify Harvest Authorisations
- Confirm Permissions
- Modify Permissions
- Transfer Linked Targets
- Enable/Disable Harvest Authorisations
- Generate Permission Requests

3.10.6 Target privileges

The permissions that control access to the Target module are listed in the Role editing page in the **Manage Targets** section.

They are:

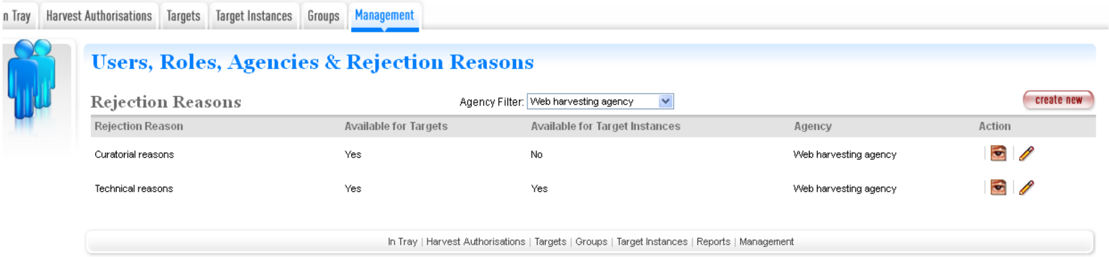
- Create Target - The user can create new Targets.
- Modify Target - The user can modify existing Targets.
- Approve Target - The user can Approve a Target.
- Cancel Target - The user can Cancel a Target.
- Delete Target - The user can Delete a Target (but only if that Target has no associated Target Instances).
- Reinstate Target - The user can reinstate a Target that is in the Cancelled or Completed state.
- Add Schedule to Target - The user can attaché a schedule to a Target.
- Set Harvest Profile Level 1 - The user can attach a profile to the Target from among the level 1 profiles.
- Set Harvest Profile Level 2 - The user can attach a profile to the Target from among the level 1 and level 2 profiles.
- Set Harvest Profile Level 3 - The user can attach a profile to the Target from among all the profiles.

  Other privileges within the Roles include the ability to manage Rejection Reasons, QA indicators and Flags. This is more of an administrative role.

3.10.7 Rejection Reasons

When a target or a target instance is rejected there needs to be a reason for it. E.g. you might want to reject a target for curatorial reasons or you might actually want to select a target for curatorial reasons, but cannot do so for technical reasons and therefore you reject it for technical reasons.

If you have an external report writer it's possible to run a report for targets that have been rejected for a specific reason.


n Tray
Harvest Authorisations
Targets
Target Instances
Groups
Management
create new
In Tray | Harvest Authorisations | Targets | Groups | Target Instances | Reports | Management

Users, Roles, Agencies & Rejection Reasons

Rejection Reasons

Agency Filter: Web harvesting agency

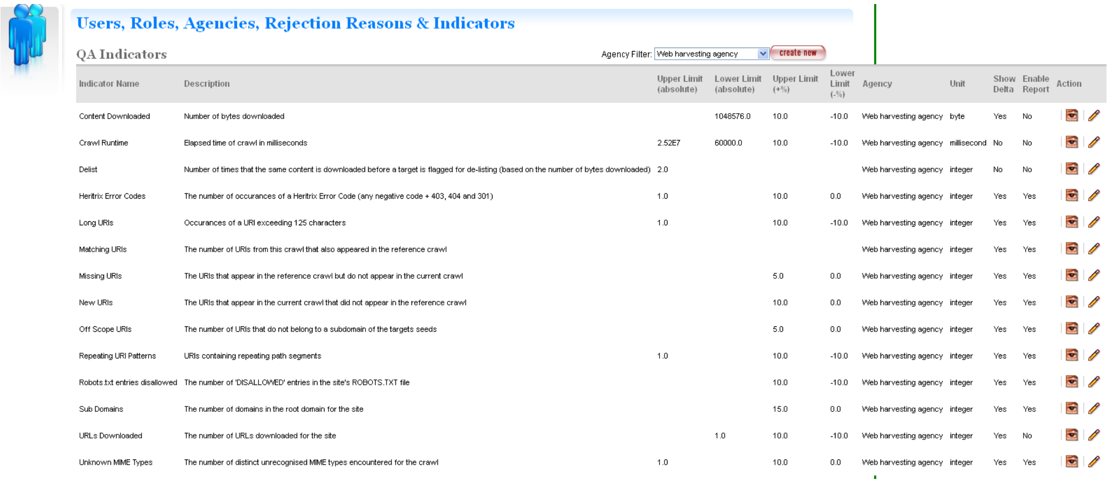
Rejection Reason	Available for Targets	Available for Target Instances	Agency	Action
Curatorial reasons	Yes	No	Web harvesting agency	
Technical reasons	Yes	Yes	Web harvesting agency	

Figure 39. Rejection Reasons

3.10.8 QA Indicators

The QA indicators are designed to assist a user to determine whether a harvested TI requires quality review or can be archived/delisted based on a number of indicators. Recommendations are viewed in the Target Instance Summary for a TI once the TI has been harvested.

The indicators below have been pre-populated by a template that can be installed when WCT is set up.



Users, Roles, Agencies, Rejection Reasons & Indicators

QA Indicators

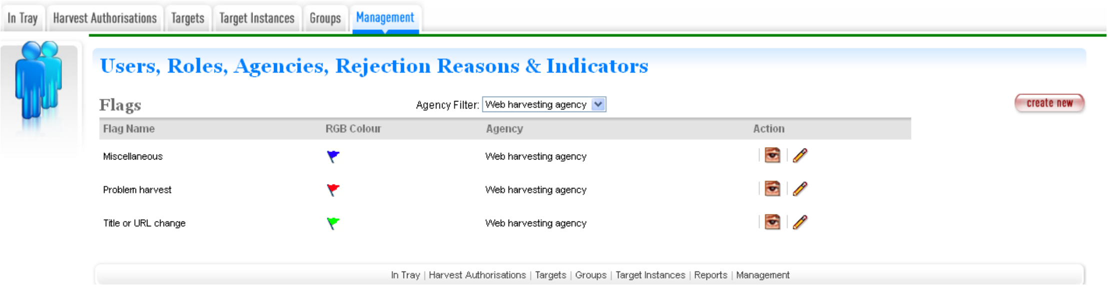
Agency Filter: Web harvesting agency

Indicator Name	Description	Upper Limit (absolute)	Lower Limit (absolute)	Upper Limit (+%)	Lower Limit (-%)	Agency	Unit	Show Delta	Enable Report	Action
Content Downloaded	Number of bytes downloaded		1048576.0	10.0	-10.0	Web harvesting agency	byte	Yes	No	
Crawl Runtime	Elapsed time of crawl in milliseconds	2.52E7	60000.0	10.0	-10.0	Web harvesting agency	millisecond	No	No	
Delist	Number of times that the same content is downloaded before a target is flagged for de-listing (based on the number of bytes downloaded)	2.0				Web harvesting agency	integer	No	No	
Heritrix Error Codes	The number of occurances of a Heritrix Error Code (any negative code + 403, 404 and 301)	1.0		10.0	0.0	Web harvesting agency	integer	Yes	Yes	
Long URIs	Occurances of a URI exceeding 125 characters	1.0		10.0	-10.0	Web harvesting agency	integer	Yes	Yes	
Matching URIs	The number of URIs from this crawl that also appeared in the reference crawl					Web harvesting agency	integer	Yes	Yes	
Missing URIs	The URIs that appear in the reference crawl but do not appear in the current crawl			5.0	0.0	Web harvesting agency	integer	Yes	Yes	
New URIs	The URIs that appear in the current crawl that did not appear in the reference crawl			10.0	0.0	Web harvesting agency	integer	Yes	Yes	
Off Scope URIs	The number of URIs that do not belong to a subdomain of the targets seeds			5.0	0.0	Web harvesting agency	integer	Yes	Yes	
Repeating URI Patterns	URIs containing repeating path segments	1.0		10.0	-10.0	Web harvesting agency	integer	Yes	Yes	
Robots.txt entries disallowed	The number of 'DISALLOWED' entries in the site's ROBOTS.TXT file			10.0	-10.0	Web harvesting agency	integer	Yes	Yes	
Sub Domains	The number of domains in the root domain for the site			15.0	0.0	Web harvesting agency	integer	Yes	Yes	
URLs Downloaded	The number of URLs downloaded for the site		1.0	10.0	-10.0	Web harvesting agency	integer	Yes	No	
Unknown MIME Types	The number of distinct unrecognised MIME types encountered for the crawl	1.0		10.0	0.0	Web harvesting agency	integer	Yes	Yes	

Figure 40. QA Indicators

3.10.9 Flags

Flags provide the ability to highlight a target instance so that action can be taken. They are set within an agency so all the users of that agency share the same flags. E.g. an agency might want to flag TI's that have harvesting issues so that an analyst can investigate them.



Users, Roles, Agencies, Rejection Reasons & Indicators

Flags

Agency Filter: Web harvesting agency

Flag Name	RGB Colour	Agency	Action
Miscellaneous	⚑	Web harvesting agency	👁 ✎
Problem harvest	⚑	Web harvesting agency	👁 ✎
Title or URL change	⚑	Web harvesting agency	👁 ✎

In Tray | Harvest Authorisations | Targets | Groups | Target Instances | Reports | Management

Figure 41. Flags

3.11 Reports

3.11.1 Introduction

The Reports screen gives users access to several types of report.

3.11.2 System usage report

The System Usage Report is a report based on the audit records that lists the usage sessions for a user (or group of users) over a selected period.

The criteria for the report are:

- Start Date;
- End Date;
- Agency (optional).

The report will take data from the audit log table and logon duration tables in the database. Note that the logon times displayed are estimates and may not be completely accurate.

3.11.3 System activity report

The System Activity Report is a report based on the audit records. The criteria for the report are:

- Start Date;
- End Date;
- Agency (optional);
- User (optional).

This report will directly take information out of the audit log table in the database. The following information extracted from the audit log:

- User ID
- Username
- User Real Name (First name plus surname)
- Activity type
- Subject Identifier number

- Message text, which gives an English description of the action.

3.11.4 Crawler activity report

The crawler activity report allows administrators to get a summary of all the crawling activity undertaken by the Web Curator Tool for a specified period.

The report has the following parameters:

- **Start date**: a date and time (to the nearest second)
- **End date**: a date and time (to the nearest second)
- **Agency** (optional).
- **User** (optional);

The report finds all Target Instances where:

- The State is other than "Scheduled" or "Queued" (i.e. they have been sent to a crawler), and
- The period when the crawl was running overlaps the interval defined by the start date and end-date parameters.

The output includes the following fields: Identifier, Target Name, status, start date, end date (if known), crawl duration, bytes downloaded, harvest agent.

3.11.5 Target/Group Schedules report

The Target/Group Schedules report is a report showing the harvest schedules for 'Approved' Targets and/or Groups.

The report has the following filter parameters:

- Agency (optional)
- User (optional)
- Target Type (optional)

The report details the schedules of all Targets and/or Groups where:

- The State is "Approved" (for Targets) or "Active" (for Groups).

The output includes the following fields: Target/Group ID, Type (Target or Group), Name, Agency, Owner, From Date, To Date (if known) and Schedule Type followed by schedule type specific details.

3.11.6 Summary Target Schedules report

The Summary Target Schedules report is a summary report of the harvest schedules for 'Approved' Targets and/or Groups.

The report has the following filter parameters:

- Agency (optional)

The report details the numbers of schedules of particular types for all Targets and/or Groups where:

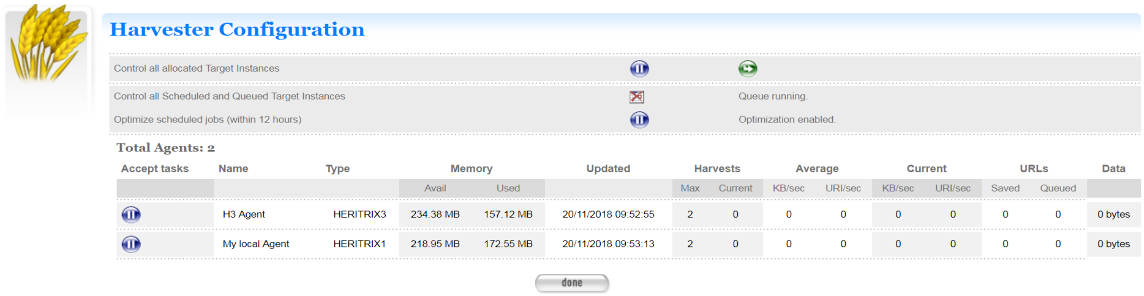
- The State is "Approved" (for Targets) or "Active" (for Groups).

The output includes the counts of all known schedule types for the selected agency or all agencies.

3.12 Harvester Configuration

3.12.1 Introduction

The **Harvester Configuration** can be found in the General tab of the Management section. It enables the user to view the current status of the harvesters and allows a certain level of control over the harvesting schedule.



Harvester Configuration

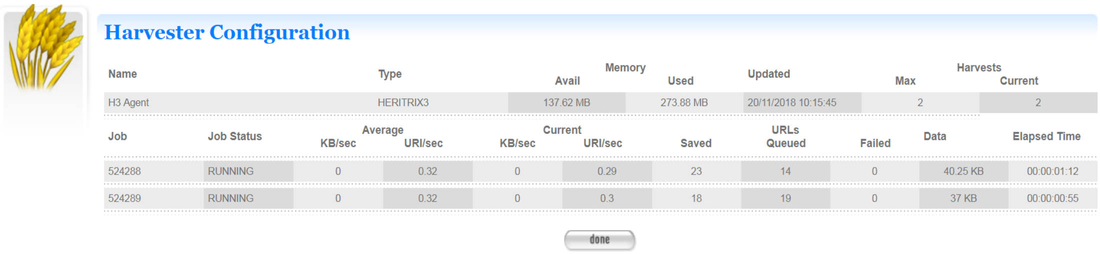
Control all allocated Target Instances		
Control all Scheduled and Queued Target Instances		Queue running.
Optimize scheduled jobs (within 12 hours)		Optimization enabled.

Total Agents: 2

Accept tasks	Name	Type	Memory		Updated	Harvests		Average		Current		URLs		Data
			Avail	Used		Max	Current	KB/sec	URI/sec	KB/sec	URI/sec	Saved	Queued	
	H3 Agent	HERITRIX3	234.38 MB	157.12 MB	20/11/2018 09:52:55	2	0	0	0	0	0	0	0	0 bytes
	My local Agent	HERITRIX1	218.95 MB	172.55 MB	20/11/2018 09:53:13	2	0	0	0	0	0	0	0	0 bytes

Figure 42. Harvester Configuration

If you click on the name of the harvester you can see which jobs are currently running. The numbers under **Job** refers to the target instance that is currently running.



Harvester Configuration

Name	Type	Memory		Updated	Harvests	
		Avail	Used		Max	Current
H3 Agent	HERITRIX3	137.62 MB	273.88 MB	20/11/2018 10:15:45	2	2

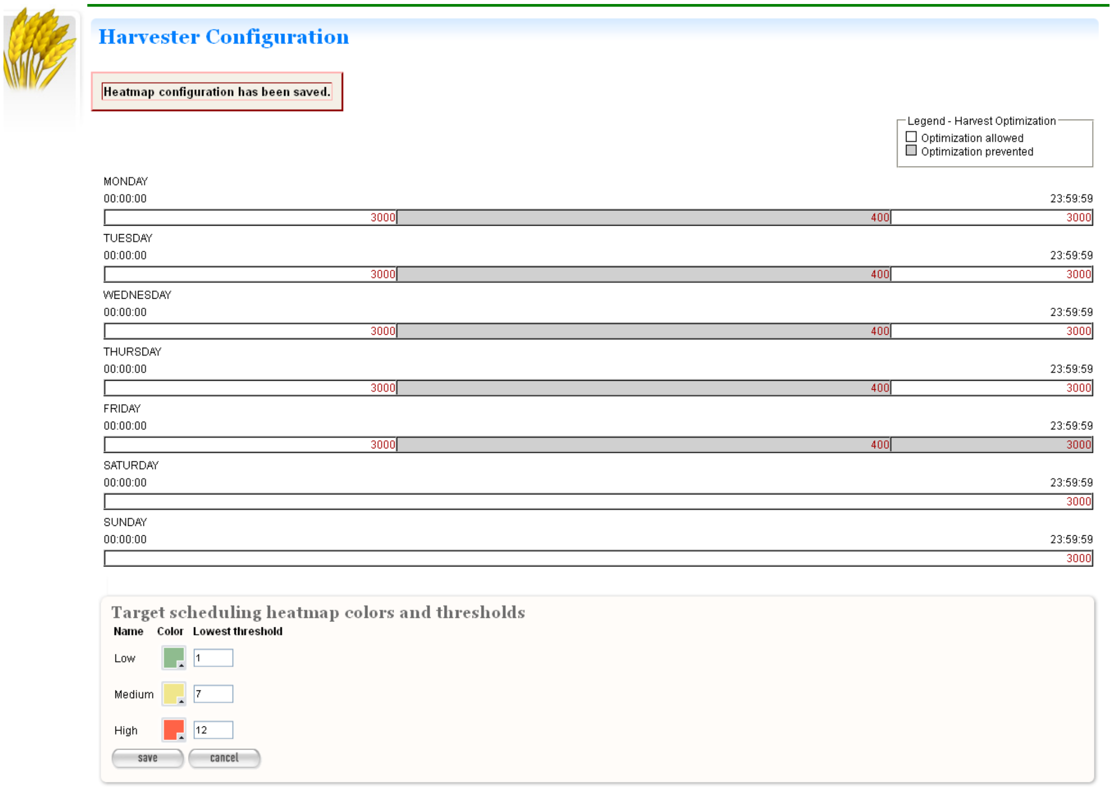
Job	Job Status	Average		Current		URLs			Data	Elapsed Time
		KB/sec	URI/sec	KB/sec	URI/sec	Saved	Queued	Failed		
524288	RUNNING	0	0.32	0	0.29	23	14	0	40.25 KB	00:00:01:12
524289	RUNNING	0	0.32	0	0.3	18	19	0	37 KB	00:00:00:55

Figure 43. Shows the number of jobs running on a particular harvester

3.12.2 Bandwidth limits

Bandwidth limits must be created before any harvesting can be undertaken. The default setting is ‘0’. Bandwidth will be allocated to a harvest as a percentage of the allowed bandwidth for the period.

In figure 36 the bandwidth has been set to run at a reduced rate during the day and run at a higher level in the evenings and weekends.



Harvester Configuration

Heatmap configuration has been saved.

Legend - Harvest Optimization
□ Optimization allowed
▨ Optimization prevented

MONDAY
00:00:00 23:59:59
3000 400 3000

TUESDAY
00:00:00 23:59:59
3000 400 3000

WEDNESDAY
00:00:00 23:59:59
3000 400 3000

THURSDAY
00:00:00 23:59:59
3000 400 3000

FRIDAY
00:00:00 23:59:59
3000 400 3000

SATURDAY
00:00:00 23:59:59
3000

SUNDAY
00:00:00 23:59:59
3000

Target scheduling heatmap colors and thresholds

Name	Color	Lowest threshold
Low	🟩	1
Medium	🟨	7
High	🟥	12

Figure 44. Bandwidth limits, harvest optimisation and heatmap

In figure 44 above if you click on the hyperlinked numbers you can choose to optimise your harvests at particular times of the day or week earlier than the schedule otherwise permits. The window for this look-ahead is configurable and defaults to 12 hours. This example shows that optimisation has been set for evenings and weekends.

You will also need to check the 'harvest optimization' button on the target schedule. If you need to run a harvest at a specified time then simply leave the 'harvest optimization' button on the target record unchecked.

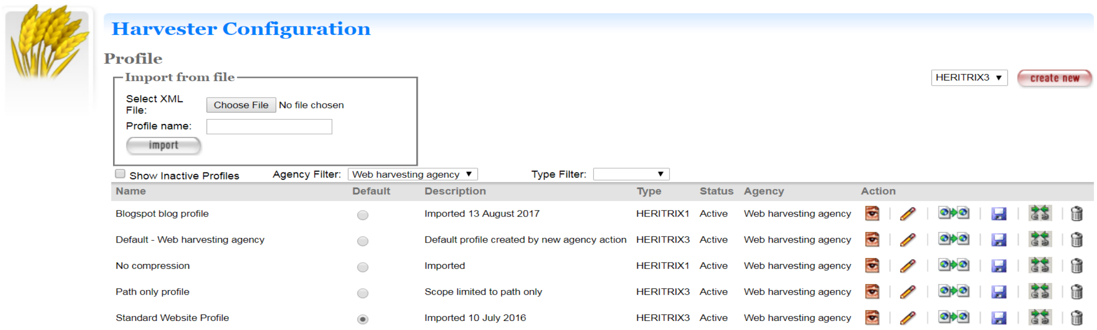
If you need to disable this feature temporarily you can do so from the Harvester Configuration general screen. Simply click on Optimize scheduled jobs button to disable and then click again when you want to enable the functionality again.

The heatmap threshold can be changed to suit what you consider to be your low, medium or high harvesting levels.

3.12.3 Profiles

The WCT profile contains settings that control how a harvest behaves. The settings for WCT profiles are based on Heritrix profiles. Profiles can be created to crawl particular kinds of websites, such as blogs.

You manage profiles from the Profiles search page:



Harvester Configuration

Profile

Import from file

Select XML File: Choose File No file chosen

Profile name:

import

HERITRIX3 ▾ create new

□ Show Inactive Profiles Agency Filter: Web harvesting agency ▾ Type Filter: ▾

Name	Default	Description	Type	Status	Agency	Action
Blogspot blog profile	○	Imported 13 August 2017	HERITRIX1	Active	Web harvesting agency	👁 ✎ ⇄ 💾 ⇆ 🗑
Default - Web harvesting agency	○	Default profile created by new agency action	HERITRIX3	Active	Web harvesting agency	👁 ✎ ⇄ 💾 ⇆ 🗑
No compression	○	Imported	HERITRIX1	Active	Web harvesting agency	👁 ✎ ⇄ 💾 ⇆ 🗑
Path only profile	○	Scope limited to path only	HERITRIX3	Active	Web harvesting agency	👁 ✎ ⇄ 💾 ⇆ 🗑
Standard Website Profile	●	Imported 10 July 2016	HERITRIX3	Active	Web harvesting agency	👁 ✎ ⇄ 💾 ⇆ 🗑

Figure 45. Profile search page

You can import a profile from an existing XML file. Once a profile is imported you will need to rename it, otherwise it will be called 'Profile Imported on. . . '

Or you can **create new** profiles

There are actions, with options to:

👁 - **View** the profile

✎ - **Edit** the profile

⇄ - **Copy** the profile and create a new one

💾 - **Export** a copy of the profile

⇆ - **Transfer** targets associated with one profile to another profile

🗑 - **Delete** profile. Profiles can only be deleted if they have no target instances associated with the profile.

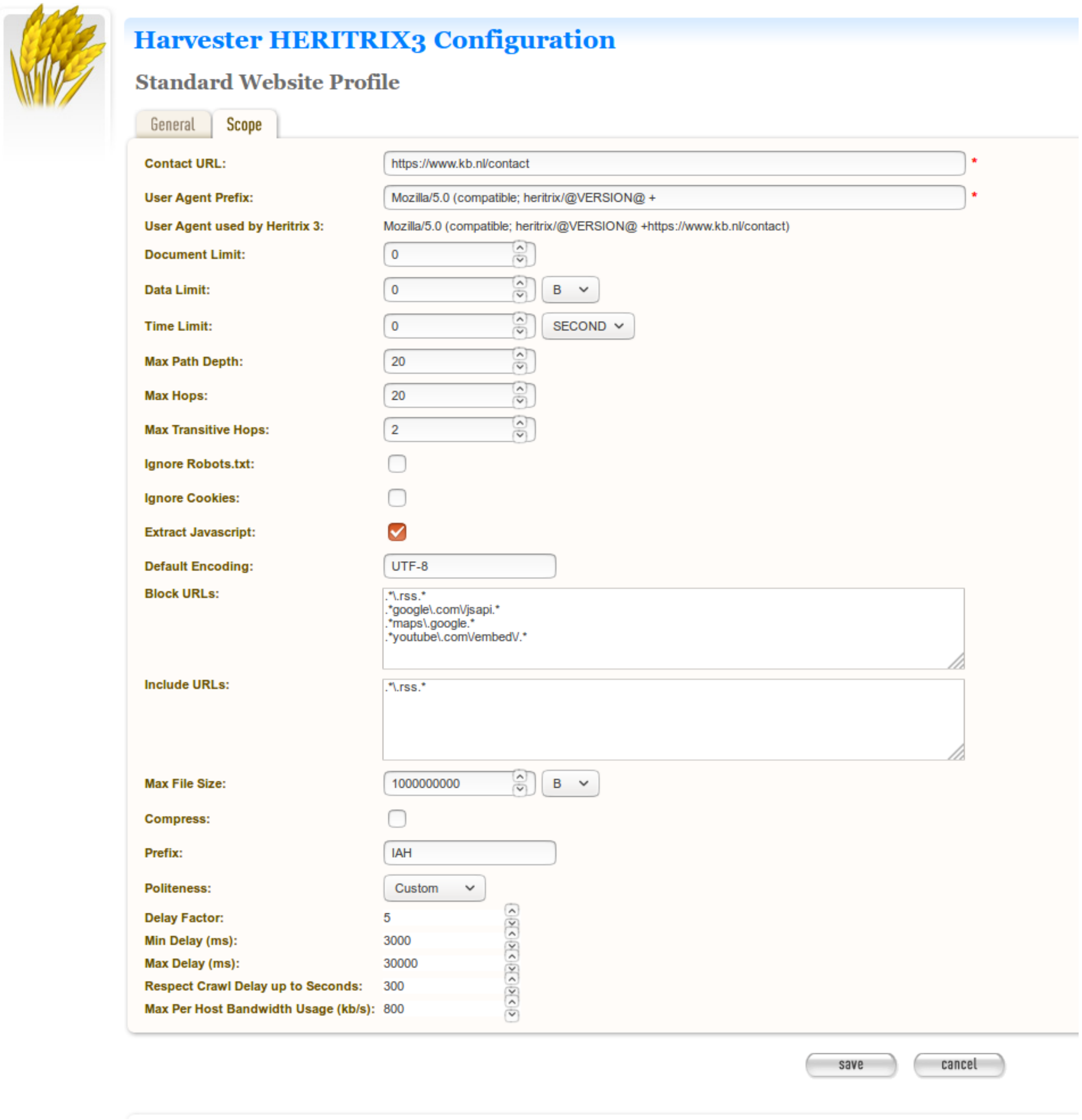
3.12.4 How to create a profile

From the **Profile** page

1. Select the harvester type (Heritrix 3, Heritrix 1)
2. Click **create new**
3. The **Create/Edit profile** page displays

The **Create/Edit profile** page includes several tabs for adding or editing information about profiles.

Heritrix 3:



Harvester HERITRIX3 Configuration

Standard Website Profile

Contact URL:	https://www.kb.nl/contact *
User Agent Prefix:	Mozilla/5.0 (compatible; heritrix/@VERSION@ + *
User Agent used by Heritrix 3:	Mozilla/5.0 (compatible; heritrix/@VERSION@ +https://www.kb.nl/contact)
Document Limit:	0
Data Limit:	0 B
Time Limit:	0 SECOND
Max Path Depth:	20
Max Hops:	20
Max Transitive Hops:	2
Ignore Robots.txt:	☐
Ignore Cookies:	☐
Extract Javascript:	☑
Default Encoding:	UTF-8
Block URLs:	.*\.rss.* .*google\.com\/jsapi.* .*maps\.google.* .*youtube\.com\/embed\/.*
Include URLs:	.*\.rss.*
Max File Size:	1000000000 B
Compress:	☐
Prefix:	IAH
Politeness:	Custom
Delay Factor:	5
Min Delay (ms):	3000
Max Delay (ms):	30000
Respect Crawl Delay up to Seconds:	300
Max Per Host Bandwidth Usage (kb/s):	800

Figure 46. Profile page

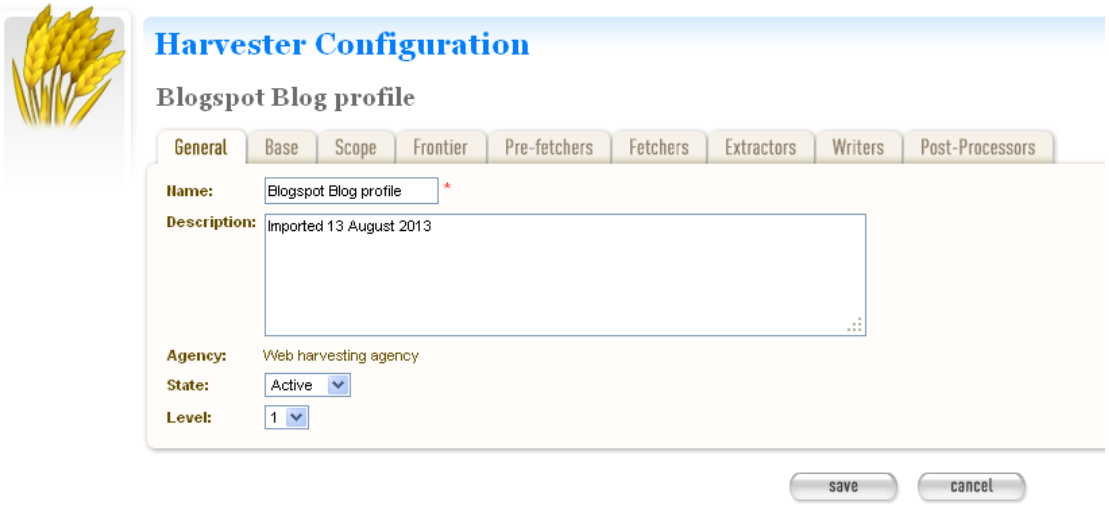
- **General** - general information about the profile, such as a name, description, agency, whether it's an active or inactive profile and what level the profile should be set.
- **Scope** - settings that decide the general crawl parameters. This is a simplified set of availble Heritrix 3 settings.

Parameter	Description
Contact URL	A contact URL for the person or entity running the crawl.
User Agent Prefix	The first piece of text that comprises the final User Agent string that Heritrix 3 will use. Be sure to replace the *@VERSION@* text with the Heritrix version you are using.
Document Limit	The maximum number of documents to harvest during the crawl. Once the document count has exceeded this limit, Heritrix will stop the crawl. A value of zero means no upper limit.
Data Limit	The maximum file size to write to disk. Once the size of all files on disk has exceeded this limit, Heritrix will stop the crawl. A value of zero means no upper limit.
Time Limit	The maximum duration for the crawl to run. Once the duration has exceeded this limit, Heritrix will stop the crawl. A value of zero means no upper limit.
Max Path Depth	Reject any URI whose total number of path-segments is over the configured threshold. A path-segment is a string in the URI separated by a "/" character, not including the first "//".
Max Hops	The maximum number of allowed hops the crawler should go when crawling linked pages.
Max Transitive Hops	The maximum number of non-navlink hops followed in the path from the original seed.
Ignore Robots.txt	Do not obey a seed's robots.txt.
Ignore Cookies	Disable cookie handling.
Extract Javascript	Toggle the extraction of URLs from javacript code.
Default Encoding	The character encoding to use for files that do not have one specified in the HTTP response headers. The default is UTF-8
Block URLs	Block all URIs matching the regular expression from being processed.
Include URLs	Allow all URIs matching the regular expression to be processed.
Max File Size	The maximum size in bytes for each WARC file. Once the WARC file reaches this size, no URIs will be written to it and another WARC file will be created to handle the remaining URIs.
Compress	Compress the WARC file content using gzip compression. Note that compression applies to each content item stored in the WARC.
Prefix	The prefix of the WARC filename.
Politeness	The politeness settings are a set of parameters that control how fast Heritrix tries to crawl a website. There are three preset options (Polite, Medium and Aggressive). To edit the individual values, choose 'Custom'.

For more information about configuring profiles see: https://github.com/internetarchive/heritrix3/wiki/Processing%20Chains https://github.com/internetarchive/heritrix3/wiki/Processor%20Settings https://github.com/internetarchive/heritrix3/wiki/Configuring%20Jobs%20and%20Profiles https:

//github.com/internetarchive/heritrix3/wiki/Basic%20Crawl%20Job%20Settings

Heritrix 1:



Harvester Configuration

Blogspot Blog profile

Name: Blogspot Blog profile *

Description: Imported 13 August 2013

Agency: Web harvesting agency

State: Active

Level: 1

Figure 46. Profile page

- **General** - general information about the profile, such as a name, description, agency, whether it's an active or inactive profile and what level the profile should be set.
- **Base** - Information about the crawl order, user-agent string, and robots honouring policy.
- **Scope** - settings that decide for each discovered URI if it's within the scope of the current crawl. Several scopes are provided with Heritrix such as DecidingScope, PathScope and HostScope
- **Frontier** - this maintains the internal state of the crawl. It effects the order in which the URIs are crawled

The remaining tabs **Pre-fetchers**, **Fetchers**, **Extractors**, **Writers**, and **Post-Processors** are a series of processors that a URI passes through when it is crawled.

For more information about creating profiles see: http://crawler.archive.org/articles/user_manual/creating.html

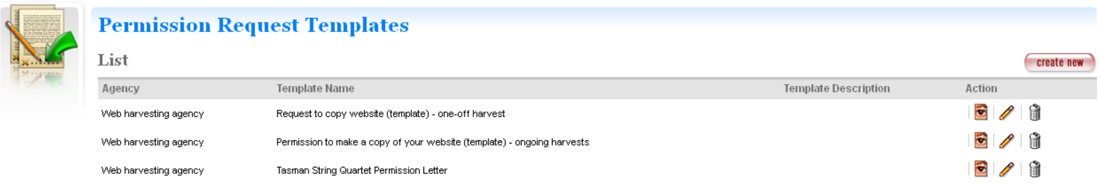
For more information about configuring profiles see: http://crawler.archive.org/articles/user_manual/config.html

3.13 Permission Request Templates

3.13.1 Introduction

The **Permission Request Templates** can be found in the Management section. It enables the user with the appropriate role to open an existing permission template, or add a new one to the list.

You can choose whether to use a generic template with information that can be attached to any harvest authorisation or set up a new one each time if specific information is required.



Permission Request Templates

List

Agency	Template Name	Template Description	Action
Web harvesting agency	Request to copy website (template) - one-off harvest		
Web harvesting agency	Permission to make a copy of your website (template) - ongoing harvests		
Web harvesting agency	Tasman String Quartet Permission Letter		

Figure 47. Permission request templates

Some agencies prefer to handle Permission requests outside of WCT and simply add the file number to the Harvest Authorisation once permission is granted.

3.14 HTML Serials

3.14.1 Introduction

Online serials in HTML format can harvested using WCT and archived as individual issues.

The National Library of New Zealand introduced this functionality when they discovered serials that were previously issued as PDFs were being issued online solely in HTML format. HTML serials functionality is closely tied in with using the Rosetta preservation system however,[2] so if you want to use this option and you're not using Rosetta, you will need to investigate alternative delivery options that allow you to view serials by issue date rather than harvest date.

HTML Serials can be set up as a separate agency within WCT. A user can only be a member of one agency, so it works best if one team does HTML serial harvesting while another team does web harvesting. If users do both then they will need to login with a different username and password for one of the agencies.

The workflow is similar to the web harvesting workflow. The target record is created for the serial. The seed URL is likely to change with each new issue. Because of this it is standard practice to use 'harvest now' rather than create ongoing schedules.

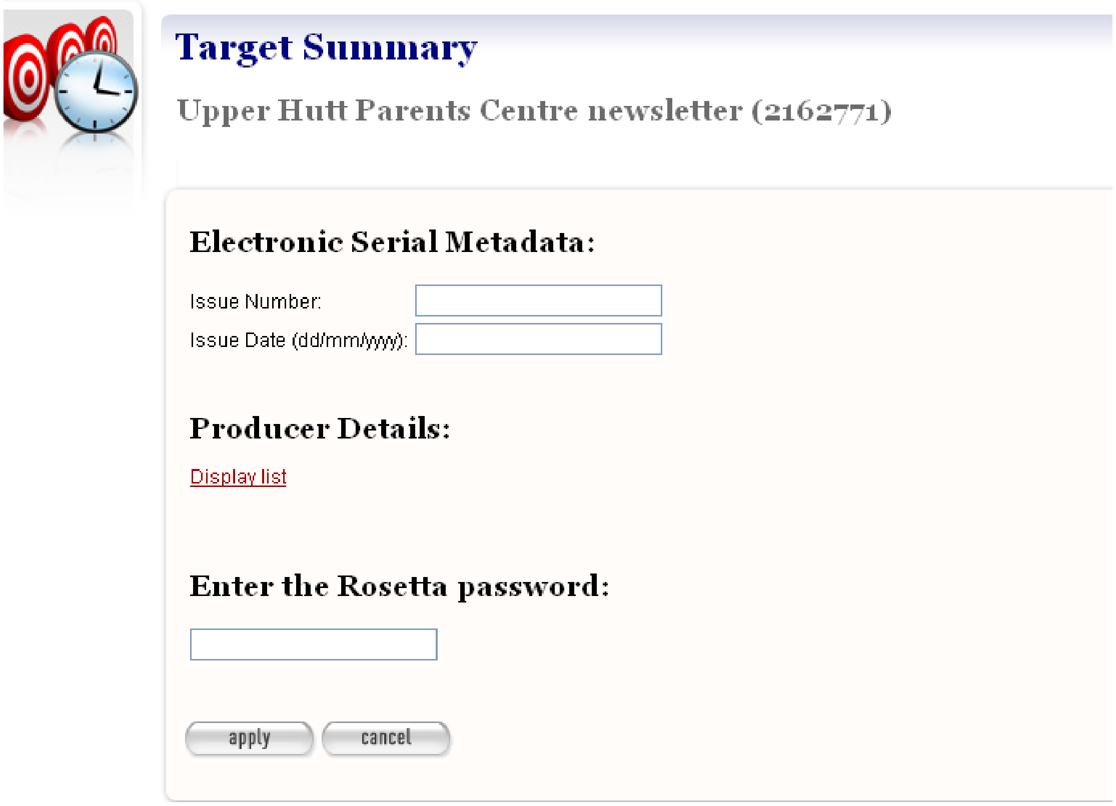
The HTML serials standard profile using Heritrix 1 is a pathscope profile.

The new QA Indicators are designed for websites so it's best to use the log files and tree view to quality review the harvested serial issue.

Once the serial issue has been harvested and is ready for archiving you can endorse the harvest. If you don't use Rosetta you can simply archive the serial. If you do use Rosetta you will see a 'next' button pop up (see figure 48 below). The National Library uses this metadata form to link the HTML serial with the producer record in the preservation system as well as add the issue number and issue date.

In Rosetta it's necessary to distinguish the HTML serials ingest from the web harvesting workflow so that the appropriate viewer is used. To do this the Target record description tab has eSerial set as a default in the HTML serials agency. The viewer in the archive will then display the serial by issue number and date.

[2] For information about the Rosetta preservation system visit: https://www.exlibrisgroup.com/products/rosetta-digital-asset-management-and-preservation/



Target Summary

Upper Hutt Parents Centre newsletter (2162771)

Electronic Serial Metadata:

Issue Number:

Issue Date (dd/mm/yyyy):

Producer Details:

Display list

Enter the Rosetta password:

Figure 48. Metadata for depositing a serial issue to Rosetta

3.15 Workflow

3.15.1 Minimal workflow

The basic workflow for harvesting a website with the Web Creator Tool is:

1. Obtain **Harvest Authorization** for the harvest and record it in a permission record.
2. Create a **Target** that defines the web material you want to harvest, technical harvest parameters and schedules for harvesting.
3. **Approve** the Target.
4. *The Web Curator Tool will create* **Target Instances** *according to your schedule, run the harvests for you, and notify you that the Target Instance is in the* **Harvested** *state and ready for review.*
5. **Quality Review** the Target Instance, then **endorse** the results.
6. Submit the harvest to a digital archive.

These steps do not always have to be performed in order, though there are some constraints on how the tasks can be performed, as outlined below.

Step	Prerequisites
1. Obtain Harvest Authorization	
2. Create a Target	
3. Approve the Target	Harvest authorisation created, Seed URLs linked to permission records.
4. Run harvests	Seed URLs linked to permission records that have been granted.
5. Quality review and endorse	Harvest has been run.
6. Submit to archive	Harvest result is endorsed.

3.15.2 General workflow example

The following diagram illustrates a possible flow of authorisations, Targets, and harvests in an institution that requires users to seek permission before initiating any harvests:

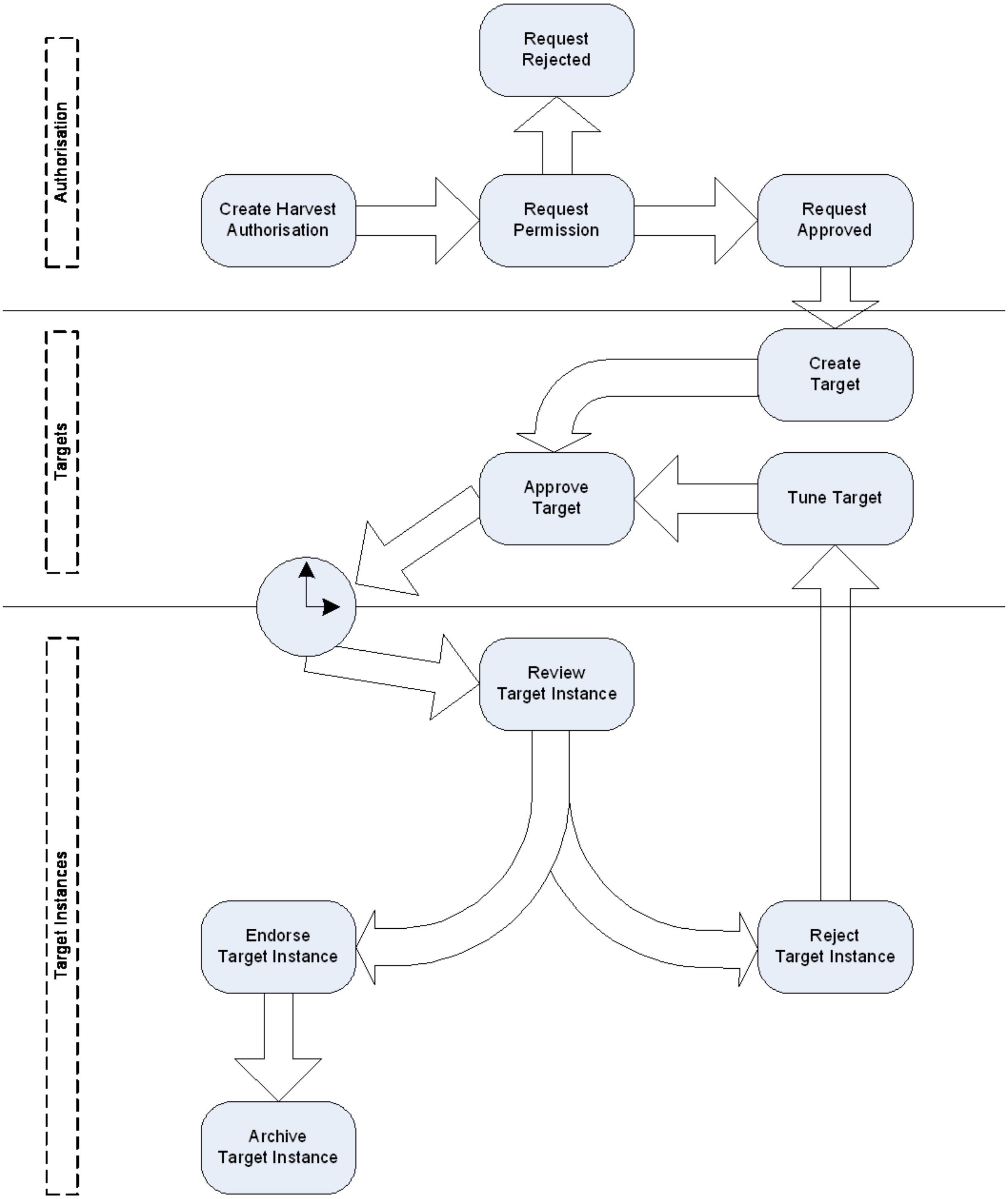


Figure 27. Web Curator Tool process flow

3.15.3 Detailed workflow example

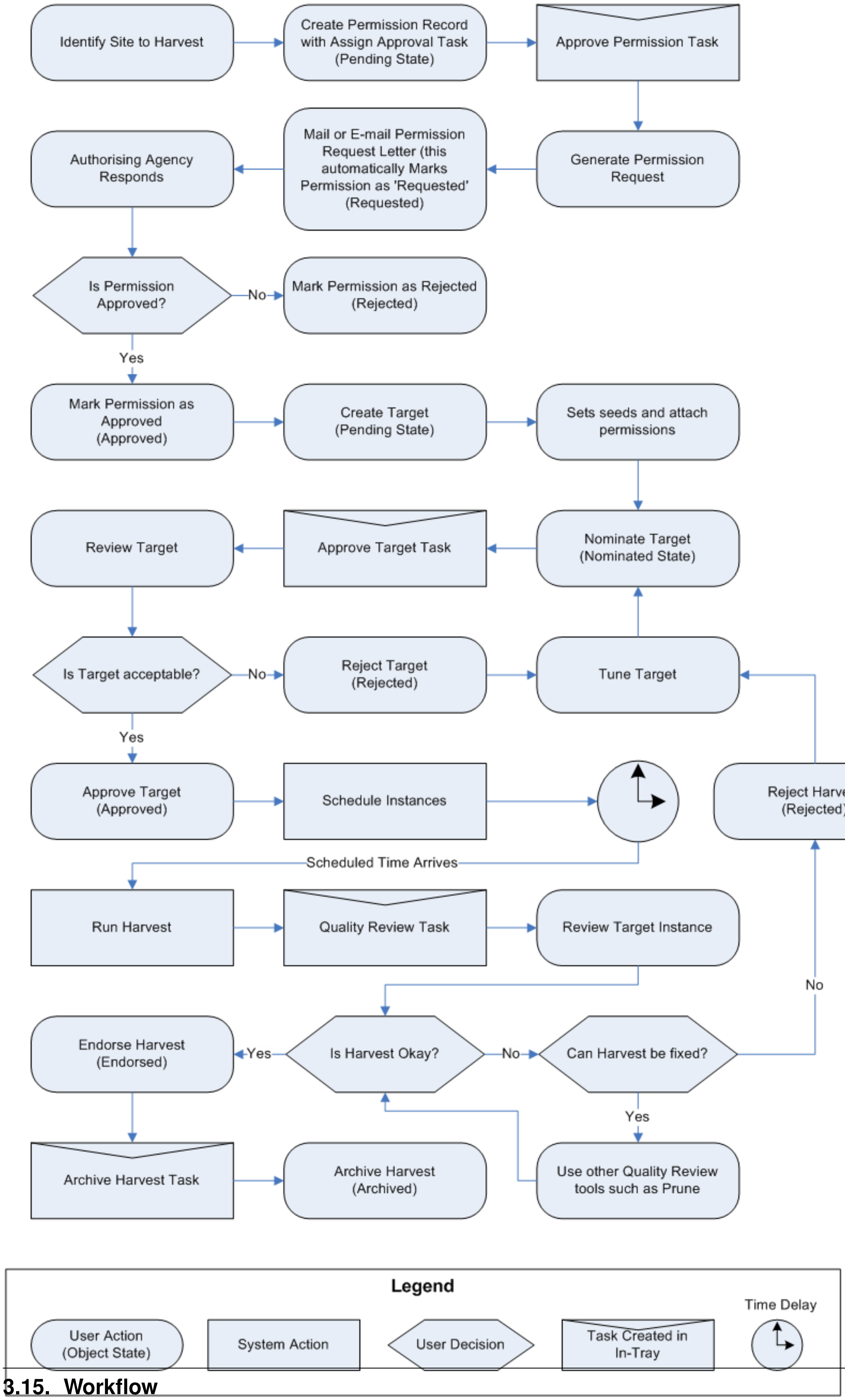

Identify Site to Harvest
Create Permission Record with Assign Approval Task (Pending State)
Approve Permission Task
Authorising Agency Responds
Mail or E-mail Permission Request Letter (this automatically Marks Permission as 'Requested' (Requested)
Generate Permission Request
Is Permission Approved?
No
Mark Permission as Rejected (Rejected)
Yes
Mark Permission as Approved (Approved)
Create Target (Pending State)
Sets seeds and attach permissions
Review Target
Approve Target Task
Nominate Target (Nominated State)
Is Target acceptable?
No
Reject Target (Rejected)
Tune Target
Yes
Approve Target (Approved)
Schedule Instances
Reject Harvest (Rejected)
Scheduled Time Arrives
Run Harvest
Quality Review Task
Review Target Instance
No
Endorse Harvest (Endorsed)
Yes
Is Harvest Okay?
No
Can Harvest be fixed?
Yes
Archive Harvest Task
Archive Harvest (Archived)
Use other Quality Review tools such as Prune
Legend
User Action (Object State)
System Action
User Decision
Task Created in In-Tray
Time Delay

Figure 28: Detailed workflow

CHAPTER 4

Quick Start Guide

4.1 Introduction

4.1.1 About the Web Curator Tool

The Web Curator Tool is a tool for managing the selective web harvesting process. It is typically used at national libraries and other collecting institutions to preserve online documentary heritage.

Unlike previous tools, it is enterprise-class software, and is designed for non-technical users like librarians. The software was developed jointly by the National Library of New Zealand and the British Library, and has been released as free software for the benefit of the international collecting community.

4.1.2 About this document

This document describes how to set up the Web Curator Tool on a Linux system in the simplest possible way. Its intended audience are users who want to quickly install and try out the software.

For a proper production set up, see the *System Administrator Guide*

4.2 Installation

4.2.1 Prerequisites

- Apache Tomcat 8.5 or newer
- MySQL 5.0.95 or newer. PostgreSQL and Oracle are also supported, but in this Quick Start Guide we'll be using MySQL/MariaDB
- Heritrix 3.3.0 or newer
- Java 8 or higher

4.2.2 Setting up Heritrix 3

We're assuming that Tomcat and MySQL have already been set up. For Heritrix 3.3.0, we'll be using a recent stable build of the 3.3.0 branch. The Heritrix 3 Github wiki contains a section detailing the current master builds available https://github.com/internetarchive/heritrix3/wiki#master-builds.

Unzip the archive containing the Heritrix binary, go into the resulting directory and execute the following:

```
user@host:/usr/local/heritrix-3.3.0-SNAPSHOT$ cd bin
user@host:/usr/local/heritrix-3.3.0-SNAPSHOT/bin$
user@host:/usr/local/heritrix-3.3.0-SNAPSHOT/bin$ ./heritrix -a admin
```

This starts up Heritrix with the password "admin" for the user admin, which is the default set of credentials used by the WCT harvest agent. You can also specify the Heritrix jobs directory using the `-j` parameter. Otherwise the default will be used **<HERITRIX_HOME>/jobs**.

4.2.3 Creating the database

Download the latest stable binary WCT release from https://github.com/DIA-NZ/webcurator/releases. Extract the archive and go into the resulting directory (in our case **/tmp/wct**). Then, to create the WCT database and its objects, run the script set-up-mysql.sh (found in the db subdirectory):

```
user@host:/tmp/wct$ cd db
user@host:/tmp/wct/db$ ./set-up-mysql.sh
```

You'll need to set the variable `$MYSQL_PWD` *in this script to the correct value for your MySQL installation.*

4.2.4 Deploying and configuring the WCT components

To deploy the WCT components into Tomcat, copy the war files from the WCT directory to the Tomcat webapps directory.

```
user@host:/tmp/wct$ cd war
user@host:/tmp/wct/war$ cp * /usr/local/apache-tomcat-9.0.12/webapps
```

If Tomcat is already running with auto deploy configured `autoDeploy="true"` *, then the war files should now be extracted into their own directories. If not, start/restart Tomcat.*

Then shutdown Tomcat so you can edit the config files in the newly created directories.

Note that, by default, WCT assumes the existence of a directory ***/usr/local/wct*** *, where it stores all its files. If you make sure that this directory exists, there should be no need to edit any config files.*

After the war files have been extracted, we first need to check whether the database connection settings are appropriate for our situation. These settings can be found in the file **webapps/wct/META-INF/context.xml**. If you have a typical MySQL setup, you shouldn't have to change anything here.

Next, we'll make sure the WCT store component uses the correct directory for storage, by setting the variable `arcDigitalAssetStoreService.baseDir` in **webapps/wct-store/WEB-INF/classes/wct-das.properties** to the appropriate value. Make sure the device on which this directory is located has enough space to store your harvests. By default, it uses **/usr/local/wct/store**.

Finally, we need to make sure the temporary directory used by the H3 Harvest Agent is suitable for our situation by setting the variable `harvestAgent.baseHarvestDirectory` in **webapps/harvest-agent-h3/WEB-INF/classes/wct-agent.properties** to the appropriate value. The default is **/usr/local/wct/harvest-agent**.

Note, the `harvestAgent.baseHarvestDirectory` *path* **cannot** *match the Heritrix 3 jobs directory. This will cause a conflict within the H3 Harvest Agent.*

You can now start Tomcat, after which you should be able to login at http://localhost:8080/wct, using the user 'bootstrap' and password 'password'. You can now create users and roles and configure the system. Refer to the User Manual for more information.

4.2.5 Caveats

This document only covers the most simple scenario for setting up WCT and will probably not result in a system that meets the production requirements of your organisation. Important topics that have not been covered here:

- WCT can also authenticate users via LDAP (see the *System Administrator Guide*)
- By default all communication between the components and between the browser and WCT is unencrypted. To enable SSL/TLS, please follow the instructions for your version of Tomcat
- You can use OpenWayback to view harvests from within WCT, see the wiki on the WCT Github page: https://github.com/DIA-NZ/webcurator/wiki/Wayback-Integration

CHAPTER 5

System Administrator Guide

5.1 Introduction

This guide, designed for a System Administrator, covers installation and setup of the Web Curator Tool. An electronic copy can be downloaded from the WCT Github site: http://dia-nz.github.io/webcurator/

For information on using the Web Curator Tool, see the Web Curator Tool Quick Start Guide and the Web Curator Tool online help.

5.1.1 Contents of this document

Following this introduction, the Web Curator Tool System Administrator Guide includes the following sections:

- **Getting Started** - covers prerequisites, supported platforms, other platforms, and optional prerequisites for using the Web Curator Tool.
- **Setting up the WCT database** - procedures for setup using Oracle, MySQL and PostgreSQL.
- **JMX setup** - procedures for setting up JMX for different WCT components.
- **Setting up the WCT Application Servers** - procedures for deploying WCT to Tomcat, includes configuration options and troubleshooting.
- **Setting up Heritrix 3** - procedures for building and running the Heritrix 3 web crawler to intergrate with WCT, includes configuration options and troubleshooting.
- **Appendix A: Creating a truststore and importing a certificate**
- **Appendix B: The OMS archive adapter**

5.2 Getting Started

The following section explains how to get the Web Curator Tool up and running.

5.2.1 Prerequisites

The following are required to successfully install and run the Web Curator Tool:

- Java 1.8 JDK or above (64bit recommended)

  During development of the latest version it was noted that large harvests would sometimes fail to transfer from the Harvest Agent to Store on completion. This was resolved by running Apache Tomcat with 64 bit Java.

- Apache Tomcat 8.x.x or above (the application has been tested on Tomcat 8.5.32)
- A database server (select one of the databases below)
  - Oracle 11g or newer
  - PostgreSQL 8.4.9 or newer
  - MySQL 5.0.95 or newer

Other versions of the required products may be compatible with the Web Curator Tool but they have not been tested. Due to the products use of Hibernate for database persistence other database platforms should work, if the product is rebuilt with the correct database dialect. However only Postgesql, Oracle 11g, and MySQL have been tested.

5.2.2 Supported platforms

The following platforms have been used during the development of the Web Curator Tool:

- Sun Solaris 10
- Red Hat Linux EL3.
- Ubuntu GNU/Linux 16.04 LTS
- Windows 7 Ultimate

5.2.3 Other platforms

The following platforms were used during the Development of the Web Curator tool but are not explicitly supported:

- Windows 2000, Windows XP Pro, Windows Server 2003

5.2.4 Optional prerequisites

The following prerequisites are optional:

- LDAP compliant directory (for external authentication)
- Apache Maven 3+ (required to build from source).
- Git (can be used to clone the project source from Github)

5.3 Setting up the WCT database

Currently the WCT has been tested with Oracle 11g, MySQL 5.0.95, MariaDB 10.0.36 and PostgreSQL 8.4.9 and 9.6.11.

5.3.1 Setup using Oracle

This guide assumes you have installed and configured Oracle 11g prior to setting up the WCT database and schema.

1. Setup two schemas: one called DB_WCT that owns the tables and one called USR_WCT that the application uses to query the tables. The USR_WCT schema should have limited rights. You can use the following SQL script to do this:

   ```
   db/latest/setup/wct-create-oracle.sql
   ```

2. Run the following SQL scripts under the DB_WCT user or SYSTEM account:

   ```
   db/latest/setup/wct-schema-oracle.sql

   db/latest/setup/wct-schema-grants.sql

   db/latest/setup/wct-indexes-oracle.sql

   db/latest/setup/wct-bootstrap-oracle.sql

   db/latest/setup/wct-qa-data-oracle.sql
   ```

The wct-qa-data-oracle.sql script will generate QA indicator template data for the new QA module for each agency, and should be run **once all agencies have been added to WCT**. *Note that if the script is re-run, it will clear out any existing template data.*

3. Locate the correct JDBC driver for Oracle, which should be distributed with the Oracle install media.

   - The JDBC driver should be called ojdbc1411g.jar
   - The driver will need to be placed into the $TOMCAT_HOME/common/lib/ directory.
   - Also required in this directory is the jta.jar

Notes: A password strategy should be defined for the system, and the db_wct & usr_wct passwords should be changed in the scripts and application property files to conform to this strategy. To encourage this, the passwords in the supplied database creation script are set to 'password'.

The bootstrap user script creates a User with a name of 'bootstrap' and a password of 'password'. Use this account to login to the application once it is up and running. You can use the bootstrap account to create other users and agencies. Once you have setup valid users, it is best to disable the bootstrap user for security reasons.

5.3.2 Setup using PostgreSQL

This guide assumes you have installed and configured PostgreSQL prior to setting up the WCT database and schema.

1. Setup two schema, using the following script:

   ```
   db/latest/setup/wct-create-postgres.sql
   ```

2. Then run the following SQL scripts under the DB_WCT user:

   ```
   db/latest/setup/wct-schema-postgresql.sql

   db/latest/setup/wct-schema-grants-postgresql.sql

   db/latest/setup/wct-indexes-postgresql.sql
   ```

(continues on next page)

(continued from previous page)

```
db/latest/setup/wct-bootstrap-postgresql.sql

db/latest/setup/wct-qa-data-postgres.sql
```

The wct-qa-data-postgres.sql script will generate QA indicator template data for the new QA module for each agency, and should be run **once all agencies have been added to WCT**. *Note that if the script is re-run, it will clear out any existing template data.*

3. The Postgres JDBC driver is included in the Github repository under /etc/ directory.

    - The Postgres driver is called postgresql-8.1-404.jdbc3.jar
    - The driver will need to be placed into the $TOMCAT_HOME/common/lib/ directory.
    - Also required in the $TOMCAT_HOME/common/lib/ directory is the jta.jar

Notes: A password strategy should be defined for the system, and the usr_wct password should be changed in the scripts and application property files to conform to this strategy. To encourage this, the password in the supplied database creation script is set to 'password'.

The bootstrap user script creates a User with a name of 'bootstrap' and a password of 'password'. Use this account to login to the application once it is up and running. You can use the bootstrap account to create other users and agencies. Once you have setup valid users, it is best to disable the bootstrap user for security reasons.

5.3.3 Setup using MySQL

This guide assumes you have installed and configured MySQL prior to setting up the WCT database and schema.

1. Create the database, using the following script:

```
db/latest/setup/wct-create-mysql.sql
```

2. Then run the following SQL scripts under the root user:

```
db/latest/setup/wct-schema-mysql.sql

db/latest/setup/wct-schema-grants-mysql.sql

db/latest/setup/wct-indexes-mysql.sql

db/latest/setup/wct-bootstrap-mysql.sql

db/latest/setup/wct-qa-data-mysql.sql
```

The wct-qa-data-mysql.sql script will generate QA indicator template data for the new QA module for each agency, and should be run **once all agencies have been added to WCT**. *Note that if the script is re-run, it will clear out any existing template data.*

3. Download the MySQL JDBC driver from the MySQL website.

    - The driver will need to be placed into the $TOMCAT_HOME/common/lib/ directory.
    - Also required in the $TOMCAT_HOME/common/lib/ directory is the jta.jar

Notes: A password strategy should be defined for the system, and the usr_wct password should be changed in the scripts and application property files to conform to this strategy. To encourage this, the password in the supplied database creation script is set to 'password'.

The bootstrap user script creates a User with a name of 'bootstrap' and a password of 'password'. Use this account to login to the application once it is up and running. You can use the bootstrap account to create other users and agencies. Once you have setup valid users, it is best to disable the bootstrap user for security reasons.

5.4 JMX setup

WCT core and every Harvest Agent require JMX Remote access. This means that JMX Remote control and access files will need to be setup for the JVM. This is done with the following steps:

1. Create a *jmxremote.password* file by copying the file *jmxremote.password.template* to the jmx remote password file that your installation will use. This template file will be in your JDK's *jrelibmanagement* directory.

   You can use the property '-Dcom.sun.management.jmxremote.password.file=<property-file>' to point to a different location.

   The monitor role and control role have passwords associated with them. These are setting withing hte jmx remote password file:

   ```
   monitorRole  apassword
   controlRole  apassword
   ```

2. It is important that this file is protected. If using Windows, refer to the following link to protect the file using the O/S: http://java.sun.com/j2se/1.5.0/docs/guide/management/security-windows.html

   If using *nix platform, protect the file using:

   ```
   chmod 600 jmxremote.password.
   ```

3. Enable the JMX Remote port used in the JVM's startup. Any high port can be used as long as it is unique on the machine that is running the component. The example here uses port *9004*, but if multiple components are running on the same machine, then each component will need a different and unique port number.

   For Tomcat, this is done by adding the following to your *$TOMCAT_HOME/bin/catalina.sh script*:

   ```
   JAVA_OPTS=-Dcom.sun.management.jmxremote.port=9004
   ```

   For a Harvest Agent, the Harvest Agent would need to include the *-Dcom.sun.management.jmxremote.port=9004* as part of the Java command line or by including it in the Java environment variable *JAVA_OPTS*.

   IMPORTANT: *Make sure your JMX port is unique. Different components of WCT will be running JMX so they will need to be configured to use different ports.*

5.5 Setting up the WCT Application Servers

5.5.1 Deploying WCT to Tomcat

There are three major components to the deployment of the Web Curator Tool:

- the web curator core (wct.war)
- the web curator harvest agent (wct-harvest-agent.war)
- the web curator digital asset store (wct-store.war).

Each of these three components must be deployed for the Web Curator Tool to be fully functional and more than one harvest agent can be deployed if necessary. Each Harvest Agent is capable of carrying out harvest actions. The more harvest agents deployed the more harvesting that can be done at any one point in time. The harvest agents and digital asset store can reside on any machine within the network, as they use SOAP over HTTP to communicate with each other.

To deploy WCT to Tomcat:

- Make sure you have installed and configured both Java 1.8 JDK and Apache-Tomcat 8.x.x successfully.
- Set up the JMX Remote control and access files for the WCT core as described in the section *JMX setup*.
- Deploy the WAR files into Tomcat. The simplest deployment is to deploy all three WAR files into the same Tomcat container.
  - You can copy the WAR files into the $TOMCAT_HOME/webapps/ directory.
  - Provided Tomcat is configured correctly, when you start Tomcat the WAR files will be exploded and the application will start.
- Shut down Tomcat once the WAR files have been extracted. This will allow you to modify the configuration files in the following steps.

Configure the Database Connection

The open source version of the Web Curator Tool is configured to use a local PostgreSQL database. If you are using any other database, or are using a database server, you will need to change the database configuration.

- Set the correct database dialect in TOMCAT/webapps/wct/WEB-INF/classes/**wct-core.properties**:

```
#Hibernate Settings

hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
hibernate.default_schema=DB_WCT
```

  The appropriate dialects are shown in the table below.

Database	Dialect
Oracle	org.hibernate.dialect.OracleDialect
PostgreSQL	org.hibernate.dialect.PostgreSQLDialect
MySQL	org.hibernate.dialect.MySQLDialect

- Edit the context.xml file in TOMCAT/webapps/wct/META-INF:

```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
    <Resource
        name="jdbc/wctDatasource"
        type="javax.sql.DataSource"
        password="**PASSWORD**"
        driverClassName="**DRIVER**"
        maxIdle="2"
        maxWait="5000"
        validationQuery="**VALIDATION_QUERY**"
        username="**USERNAME**"
        url="**JDBC_URL**"
        maxActive="10 "/>
</Context>
```

Set the username and password properties as appropriate for your database. If you have followed the defaults, then these should remain as USR_WCT/USR_WCT.

The remaining properties should be set as follows:

Oracle

Attribute	Value
DRIVER	oracle.jdbc.driver.OracleDriver
VALIDATION_QUERY	select count(1) from DUAL
JDBC_URL	jdbc:oracle:thin:@servername:port/SID

PostgreSQL

Attribute	Value
DRIVER	org.postgresql.Driver
VALIDATION_QUERY	select 1+1
JDBC_URL	jdbc:postgresql://servername:port/database

MySQL

Attribute	Value
DRIVER	com.mysql.jdbc.Driver
VALIDATION_QUERY	select 1+1
JDBC_URL	jdbc:mysql://servername:port/database

- Copy the context.xml file to the TOMCAT/conf/Catalina/localhost directory. Delete the existing wct.xml file if it exists. Now rename the context.xml file to wct.xml.

Configure LDAP Authentication (Unencrypted)

- If you wish to use an external Directory for Authentication, then WCT should be configured to allow this. Unencrypted authentication can be done very simply with your directory by modifying the wct-core-security.xml and the wct-core.properties file.

  The Directory must support LDAP.

  In wct-core-security.xml, uncomment the ldapAuthenticator bean:

```
<bean id="authenticationManager"
class="org.acegisecurity.providers.ProviderManager" abstract="false"
singleton="true" lazy-init="default" autowire="default"
dependency-check="default">
    <property name="providers">
        <list>
            <ref bean="ldapAuthenticator" />
            <ref bean="daoAuthenticationProvider" />
        </list>
    </property>
</bean>
```

  In wct-core.properties, set the following parameters:

```
#LDAP Settings
ldap.url=ldap://yourldaphost.domain.com:389
ldap.dn=cn={0},OU=OrgUnit,O=Organisation
```

The two parameters of interest are:

  - ldap.url, which defines the URL for the directory. This is normally something like ldap://mydirectory.natlib.co.nz/
  - ldap.dn. This allows the Directory DN to be defined. For example, if a user logs in with the username “gordonp” the Directory will be queried using the distinguished name of “cn=gordonp, ou=wct, o=global”. So the user must exist within the global organisation and the wct organisation unit.

Configure LDAP Authentication (Encrypted using TLS or SSL)

- If you want all credentials passed to the Directory server to be protected then the ldap traffic should be encrypted using TLS or SSL.
  - The only difference to the wct-core.properties file from step 4 is the following change:

    ```
    ldap.url=ldaps://yourldaphost.domain.com:389
    ```

  - If using TLS or SSL then you must configure Tomcat to allow secure communication with your Directory by adding the following to your $TOMCAT_HOME/bin/catalina.sh script:

    ```
    JAVA_OPTS= -Djavax.net.ssl.trustStore=/var/wctcore/ssl/wct.ts
    -Djavax.net.ssl.trustStorePassword=password
    ```

    This points tomcat to a Truststore that contains the public key for you directory. If your directory utilises a correctly signed certificate, you may not need this, as the default truststore provided by Java contains all the major root certificates. However if you directory uses a self-signed certificate then you will need to export the public key of that certificate and import it into your truststore (i.e. /var/wctcore/ssl/wct.ts). Alternatively you can import the self-signed certificate into the default Java truststore.

    For details on how to create a truststore and import a certificate, see Appendix A: Creating a truststore and importing a certificate.

Configure the Digital Asset Store

- Set the Base Directory of the Digital Asset Store to a valid location on the server. Also make sure the directory or share has enough free disk space.

  The configuration for the DAS is found in the **wct-das.properties** file:

```
#WctCoreWsEndpoint

wctCoreWsEndpoint.service=/wct/services/urn:WebCuratorTool
wctCoreWsEndpoint.host=localhost
wctCoreWsEndpoint.port=8080

#ArcDigitalAssetStoreService

# the base directory for the arc store
arcDigitalAssetStoreService.baseDir=/tmp/arcstore
```

Configure a Heritrix 3 - Harvest Agent

- Make sure the following parameters are correct for your environment in the **wct-agent.properties** file:

```
#HarvestAgent

# name of the directory where the temporary harvest data is stored
harvestAgent.baseHarvestDirectory=/wct/harvest-agent
# agent host name or ip address that the core knows about
harvestAgent.host=localhost
# the port the agent is listening on for http connections
harvestAgent.port=8080
# the name of the harvest agent web service
harvestAgent.service=/harvest-agent-h3/services/urn:HarvestAgent
# the name of the harvest agent log reader web service
harvestAgent.logReaderService=/harvest-agent-h3/services/urn:LogReader
# the max number of harvest to be run concurrently on this agent
harvestAgent.maxHarvests=2
# the name of the agent. must be unique
harvestAgent.name=My local H3 Agent
# the note to send with the harvest result.
harvestAgent.provenanceNote=Original Harvest
# the number of alerts that occur before a notification is sent
harvestAgent.alertThreshold=200
# whether to attempt to recover running harvests from H3 instance on startup.
harvestAgent.attemptHarvestRecovery=true


#HarvestCoordinatorNotifier

# the name of the core harvest agent listener web service
harvestCoordinatorNotifier.service=/wct/services/urn:WebCuratorTool
# the host name or ip address of the core
harvestCoordinatorNotifier.host=localhost
# the port that the core is listening on for http connections
harvestCoordinatorNotifier.port=8080


#DigitalAssetStore

# the name of the digital asset store web service
digitalAssetStore.service=/wct-store/services/urn:DigitalAssetStore
# the host name or ip address of the digital asset store
digitalAssetStore.host=localhost
# the port that the digital asset store is listening on for http connections
digitalAssetStore.port=8080

...

#Triggers

# startDelay: delay before running the job measured in milliseconds
# repeatInterval: repeat every xx milliseconds (Note that once a day is
86,400,000 millseconds)

heartbeatTrigger.startDelay=10000
heartbeatTrigger.repeatInterval=30000
```

- In addition to setting the Harvest Agent parameters, you may also want to change the default Heritrix v3 profile that is shipped with the WCT. See the *Default profile* section.

Configure a Heritrix 1 - Harvest Agent

- Make sure the following parameters are correct for your environment in the **wct-agent.properties** file:

```
#HarvestAgent

# name of the directory where the temporary harvest data is stored
harvestAgent.baseHarvestDirectory=/wct/harvest-agent
# agent host name or ip address that the core knows about
harvestAgent.host=localhost
# the port the agent is listening on for http connections
harvestAgent.port=8080
# the name of the harvest agent web service
harvestAgent.service=/harvest-agent-h1/services/urn:HarvestAgent
# the name of the harvest agent log reader web service
harvestAgent.logReaderService=/harvest-agent-h1/services/urn:LogReader
# the max number of harvest to be run concurrently on this agent
harvestAgent.maxHarvests=2
# the name of the agent. must be unique
harvestAgent.name=My local H1 Agent
# the note to send with the harvest result.
harvestAgent.provenanceNote=Original Harvest
# the number of alerts that occur before a notification is sent
harvestAgent.alertThreshold=200


#HarvestCoordinatorNotifier

# the name of the core harvest agent listener web service
harvestCoordinatorNotifier.service=/wct/services/urn:WebCuratorTool
# the host name or ip address of the core
harvestCoordinatorNotifier.host=localhost
# the port that the core is listening on for http connections
harvestCoordinatorNotifier.port=8080


#DigitalAssetStore

# the name of the digital asset store web service
digitalAssetStore.service=/wct-store/services/urn:DigitalAssetStore
# the host name or ip address of the digital asset store
digitalAssetStore.host=localhost
# the port that the digital asset store is listening on for http connections
digitalAssetStore.port=8080

...

#Triggers

# startDelay: delay before running the job measured in milliseconds
# repeatInterval: repeat every xx milliseconds (Note that once a day is
86,400,000 millseconds)

heartbeatTrigger.startDelay=20000
heartbeatTrigger.repeatInterval=30000
```

- In addition to setting the Harvest Agent parameters, you may also want to change the default Heritrix v1.14 profile that is shipped with the WCT. The most likely settings to change are what web proxy server to use when harvesting content. The setting can be found in the **WEB-INF/classes/default-profile.xml**:

  ```
  <newObject name="HTTP" class="org.archive.crawler.fetcher.FetchHTTP">
      <boolean name="enabled">true</boolean>
      <map name="filters">
      </map>
      <map name="midfetch-filters">
      </map>
      <integer name="timeout-seconds">1200</integer>
      <integer name="sotimeout-ms">20000</integer>
      <long name="max-length-bytes">0</long>
      <boolean name="ignore-cookies">false</boolean>
      <boolean name="use-bdb-for-cookies">true</boolean>
      <string name="load-cookies-from-file"></string>
      <string name="save-cookies-to-file"></string>
      <string name="trust-level">open</string>
      <stringList name="accept-headers">
      </stringList>
      <string name="http-proxy-host"></string>
      <string name="http-proxy-port"></string>
      <string name="default-encoding">ISO-8859-1</string>
      <boolean name="sha1-content">true</boolean>
      <boolean name="send-connection-close">true</boolean>
      <boolean name="send-referer">true</boolean>
      <boolean name="send-range">false</boolean>
  </newObject>
  ```

  - If you don’t have a web proxy then just leave the values blank.

    Heritrix v1.14 does not currently support authenticated proxy access, so the proxy server must allow unauthenticated access.

Set the Attachments Directories

- Set the attachments directories in the server-config.wsdd files for all three components. This file is found in the WEB-INF directory of each application. This directory must exist and be accessible by the Tomcat server.

```
<parameter name="attachments.Directory" value="/tmp/attach"/>
```

Logon to WCT

Once you have started up the Web Curator Tool logon to the application using the ‘bootstrap’ user with the default password of ‘password’. This account has enough privilege to create other Agencies and Users within the system. Once you have configured valid WCT users and tested their login’s work, you should disable the bootstrap user.

The URL to access WCT running on Apache/Tomcat will be similar to the one displayed below:

http://localhost/wct/ where ‘localhost’ can be replaced with your server name. Note, if using tomcat only, the default port for tomcat is 8080, changing the URL to http://localhost:8080/wct/ will allow you to connect directly to Tomcat.

Heritrix v1 Harvest Agent use only The other common trap is not defining the default bandwidth for the system. On start-up of WCT the system bandwidth is set to 0 KB’s for every day of the week. Before Harvests can be initiated you must specify a base bandwidth for each of the days you plan to harvest on.

In order to setup the bandwidth you must logon as a user that has the ‘Manage Web Harvester System’ privilege set (usually an WCT Administrator). The Bandwidth screen can be found under the ‘Management -> Harvester Configuration -> Bandwidth’ section of the site.

5.5.2 Troubleshooting setup

See the following table to troubleshoot Web Curator Tool setup.

Problem	Possible solution
Database connection failure	Check that the WCT core data source is defined correctly in the wct/META-INF/context.xml and that the server can communicate with this host on the specified port.
LDAP configuration failure	If problems occur with getting TLS working with ldap, then switch on the SSL debug mode within Tomcat by adding the following to the JAVA_OPTS environment variable. The debug will display on the console. -Djavax.net.debug=ssl
JMX remote register failure	Tomcat will not start if the permissions are incorrect on the jmxremote.password file. Check that the jmxremote.password file exists and has the correct ownership.
Communication failure on Heartbeat	Validate that the distributed agents have the correctly defined central host and can communicate with this host over HTTP.
Failure on storing the harvest to the store	Validate that the Digital Asset Store has been configured with the correct directory settings and has write access to the specified directory.
Failure on Harvest attempt (or Harvest action appears to hang)	2006-07-04 07:51:31,640 ERROR [http-8080-Processor24] agent.HarvestAgentHeritrix (HarvestAgentHeritrix.java:88) - Failed to initiate harvest for 262147 : Failed to create the job profile C:tmpharvest-agent262147ord er.xml. org.webcurator.core.harvester.a gent.exception.HarvestAgentExcept ion: Failed to create the job profile **C:tmpharvest-agent262147o rder.xml.** at org.webcurator.core.harvester.a gent.HarvestAgentHeritrix.createP rofile(HarvestAgentHeritrix.java: 542) at org.webcurator.core.harvester.a gent.HarvestAgentHeritrix.initiat eHarvest(HarvestAgentHeritrix.jav a:79) at org.webcurator.core.harvester.a gent.HarvestAgentSOAPService.init iateHarvest(HarvestAgentSOAPServi ce.java:37) If any error similar to the one above occurs, it is usually related to an incomplete harvest taking place. If this occurs you will need to remove the Target Instance sub-directory from the deployed baseHarvestDirectory as specified in the wct-agent.xml. In the example above you would delete the directory called c:tmpharvest-agent262147
QA Process does not appear to run or QA indicators are	Check that QA indicators have been defined in the Management tab of WCT. The \sqlwct-qa-data-1_6-[mysql/orac le/postgres].sql scripts have been provided to generate initial values for the QA indicators.

5.5.3 Configuration options

This section describes options for configuring the Web Curator Tool.

Web Curator Core - context.xml

The /META-INF/context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
    <Resource
        name="jdbc/wctDatasource"
        type="javax.sql.DataSource"
        password="${schema.password}"
        driverClassName="${schema.driver}"
        maxIdle="${schema.maxIdle}"
        maxWait="5000"
        validationQuery="${schema.query}"
        username="${schema.user}"
        url="${schema.url}"
        maxActive="${schema.maxActive}"
    />
</Context>
```

This file defines the data source to use for the WCT and specifies the JDBC driver class, database URL, username, password, max and min connections and the keep alive query. The parameters surrounded by ${ } characters are replaced when this file is built using maven, with the appropriate values from the build.properties at build time, or wct-core.properties files at run time.

Web Curator Core - wct-core.xml

The /WEB-INF/classes/wct-core.xml

```
<bean id="schedulePatternFactory"
class="org.webcurator.domain.SpringSchedulePatternFactory">
    <property name="patterns">
        <list>
        <bean class="org.webcurator.domain.model.core.SchedulePattern">
            <property name="scheduleType" value="1"/>
            <property name="description" value="Every Monday at 9:00pm"/>
            <property name="cronPattern" value="00 00 21 ? * MON *"/>
        </bean>
        </list>
    </property>
</bean>
```

The **schedulePatternFactory** defines all the default CRON patterns used by the WCT to schedule Targets for harvest. For each additional SchedulePattern required an additional SchedulePattern bean should be added to the list.

```
<bean id="politePolitenessOptions" class="org.webcurator.core.profiles.
↪PolitenessOptions"
abstract="false" singleton="true" lazy-init="default" autowire="default" dependency-
↪check="default">
    <!-- Delay Factor -->
    <constructor-arg index = "0" type = "double" value = "10.0"/>
```

(continues on next page)

(continued from previous page)

```
    <!-- Min Delay milliseconds -->
    <constructor-arg index = "1" type = "long" value = "9000"/>
    <!-- Max Delay milliseconds -->
    <constructor-arg index = "2" type = "long" value = "90000"/>
    <!-- Respect crawl delay up to seconds -->
    <constructor-arg index = "3" type = "long" value = "180"/>
    <!-- Max per host bandwidth usage kb/sec -->
    <constructor-arg index = "4" type = "long" value = "400"/>
</bean>

<bean id="mediumPolitenessOptions" class="org.webcurator.core.profiles.
↪PolitenessOptions"
abstract="false" singleton="true" lazy-init="default" autowire="default" dependency-
↪check="default">
    <!-- Delay Factor -->
    <constructor-arg index = "0" type = "double" value = "5.0"/>
    <!-- Min Delay milliseconds -->
    <constructor-arg index = "1" type = "long" value = "3000"/>
    <!-- Max Delay milliseconds -->
    <constructor-arg index = "2" type = "long" value = "30000"/>
    <!-- Respect crawl delay up to seconds -->
    <constructor-arg index = "3" type = "long" value = "30"/>
    <!-- Max per host bandwidth usage kb/sec -->
    <constructor-arg index = "4" type = "long" value = "800"/>
</bean>

<bean id="aggressivePolitenessOptions" class="org.webcurator.core.profiles.
↪PolitenessOptions"
abstract="false" singleton="true" lazy-init="default" autowire="default" dependency-
↪check="default">
    <!-- Delay Factor -->
    <constructor-arg index = "0" type = "double" value = "1.0"/>
    <!-- Min Delay milliseconds -->
    <constructor-arg index = "1" type = "long" value = "1000"/>
    <!-- Max Delay milliseconds -->
    <constructor-arg index = "2" type = "long" value = "10000"/>
    <!-- Respect crawl delay up to seconds -->
    <constructor-arg index = "3" type = "long" value = "2"/>
    <!-- Max per host bandwidth usage kb/sec -->
    <constructor-arg index = "4" type = "long" value = "2000"/>
</bean>
```

The **PolitenessOptions** define the Heritrix 3 politeness settings. These values are shown in the UI when editing a Heritrix 3 profile, and are used to adjust whether a crawl will be performed in an aggressive, moderate or polite manner.

Web Curator Core - wct-core.properties

The /WEB-INF/classes/wct-core.properties

```
# name of the directory where the h3 scripts are stored
h3.scriptsDirectory=/tmp/h3scripts
```

See *Scripts directory* under *Setting up Heritrix 3*.

```
#HarvestCoordinator settings

harvestCoordinator.minimumBandwidth=10
harvestCoordinator.maxBandwidthPercent=80
harvestCoordinator.daysBeforeDASPurge=14
harvestCoordinator.daysBeforeAbortedTargetInstancePurge=7
```

The **harvestCoordinator** is responsible for the coordination of harvest activity across all of the Harvest Agents. This is where the minimum bandwidth (in KB/s) and maximum bandwidth percentages are defined for all agents. Also defined in the Co-ordinator is the number of days before the Digital Asset Store is purged as well as the number of days before data remaining after aborted harvests is purged.

```
harvestCoordinator.harvestOptimizationEnabled=true
harvestCoordinator.harvestOptimizationLookaheadHours=12
harvestCoordinator.numHarvestersExcludedFromOptimisation=1
```

The harvest coordinator is able to "optimize" harvests that are configured to be optimizable. Optimizable harvests will begin earlier than their scheduled time, when the harvests can support the extra harvest, and when the scheduled time is within the look-ahead window configuration. A number of harvesters can also be excluded from optimization, to allow for non-optimizable harvests to execute on schedule.

Targets can be configured as optimizable on the target edit screen.

Note also that there is also the ability to prevent harvest optimization during certain hours, based on the bandwidth settings, in the Management->Bandwidth area.

```
processScheduleTrigger.startDelay=10000
processScheduleTrigger.repeatInterval=30000
```

The **processScheduleTrigger** defines when the heartbeat activity is checked on the registered Agents. The time is measured in milliseconds.

```
#MailServer settings

mailServer.smtp.host=yourhost@yourdomain.co.uk
mail.smtp.port=25
```

The **mailServer** bean is responsible for communicating with an SMTP mail server for sending email notifications.

```
#InTrayManager settings

inTrayManager.sender=noreply@yourdomain.com
inTrayManager.wctBaseUrl=http://localhost:8080/wct/
```

The **inTrayManager** is responsible for informing users of Tasks or Notification messages. This uses the mailServer to send email. Also defined here is the sender of the automated system Tasks and Notifications.

```
#GroupSearchController settings

groupSearchController.defaultSearchOnAgencyOnly=true
```

The **groupSearchController** defines how the default search is handled on the Groups tab. When **defaultSearchOnAgencyOnly** is set to *true*, the user name is omitted from the default Group search filter allowing the display of all groups for the current user's agency. When **defaultSearchOnAgencyOnly** is set to *false*, the user name is included in the filter and only those Groups owned by the current user are displayed.

```
#ArchiveAdapter settings

archiveAdapter.targetReferenceMandatory=false
```

The **archiveAdapter** The archive adapter provides the mechanism for archiving a harvested target instance into an archive repository. When **targetReferenceMandatory** is set to *true (or is omitted)*, the owning Target for a Target Instance being archived must have a Target Reference defined in order for archiving to be attempted. When **targetReferenceMandatory** is set to *false*, there is no need for the owning Target to have a Target Reference defined.

```
#QualityReviewToolController settings

qualityReviewToolController.enableBrowseTool=true
qualityReviewToolController.enableAccessTool=false
qualityReviewToolController.archiveUrl=http://web.archive.org/web/*/
qualityReviewToolController.archiveName=Wayback
qualityReviewToolController.archive.alternative=http://web.archive.org/web/*/
qualityReviewToolController.archive.alternative.name=Another Wayback

#HarvestResourceUrlMapper settings

#Used to rewrite urls to use an external Quality Review Tool. Note that for use
#with Wayback, the Wayback indexer should be enabled in wct-das.properties
#Available substitution values:

# {$HarvestResult.Oid}
# {$HarvestResult.HarvestNumber}
# {$HarvestResult.State}
# {$HarvestResult.CreationDate,yyyyMMdd}
# {$HarvestResult.DerivedFrom}
# {$HarvestResult.ProvenanceNote}
# {$HarvestResource.Oid}
# {$HarvestResource.Name}
# {$HarvestResource.Length}
# {$HarvestResource.StatusCode}
# {$ArcHarvestResource.FileDate}

harvestResourceUrlMapper.urlMap=http://localhost.archive.org:8080/wayback
/wayback/{$ArcHarvestResource.FileDate}/{$HarvestResource.Name}
```

The **QualityReviewToolController** settings control whether the standard browse tool, and external access tool, or both are available to the user. The **ArchiveUrl** setting specifies the location of the archive access tool, to allow the user to view copies of the target already stored in the archive. The **ArchiveName** is the name displayed on the review screen. The **archive.alternative** allows the use of a second review tool, with it's corresponding name. The alternative can be commented out in the configuration if it is not required.

The **harvestResourceUrlMapper** is responsible for writing the access tool URLs in with the review tool using a custom url and replacing elements of that url with the correct items in the harvest resource.

The urlMap property of the **harvestResourceUrlMapper** can have any of the following substituted value from the harvest resource:

- {$HarvestResource.Name}
- {$HarvestResource.Length}
- {$HarvestResource.Oid}
- {$HarvestResource.StatusCode}
- {$ArcHarvestResource.FileDate}

- {$HarvestResult.CreationDate[,DateFormat]}
- {$HarvestResult.DerivedFrom}
- {$HarvestResult.HarvestNumber}
- {$HarvestResult.Oid}
- {$HarvestResult.ProvenanceNote}
- {$HarvestResult.State}

The HarvestResult.CreationDate substitution's format can be controlled by supplying a valid simple date format after a comma within the curly brackets e.g. {$HarvestResult.CreationDate,ddMMyy } for 1 Nov 2008 will show "011108".

The **QualityReviewController.enableAccessTool** and **HarvestResourceUrlMapper** settings can be used to allow Wayback to be used as an access tool for the WCT; either instead of, or in addition to the standard Browse tool. An example of how this may be achieved is detailed on the WCT Wiki. See https://github.com/DIA-NZ/webcurator/wiki/Wayback-Integration.

Note that if Wayback is being used as an access tool, the WaybackIndexer must be enabled and configured (see wct-das.properties below and https://github.com/DIA-NZ/webcurator/wiki/Wayback-Integration).

Web Curator Core - wct-core-security.xml

The **wct-core-security.xml** contains all of the security, Authentication and Authorisation settings to be used by the Web Curator Tool.

```
<bean id="authenticationManager"
class="org.acegisecurity.providers.ProviderManager" abstract="false"
singleton="true" lazy-init="default" autowire="default"
dependency-check="default">
    <property name="providers">
        <list>
            <ref bean="ldapAuthenticator" />
            <ref bean="daoAuthenticationProvider" />
        </list>
    </property>
</bean>
```

This is where the **LDAPAuthenticator** can be plugged in if the Tool is to use an external Directory service for Authentication. In wct-core.properties, set the following parameters:

```
#LDAP Settings
ldap.url=ldap://yourldaphost.domain.com:389
ldap.dn=cn={0},OU=OrgUnit,O=Organisation
```

Web Curator Digital Asset Store - wct-das.properties

```
#WctCoreWsEndpoint

wctCoreWsEndpoint.service=/wct/services/urn:WebCuratorTool
wctCoreWsEndpoint.host=localhost
wctCoreWsEndpoint.port=8080
```

This section of the file specifies the service, hostname and port for the WCTCore component.

```
#ArcDigitalAssetStoreService

# the base directory for the arc store
arcDigitalAssetStoreService.baseDir=/wct/store

# The file mover type to use for this installation (uncomment only one
line).
# For use when the DAS attachments directory is on a different
filesystem than the store directory.
arcDigitalAssetStoreService.dasFileMover=inputStreamDasFileMover
# For use when the DAS attachments directory is on the same filesystem
than the store directory.
##arcDigitalAssetStoreService.dasFileMover=renameDasFilemover

# The archive type to use for this installation (one of: fileArchive,
omsArchive, dpsArchive).
arcDigitalAssetStoreService.archive=fileArchive
```

This section of the file specifies the location where Archives are stored on the file system. The Digital Asset store holds these files for a period of time before they are purged. See the wct-core.properties file for the purge parameters.

Using the File Archive Adapter (Default option)

```
#File Archive

fileArchive.archiveRepository=/wct/filestore
fileArchive.archiveLogReportFiles=crawl.log,progress-statistics.log,local-errors.log,
↪runtime-errors.log,uri-errors.log,hosts-report.txt,mimetype-report.txt,responsecode-
↪report.txt,seeds-report.txt,processors-report.txt
fileArchive.archiveLogDirectory=logs
fileArchive.archiveReportDirectory=reports
fileArchive.archiveArcDirectory=arcs
```

The **FileArchive** writes files to a file system when they are archived. This directory should be permanent storage that is backed up, as these files are the definitive web archives that user wishes to store for prosperity.

Using other Archive Adapters

Other archive adapters may be specified by modifying the arcDigitalAssetStoreService.archive property. Current available types are fileArchive, omsArchive, dpsArchive.

Additional Indexers

```
#WaybackIndexer

# Enable this indexer
waybackIndexer.enabled=false
# Frequency of checks on the merged folder (milliseconds)
waybackIndexer.waittime=1000
# Time to wait for the file to be indexed before giving up
(milliseconds)
waybackIndexer.timeout=300000
```

(continues on next page)

(continued from previous page)

```
# Location of the folder Wayback is watching for auto indexing
waybackIndexer.waybackInputFolder=/tmp/wayback/arcs
# Location of the folder where Wayback places merged indexes
waybackIndexer.waybackMergedFolder=/tmp/wayback/index-data/merged
# Location of the folder where Wayback places failed indexes
waybackIndexer.waybackFailedFolder=/tmp/wayback/index-data/failed

#CDXIndexer
# Enable this indexer
cdxIndexer.enabled=false
```

This section of the file allows configuration of additional indexers, which run concurrently with the standard WCT indexer. There are currently two additional indexers available (both disabled by default):

- **WaybackIndexer** configures WCT to make copies of the ARC or WARC files and move them to the **waybackInputFolder** for automatic indexing by an installed Wayback instance. Wayback will eventually deposit a file of the same name in either the **waybackMergedFolder** (if successful) or the **waybackFailedFolder** (if unsuccessful). This action triggers the indexing complete message.
- **CDXIndexer** generates a CDX index file in the same folder as the ARC/WARC files. When a target instance is submitted to the archive, the CDX index will be copied along with the ARC/WARC file(s).

Web Curator Harvest Agent - wct-agent.properties

The configuration for the Heritrix 1 and Heritrix 3 harvest agent is stored within the /WEB-INF/classes/wct-agent.properties file.

```
#HarvestAgent

# name of the directory where the temporary harvest data is stored
harvestAgent.baseHarvestDirectory=/wct/harvest-agent
# agent host name or ip address that the core knows about
harvestAgent.host=localhost
# the port the agent is listening on for http connections
harvestAgent.port=8080
# the name of the harvest agent web service
harvestAgent.service=/harvest-agent-h3/services/urn:HarvestAgent
# the name of the harvest agent log reader web service
harvestAgent.logReaderService=/harvest-agent-h3/services/urn:LogReader
# the max number of harvest to be run concurrently on this agent
harvestAgent.maxHarvests=2
# the name of the agent. must be unique
harvestAgent.name=My local Agent
# the note to send with the harvest result.
harvestAgent.provenanceNote=Original Harvest
# the number of alerts that occur before a notification is sent
harvestAgent.alertThreshold=200
```

The **HarvestAgent** is responsible for specifying where the harvest agent is located and its name. This is also where the agent specifies the maximum number of concurrent harvests it can carry out.

```
# whether to attempt to recover running harvests from H3 instance on startup.
harvestAgent.attemptHarvestRecovery=true
```

The **attemptHarvestRecovery** is responsible for triggering a harvest recovery in the Heritrix 3 Harvest Agent. This checks for running harvests in WCT-Core and Heritrix 3 and resumes them. This allows for restarting of the H3

Harvest Agent without orphaning the running jobs in Heritrix 3.

```
#HarvestCoordinatorNotifier

# the name of the core harvest agent listener web service
harvestCoordinatorNotifier.service=/wct/services/urn:WebCuratorTool
# the host name or ip address of the core
harvestCoordinatorNotifier.host=localhost
# the port that the core is listening on for http connections
harvestCoordinatorNotifier.port=8080
```

The **harvestCoordinatorNotifier** section is used to specify how the Harvest Agent should communicate back to the WCT Core.

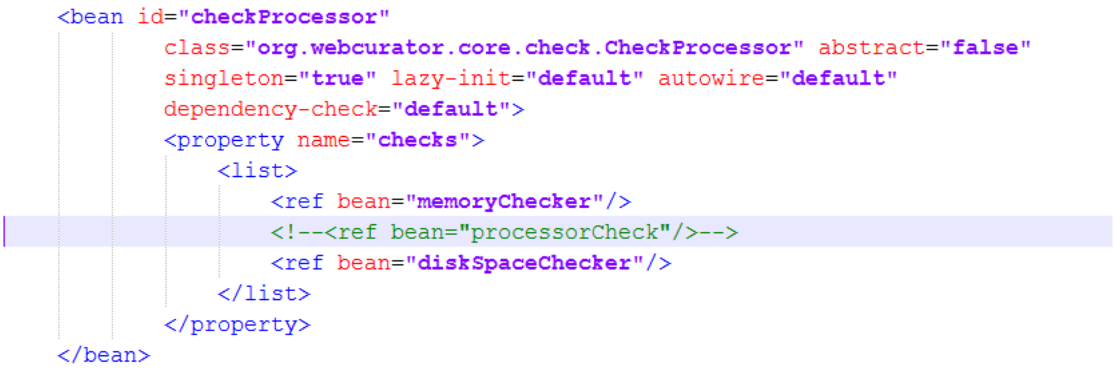
```
#DigitalAssetStore

# the name of the digital asset store web service
digitalAssetStore.service=/wct-store/services/urn:DigitalAssetStore
# the host name or ip address of the digital asset store
digitalAssetStore.host=localhost
# the port that the digital asset store is listening on for http
connections
digitalAssetStore.port=8080
```

The **digitalAssetStore** section is used to specify how the Harvest Agent communicates back to the Digital Asset Store.

```
#MemoryChecker

# The amount of memory in KB that can be used before a warning
notification is sent
memoryChecker.warnThreshold=512000
# The amount of memory in KB that can be used before an error
notification is sent
memoryChecker.errorThreshold=640000

#ProcessorCheck

# The minimum percentage of processor available before a warning
notification is sent
processorCheck.warnThreshold=30
# The minimum percentage of processor available before an error
notification is sent
processorCheck.errorThreshold=20

#DiskSpaceChecker

# the percentage of disk used before a warning notification is sent
diskSpaceChecker.warnThreshold=80
# the percentage of disk used before an error notification is sent
diskSpaceChecker.errorThreshold=90
```

The three checker beans allow the Harvest Agent to monitor Disk, Processor and Memory. Each of the checkers are configurable to allow different alert and error thresholds. A Notification event will be sent on either the alert or error threshold being exceeded.

From release 1.5.2 onwards, the processorCheck bean has been disabled by default. This was done by commenting out the relevant line in the file wct-agent.xml as follows;



```
<bean id="checkProcessor"
        class="org.webcurator.core.check.CheckProcessor" abstract="false"
        singleton="true" lazy-init="default" autowire="default"
        dependency-check="default">
        <property name="checks">
            <list>
                <ref bean="memoryChecker"/>
                <!--<ref bean="processorCheck"/>-->
                <ref bean="diskSpaceChecker"/>
            </list>
        </property>
</bean>
```

It should be noted that the processorCheck bean actually runs the following Unix command line utility to determine processor utilisation - (this command fails when running on Windows hosts);

> **“sar -u”**

Web Curator Harvest Agent - wct-agent.xml

The configuration for the harvest agent is stored within the /WEB-INF/classes/wct-agent.xml file.

If this harvest agent can only harvest material for a set number of agencies, then they can be listed in the *allowedAgencies* property. An empty list implies that any Agency can use the Harvest Agent. The configuration below shows two agencies defined

```
<property name="allowedAgencies">
    <list>
        <value>National Library of New Zealand</value>
        <value>British Library</value>
    </list>
</property>
```

Web Curator Tool - SOAP Service Configuration

The /WEB-INF/server-config.wsdd

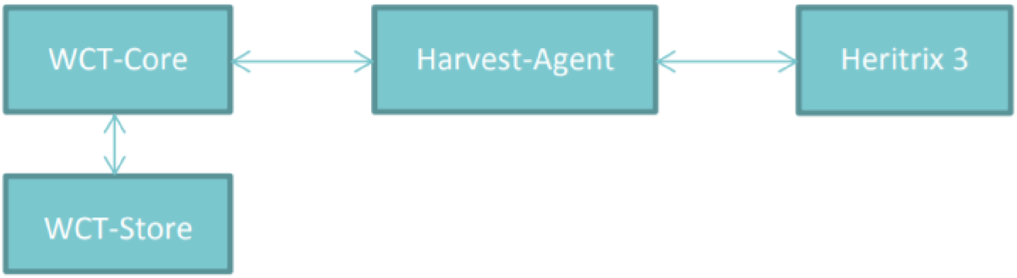
All three components have a server-config.wsdd file. This file is used by Apache Axis to configure the SOAP services used within the Web Curator Tool.

The only attribute that should be modified in the Axis configuration is the location of the temporary directory that Axis should use for attachments. Make sure that this directory exists and is accessible to the Apache Tomcat server.

```
<parameter name="attachments.Directory" value="/tmp/attach"/>
```

5.6 Setting up Heritrix 3

5.6.1 Integration with WCT



Heritrix 3 (H3) integrates with WCT through the new H3-Harvest-Agent. As an interface between WCT-Core and Heritrix 3, the Harvest Agent has three primary functions:

- actioning crawl commands from the WCT UI (start, stop, pause, abort).
- retrieving job status updates from Heritrix 3, to send onto WCT-Core.
- copying completed harvest files from Heritrix 3 job directory to WCT-Store.

Previously Heritrix (v1.14) was bundled within the Harvest Agent, as a .jar dependency. Heritrix 3 is now a standalone application external from WCT.

The H3 Harvest Agent requires a corresponding Heritrix 3 instance to be running. If Heritrix 3 is not runnning then new Target Instances will fail to start crawling.

5.6.2 Prerequisites

- **Java** - A minimum of Java 7 is required. However due to an https issue with H3, it is recommended to and run it using Java 8.

  For simplicity, it is recommended to run Heritrix 3 using the same Java version as WCT, which is now 64bit Java 8.

5.6.3 Download

The Heritrix 3 Github wiki contains a section detailing the current master builds available https://github.com/internetarchive/heritrix3/wiki#master-builds

For the latest official stable builds visit: https://builds.archive.org/job/Heritrix-3/lastStableBuild/org.archive.heritrix%24heritrix/

Note - *the official releases available in the Github repository are not up to date, with the latest being 3.2.0*

Other versions

Heritrix 3.3.0-LBS-2016-02 - From the National Library of Iceland, a stable version based on the Heritrix 3.3.0 master from May 2016. https://github.com/internetarchive/heritrix3/wiki#heritrix-330-lbs-2016-02-may-2016

Building from source

Optionally, Heritrix 3 can be built from source. Use the Github repository: https://github.com/internetarchive/heritrix3/

Maven is required to build the project

The build of the Heritrix3 crawler is done from the directory that contains the cloned Heritrix3 github repository.

It's recommended to skip the tests when building the Heritrix3 crawler as they can take a considerable amount of time to run (many minutes to hours).

```
mvn clean install -DskipTests=true
```

The build produces a *heritrix-<heritrix-version>-SNAPSHOT-dist.zip* in *./dist/target*.

Unzip this zip in the parent folder of *$HERITRIX_HOME*.

5.6.4 Configuration

Location

It is recommened to run Heritrix 3 as close to it's corresponding H3 Harvest Agent as possible, i.e. the same server. Running Heritrix 3 and the H3 Harvest Agent on separate servers has not been tested.

Memory

If Heritrix 3 and it's corresponding Harvest Agent are running on the same server as WCT Core and Store, then Heritrix 3 may need greater memory allocation.

Or depending on how many concurrent harvests you want to allow the H3 Harvest Agent to run, increasing the memory allocation for Heritrix 3 might be required.

Place the following lines near the top of *heritrix-3.3.0/bin/heritrix*

```
#Java Configuration
JAVA_OPTS=" -Xms256m -Xmx1024m"
```

Or set the JAVA_OPTS environment variable on the command line prior to running the Heritrix startup script:

```
export JAVA_OPTS=" -Xms256m -Xmx1024m"
```

Jobs directory

Heritrix 3 creates a folder in it's job directory for each new job. After the registering of a new job in Heritrix 3 by the H3 Harvest Agent, the Agent completes the initial setup by copying the crawl profile (`crawler-beans.cxml`) and seeds (`seeds.txt`) into the new job folder.

The Apache Tomcat running the H3 Harvest Agent **must have read and write access** to the top level jobs directory (and any child job folders) for Heritrix 3.

On completion or termination of a Heritrix 3 job, the H3 Harvest Agent will attempt to clean up by removing the job folder.

The Heritrix 3 jobs directory must remain separate from the H3 Harvest Agent **harvestAgent.baseHarvestDirectory**. *If the same directory is used, an empty profile will be given to Heritrix 3, causing a job to fail.*

Scripts directory

The H3 scripts directory is used for storing pre-defined Heritrix 3 scripts (js, groovy, beanshell) that WCT makes available for use through the scripting console window. These scripts can be run against harvests running on Heritrix 3.

- The directory needs to be readable by the user running Tomcat.
- The directory path needs to be set in **wct-core.properties.**

For more information, please see:

- https://github.com/internetarchive/heritrix3/wiki/Heritrix3-Useful-Scripts
- https://heritrix.readthedocs.io/en/latest/api.html#execute-script-in-job

Default profile

There are only a select group of Heritrix 3 profile settings available through the WCT UI to configure. If configuration of additional settings is required, then the default Heritrix 3 profile used by WCT can be edited. **This is only recommened for advanced users.**

The default profile is located in the project source:

```
harvest-agent-h3/build/defaultH3Profile.cxml
```

The H3 Harvest Agent must be re-built to include any changes to the default profile.

Care must be taken if editing the default profile xml. The WCT Heritrix 3 profile editor relies on a select group of xml elements being present and correctly formatted. The following list of xml elements must remain untouched in the xml. Other properties can be edited.

- Where properties are shown, WCT edits those values
- Where just the bean is shown, with no properties, WCT edits the entire bean element.

```
<bean id="metadata" class="org.archive.modules.CrawlMetadata" autowire="byName">
    <!-- <property name="robotsPolicyName" value="obey"/> -->
    <!-- <property name="userAgentTemplate" value="Mozilla/5.0 (compatible; heritrix/
↪@VERSION@ +@OPERATOR_CONTACT_URL@)"/> -->
</bean>

...

<bean class="org.archive.modules.deciderules.TooManyHopsDecideRule">
    <!-- <property name="maxHops" value="20" /> -->
</bean>

...

<bean class="org.archive.modules.deciderules.TransclusionDecideRule">
    <!-- <property name="maxTransHops" value="2" /> -->
</bean>
```

(continues on next page)

(continued from previous page)

```
...

<bean class="org.archive.modules.deciderules.TooManyPathSegmentsDecideRule">
    <!-- <property name="maxPathDepth" value="20" /> -->
</bean>

...

<bean class="org.archive.modules.deciderules.MatchesListRegexDecideRule">
</bean>

...

<bean id="fetchHttp" class="org.archive.modules.fetcher.FetchHTTP">
    <!-- <property name="defaultEncoding" value="ISO-8859-1" /> -->
    <!-- <property name="ignoreCookies" value="false" /> -->
</bean>

...

<bean id="warcWriter" class="org.archive.modules.writer.WARCWriterProcessor">
    <!-- <property name="compress" value="true" /> -->
    <!-- <property name="prefix" value="IAH" /> -->
    <!-- <property name="maxFileSizeBytes" value="1000000000" /> -->
</bean>

...

<bean id="crawlLimiter" class="org.archive.crawler.framework.CrawlLimitEnforcer">
    <!-- <property name="maxBytesDownload" value="0" /> -->
    <!-- <property name="maxDocumentsDownload" value="0" /> -->
    <!-- <property name="maxTimeSeconds" value="0" /> -->
</bean>

...

<bean id="disposition" class="org.archive.crawler.postprocessor.DispositionProcessor">
    <!-- <property name="delayFactor" value="5.0" /> -->
    <!-- <property name="minDelayMs" value="3000" /> -->
    <!-- <property name="respectCrawlDelayUpToSeconds" value="300" /> -->
    <!-- <property name="maxDelayMs" value="30000" /> -->
    <!-- <property name="maxPerHostBandwidthUsageKbSec" value="0" /> -->
</bean>
```

Proxy Access

Configuring Heritrix 3 for proxy access also requires editing of the default Heritrix 3 profile.

The default profile is located in the project source:

```
harvest-agent-h3/build/defaultH3Profile.cxml
```

The H3 Harvest Agent must be re-built to include any changes to the default profile.

Care must be taken if editing the default profile xml. The WCT Heritrix 3 profile editor relies on a select group of xml elements being present and correctly formatted.

The following properties in the `fetchHTTP` bean can configured for web proxy access:

```
<bean id="fetchHttp" class="org.archive.modules.fetcher.FetchHTTP">
    <!-- <property name="httpProxyHost" value="" /> -->
    <!-- <property name="httpProxyPort" value="0" /> -->
    <!-- <property name="httpProxyUser" value="" /> -->
    <!-- <property name="httpProxyPassword" value="" /> -->
</bean>
```

JMX setup for Heritrix 3

Ensure that the *JMX setup* has been completed for Heritrix 3.

5.6.5 Running Heritrix 3

Credentials

By default the H3 Harvest Agent is configured to connect to H3 using:

- username: admin
- password: admin

Starting Heritrix 3

- **Linux/Unix** `./heritrix-3.3.0/bin/heritrix -a admin:admin -j /mnt/wct-harvester/dev/heritrix3`
- **Windows** `./heritrix-3.3.0/bin/heritrix.cmd -a admin:admin -j /mnt/wct-harvester/dev/heritrix3`

Stopping Heritrix 3

Heritrix 3 can be stopped using two methods:

- **Via the UI**. This will notify you of any jobs still running.
- **Kill the Java process**. Your responsibility to check for and stop any running jobs.

5.6.6 Operation of Heritrix 3

Jobs

Two types of jobs are created in Heritrix 3 by the H3 Harvest Agent:

- **Crawl Jobs** - standard crawl jobs for WCT Target Instances. Created for the duration of running crawls.
- **Profile Validation Jobs** - a single re-used job to validate Heritrix 3 profiles created/edited in WCT-Core.

Heritrix management UI

Accessible via https://localhost:8443/engine

Logging

The Heritrix 3 output log can be located in the `heritrix-3.3.0/heritrix_out.log` file.

Additional notes

TODO Does this still apply?

This Harvest Agent implementation handles the creation and cleanup up of jobs within the Heritrix 3.x instance. You should only see job directories within Heritrix while a harvest is running or waiting to be completed. Once the harvest is complete and WCT has transferred the assets, logs and reports to the Store then the Heritrix job is torn down and directory deleted. The only occasions where a Heritrix job directory will not be cleaned up is if a job fails to build/start or an error has occurred during the harvest. This allows you to investigate the Heritrix job log to determine the cause.

5.6.7 Troubleshooting

TODO

- Gathering information from logs.
- When things don't work - what to check.
- Heritrix 3 won't crawl.
- This information might be better presented in a table.

Interacting with Heritrix 3 directly

Heritrix 3 can be operated directly (outside of WCT). Either use the UI or REST API to manually start a crawl. TODO Does this work?

Curl can be used to send actions to H3. See https://webarchive.jira.com/wiki/spaces/Heritrix/pages/5735014/Heritrix+3.x+API+Guide for details on how this is done.

Jobs won't build

- Check the Heritrix log, *heritrix_log.out*.
- Is the *seed.txt* and *crawler-beans.cxml* being created in the harvest agent base directory, is it being transferred to the H3 job dir location?
- Check file permissions.

Jobs fail

- Fail to build
- Fail during crawl

TODO How to solve.

Old job dirs not being removed

Occasionaly there are nfs hidden files that prevent these folders from deleting fully. Make sure all hidden files are removed.

Web proxy access

TODO Describe how to deal with web proxy access.

OpenSSL errors with Solaris and Java 7

If running on Solaris with Java 7 and you get openssl errors when the Harvest Agent tries to connect the Heritrix 3.x, try running Heritrix 3.x with Java 8.

Copying issues with larger harvests

If running Apache Tomcat with 32bit Java 7, you may experience issues with larger harvests copying between the Harvest Agent and the Store on completion of a crawl. This was resolved by running Apache Tomcat with 64bit Java 7.

5.7 Graceful shutdown and restart

The system can be taken down manually or automatically for maintenance.

To shut down and restart the Core and the DAS, but leave the harvesters running (so that they can continue harvesting when the Core and DAS are unavailable), follow these steps:

1. Admin or script shuts down Tomcat on the server that hosts Core and DAS.
2. Admin or script shuts down Oracle.
3. Admin or script does backup or whatever. WCT Agents continue harvesting.
4. Admin or script starts Oracle.
5. Admin or script starts Tomcat.
6. WCT Harvest Agents re-register themselves with WCT Core, and then copy any completed harvests to DAS and notify Core.

To shut down everything including the harvest agents, then the procedure is:

1. Wait until harvest agents have no crawl jobs running and shut them down (either directly or Tomcat container). This can be best achieved by halting all Scheduled and Queued target instances using the ‘Calendar’ icon on the Harvester Configuration screen, and then waiting until the currently running jobs finish.
2. Admin shuts down Tomcat on the server that hosts Core and DAS.
3. Admin shuts down database.

Restart the system again in the reverse order.

Note that when you shut down a harvest agent, running jobs are lost (when the agent restarts it does not know how to restart the harvest. If you pause a harvest (or all the harvests) then it stays in a paused state on the harvest agent, and is similarly lost when you shut down.

5.8 Appendix A: Creating a truststore and importing a certificate

To create a truststore and import a certificate:

1. First export your public key from your Directory server.

    - Refer to the documentation from your Directory server, in order to complete this task.
    - If possible export the certificate as a binary file. We will assume your exported certificate is called mydirectorycert.der

2. Create a truststore and dummy key. Using the keytool provided with the java SDK:

```
keytool -genkey -dname "cn=dummy, ou=dummy, o=dummy, c=US" -alias dummy -keypass␣
↪dummy -keystore /var/wctcore/ssl/wct.ts -storepass password
```

5. You need to import the X509 certificate for your directory server:

```
keytool -import -file mydirectorycert.der -keystore
/var/wctcore/ssl/wct.ts
```

5.9 Appendix B: The OMS archive adapter

The OMSArchive bean is only used for the National Library of New Zealand to archive files into their Object Management System. For all other implementations the more generic FileSystemArchive Bean should be used.

To enable the OMS Archive, set the **archive** property in the **arcDigitalAssetStoreService** section of wct-das.properties to **omsArchive**.

```
#OMS Archive

omsArchive.archiveLogReportFiles=crawl.log,progress-statistics.log,local-errors.log,
↪runtime-errors.log,uri-errors.log,hosts-report.txt,mimetype-report.txt,responsecode-
↪report.txt,seeds-report.txt,processors-report.txt
omsArchive.url= http://omsserver/oms/upload
omsArchive.partSize=1000000
omsArchive.ilsTapuhiFlag=RT_ILS
omsArchive.collectionType=CT_EPB
omsArchive.objectType=OT_WWW
omsArchive.agencyResponsible=AR_NLNZ
omsArchive.instanceRole=IRC_PM
omsArchive.instanceCaptureSystem=CS_HER
omsArchive.instanceType=IT_COM
omsArchive.user_group=4
omsArchive.user=username
omsArchive.password=password
```

CHAPTER 6

Upgrade Guide

6.1 Introduction

This guide, intended for system administrators, covers upgrade of the Web Curator Tool from version 1.6.2 to version 2.0. If you are on an earlier version you can still follow these instructions to upgrade the database, but you will need to manually merge your old configuration files with the new files, or configure your installation from scratch.

For information on how to install and setup the Web Curator Tool from scratch, see the Web Curator Tool System Administrator Guide. For information about developing and contributing to the Web Curator Tool, see the Developer Guide. For information on using the Web Curator Tool, see the Web Curator Tool Quick User Guide and the Web Curator Tool online help.

The source for both code and documentation for the Web Curator Tool can be found at http://dia-nz.github.io/webcurator/.

6.1.1 Contents of this document

Following this introduction, the Web Curator Tool Upgrade Guide includes the following sections:

- **Upgrade requirements** - Covers requirements for upgrading.
- **Shut Down the WCT** - Describes shutting down WCT prior to upgrading.
- **Upgrading the WCT database schema** - Describes how to upgrade the database schema.
- **Upgrading the application** - How to upgrade the application.
- **Configuration** - New configuration parameters.
- **Post-upgrade notes** - Additional post migration steps.

6.2 Upgrade requirements

The following section explains the requirements for upgrading to version 2.0 of the Web Curator Tool.

6.2.1 Prerequisites

The following are required to successfully upgrade the Web Curator Tool to version 2.0:

- Installed and running version of the Web Curator Tool – version 1.6.2 (or older) running against Oracle *11g* or newer, PostgreSQL *8.4.9* or newer, or MySQL *5.0.95* or newer.
- Access to the Tomcat server(s) for the Core, Digital Asset Store, and Harvest Agent components.

Note that the Web Curator Tool has been tested with Oracle ‘11g‘, PostgreSQL ‘8.4.9‘ and ‘9.6.11‘, MySQL ‘5.0.95‘ and MariaDB ‘10.0.36‘, although newer versions of these products are expected to work as well. Due to the use of Hibernate for database persistence other database platforms should work, if the product is rebuilt with the correct database dialect, using the required JDBC driver. However, only MySQL, PostgreSQL and Oracle have been tested.

6.3 Shut Down the WCT

The major components to the deployment of the Web Curator Tool are:

- The web curator core (*wct.war*).
- The web curator harvest agent for Heritrix 1 (*harvest-agent-h1.war*, optional, only needed if Heritrix 1 support is desired).
- The web curator harvest agent for Heritrix 3 (*harvest-agent-h3.war*).
- The web curator digital asset store (*wct-store.war*).

Note that the *wct-agent.war* module has been replaced by two new modules *harvest-agent-h1.war* and *harvest-agent-h3.war*.

This document assumes that 1.6.2 (or an earlier version) is currently deployed to your Tomcat instance.

To begin the upgrade of the WCT to version 2.0:

1. Make sure that all target instances have completed.
2. Shut down the Tomcat instance(s) running the Harvest Agents, WCT Core, and Digital Asset Store.

6.4 Upgrading WCT Database Schema

Version 2.0 of the Web Curator Tool is supported under MySQL *5.0.95* and up, Oracle *11g* and up, and PostgreSQL *8.4.9* and up. Database schema upgrade scripts have been provided for all three databases.

6.4.1 Upgrade scripts

To upgrade from an older version to 2.0, you first need to upgrade to version 1.6.2 (which is actually version 1.6.1 of the database schema, since there were no changes to the schema between 1.6.1 and 1.6.2). The scripts for upgrading to 1.6.2 can be found in *wct-core/db/legacy/upgrade*. The scripts that get you from 1.6.2 to 2.0 are located in *wct-core/db/latest/upgrade*.

Upgrade script names are of the format:

```
upgrade-<database-type>-<source-version>-to-<target-version>.sql
```

where <*database-type*> is one of *mysql*, *oracle* or *postgres*.

The <*source-version*> is the current or source version (the version you're migrating *from*).

The <*target-version*> is the target version (the version you're migrating *to*).

No script means no database change. *If there is no script for a particular version it means that there were no database changes.*

6.4.2 Upgrades are incremental

Upgrade scripts only cover a single upgrade step from one version to another. This means that upgrading across several versions requires that all the scripts between the source and target version be executed in sequence.

For example, to upgrade a MySQL database from version 1.4.0 to 2.0, the following scripts would need to be executed in this order:

From db/legacy/upgrade:

1. *upgrade-mysql-1_4-to-1_4_1.sql*
2. *upgrade-mysql-1_5-to-1_5_1.sql*
3. *upgrade-mysql-1_5_1-to-1_5_2.sql*
4. *upgrade-mysql-1_5_2-to-1_6.sql*
5. *upgrade-mysql-1_6-to-1_6_1.sql*

Then, from db/latest/upgrade:

1. *upgrade-mysql-1_6_1-to-2_0.sql*

Note that some scripts may complain about columns already existing or timestamp column definitions having the wrong precision. You can safely ignore these errors. You might also get warnings about implicit indexes being created. These are harmless as well.

6.4.3 Upgrading on Oracle

This guide assumes that the source version's schema is already configured on your Oracle database under the schema *DB_WCT*.

1. Log on to the database using the *DB_WCT* user.
2. Run the following SQL to upgrade the database:

```
db[/legacy]/upgrade/upgrade-oracle-<source-version>-to-<target-version>.sql

SQL> conn db_wct@<sid-name>

SQL> @upgrade-oracle-<source-version>-to-<target-version>.sql

SQL> exit;
```

6.4.4 Upgrading on PostgreSQL

This guide assumes that the source version's schema is already configured on your PostgreSQL database under the schema *DB_WCT*.

1. Log on to the database using the *postgres* user.

2. Run the following SQL to upgrade the database:

```
db[/legacy]/upgrade/upgrade-postgresql-<source-version>-to-<target-version>.sql

postgres=# \c Dwct

postgres=# \i upgrade-postgresql-<source-version>-to-<target-version>.sql

postgres=# \q
```

6.4.5 Upgrading on MySQL

This guide assumes that the previous version's schema is already configured on your MySQL database under the schema *DB_WCT*.

1. Log on to the database using the *root* user.
2. Run the following SQL to upgrade the database:

```
db[/legacy]\upgrade\upgrade-mysql-<source-version>-to-<target-version>.sql

mysql> use db_wct

mysql> source upgrade-mysql-<source-version>-to-<target-version>.sql

mysql> quit
```

6.5 Upgrading the application

6.5.1 Deploying WCT to Tomcat

3. Copy any settings/properties/configuration files you wish to keep from the Apache Tomcat webapps directory.
4. Remove the applications from the Apache Tomcat webapps directory, including the expanded directory and WAR files.
5. Copy the version 2.0 WAR files into the Apache Tomcat webapps folder.
6. If your Tomcat instance is not set to auto-deploy then expand the WAR files as follows:

```
cd $TOMCAT/webapps

mkdir wct

cd wct

$JAVA_HOME/bin/jar xvf ../wct.war

cd $TOMCAT/webapps

mkdir wct-harvest-agent

cd wct-harvest-agent

$JAVA_HOME/bin/jar xvf ../wct-harvest-agent.war
```

(continues on next page)

(continued from previous page)

```
cd $TOMCAT/webapps

mkdir wct-store

cd wct-store

$JAVA_HOME/bin/jar xvf ../wct-store.war
```

7. When migrating from 1.6.2: copy any settings/properties/configuration files you backed-up in step 3 back into your Apache Tomcat webapps directory. When migrating from an older version: start from the new configuration files and merge any relevant values from your old configuration files back in.

6.6 Configuration

See the WCT System Administrator Guide for more information about configuring the Web Curator Tool.

Of note, please ensure that the *TOMCAT/webapps/wct/META-INF/context.xml* is updated to correctly identify your database.

The Spring and Log4J XML files should also be checked as per the WCT System Administrator Guide to ensure their values are appropriate for your deployment.

6.6.1 New configuration parameters in 2.0

TOMCAT/webapps/wct/WEB-INF/classes/wct-core.properties

There's a new variable that tells the core where to find its Heritrix 3 scripts (used by the H3 script console).

```
h3.scriptsDirectory=/usr/local/wct/h3scripts
```

TOMCAT/webapps/harvest-agent-h3/WEB-INF/classes/wct-agent.properties

The harvest agent now needs to have a (unique) name and the path of its logReaderService must be specified. (This variable is also needed in the wct-agent.properties file for Heritrix 1 agents.)

```
harvestAgent.service=My Agent
harvestAgent.logReaderService=/harvest-agent-h3/services/urn:LogReader
```

There are now settings that tell the agent how to connect to its Heritrix 3 instance.

```
h3Wrapper.host=localhost
h3Wrapper.port=8443
h3Wrapper.keyStoreFile=
h3Wrapper.keyStorePassword=
h3Wrapper.userName=admin
h3Wrapper.password=admin
```

6.6.2 New configuration parameters in 1.6.3

TOMCAT/webapps/wct-store/WEB-INF/classes/wct-das.properties

Changes required by the National Library of New Zealand to be compatible with archiving to a Rosetta DPS integrated with Alma (library cataloguing and workflow management system from Ex Libris). All changes have been implemented as backward compatible as possible. The exposure of these changes and their configuration are through the files wct-das.properties, wct-das.xml inside WCT-Store.

Setting Mets CMS section

The section used in the DNX TechMD for the CMS data is now configurable. The CMS section can be set to either of the following inside wct-das.properties

```
dpsArchive.cmsSection=CMS
dpsArchive.cmsSystem=ilsdb

OR

dpsArchive.cmsSection=objectIdentifier
dpsArchive.cmsSystem=ALMA
```

Preset producer ID for custom deposit forms

The Producer ID can now be preset for deposits that use a custom form, particularly useful if only one Producer is used and saves the user having to input their Rosetta password each time to search for one. If no Producer ID is set in wct-das.properties then it will revert to the old process of loading a list of available Producers from Rosetta.

```
dpsArchive.htmlSerials.producerIds=11111
```

Toggle HTML Serial agencies using non HTML Serial entity types

Used when a user is under an HTML Serial agency but wants to submit a custom type. Set to *False* to enable the use of custom types.

```
dpsArchive.htmlSerials.restrictAgencyType=true
```

Custom Types

Custom Types for Web Harvests, follow the same method as the htmlSerials. If there are more than one value for each of these, separate them using comma. Make sure there is an equal number of values for each attribute.

```
dpsArchive.webHarvest.customTargetDCTypes=eMonograph
dpsArchive.webHarvest.customerMaterialFlowIds=11111
dpsArchive.webHarvest.customerProducerIds=11111
dpsArchive.webHarvest.customIeEntityTypes=HTMLMonoIE
dpsArchive.webHarvest.customDCTitleSource=TargetName
```

Set source of Mets DC Title for custom types

For custom entity tpes, the field of which the Mets DC Title gets populated with for the mets.xml can now be set. The available fields are the Target Seed Url or the Target Name. This is switched in wct-das.properties.

```
dpsArchive.webHarvest.customDCTitleSource=SeedUrl

OR

dpsArchive.webHarvest.customDCTitleSource=TargetName
```

6.6.3 New configuration parameters in 1.6.2

TOMCAT/webapps/wct-store/WEB-INF/classes/wct-das.properties

There is now the option of setting Rosetta access codes for when archiving harvests to the Rosetta DPS.

```
dpsArchive.dnx_open_access=XXX
dpsArchive.dnx_published_restricted=XXX
dpsArchive.dnx_unpublished_restricted_location=XXX
dpsArchive.dnx_unpublished_restricted_person=XXX
```

These will only be used if the archive type is set to 'dpsArchive'.

```
arcDigitalAssetStoreService.archive=dpsArchive
```

6.6.4 Updating older configurations

To update the configuration files when migrating from versions older than 1.6.2, it is recommended to start from the new configuration files and merge any relevant differences with your existing configuration back in as needed. In most cases new variables have been added. Only rarely have variables been dropped or renamed.

6.7 Post-upgrade notes

Once the Web Curator Tool has been upgraded you will be able to start the Tomcat instances and log in as any of the users that existed prior to the upgrade.

6.7.1 Notes on the Upgrade Effects

Please see the Release Notes for further information regarding the changes introduced in WCT 2.0.

CHAPTER 7

Wayback Integration Guide

7.1 Introduction

In order to use Wayback as an review tool within WCT, you need to deploy and configure an instance of Wayback to run inside your Tomcat container. It is this instance of Wayback that performs the indexing.

This guide shows how to deploy and configure an instance of Wayback to run inside your Tomcat container.

7.1.1 Contents of this document

Following this introduction, the Wayback Integration Guide includes the following sections:

- **Wayback Vs OpenWayback** - Covers the Wayback options.
- **Installation** - Covers installing Wayback.
- **Configuration** - Covers configuring Wayback.
- **Wayback as a Review Tool in WCT** - Covers configuring Wayback for use as a review tool in the Web Curator Tool.
- **Testing** - Covers testing the Wayback installation.
- **More information** - Provides some links for more information.

7.2 Wayback vs OpenWayback vs PyWayback

There are a number of options for Wayback, including Wayback from the Internet Archive, OpenWayback from IIPC and PyWB from Rhizome. Web Curator Tool was originally developed and tested with Wayback however OpenWayback and PyWB are more actively developed at the moment.

Here is an explanation of the history and differences between Wayback and OpenWayback. For documentation on PyWB click here.

7.2.1 Downloading

Download Wayback here.

Download OpenWayback here.

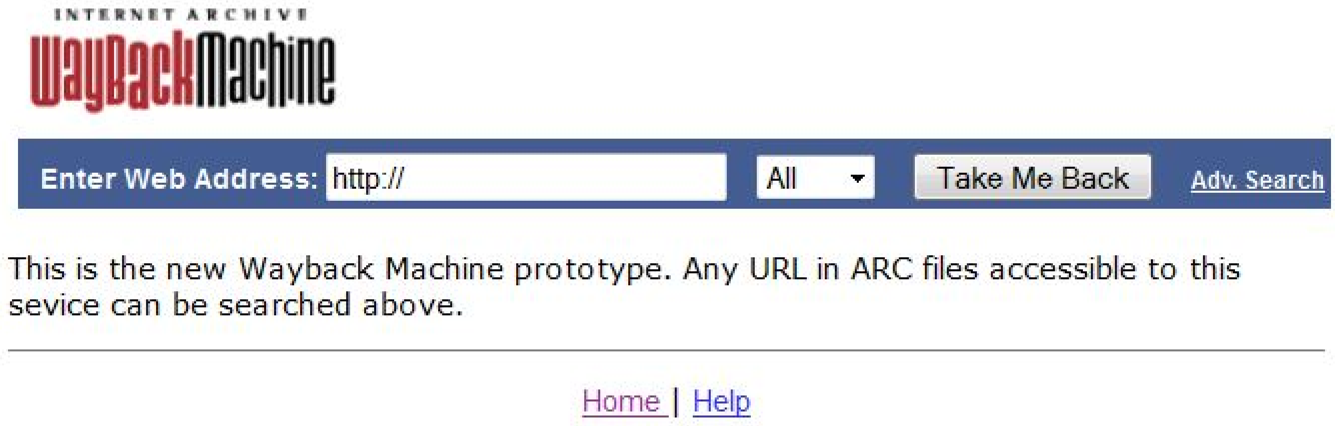
Download PyWB here.

7.3 Installation

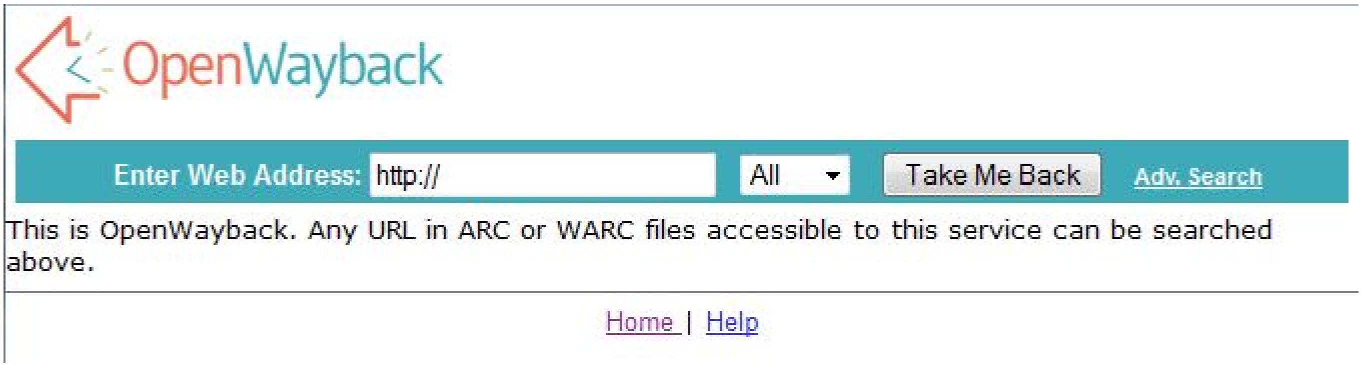
The OpenWayback Wiki contains a useful install guide and configuration guide. These are also relevant for Wayback.

PyWB has useful setup information to get started and configure your archive, keep in mind that PyWB is set up differently to OpenWayback. Unlike OpenWayback, which runs in Tomcat, PyWB runs on its own Gevent server so you will need to keep in mind which port you want to run PyWB on. You will also need to factor in the file structure of the archive and where WCT is storing the warcs.

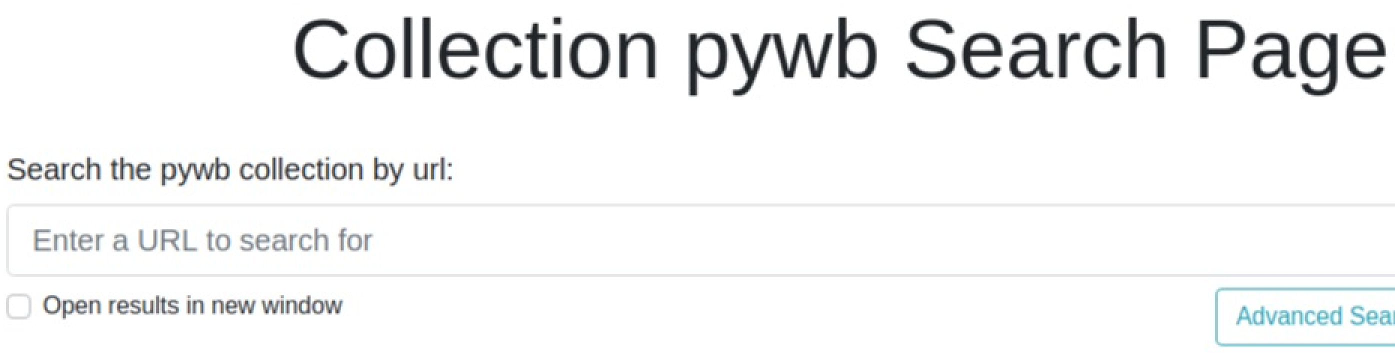
Once you have the tool running in Tomcat or Gevent and can see the homepage in your browser then you are ready to configure the interaction with WCT.



This is the new Wayback Machine prototype. Any URL in ARC files accessible to this sevice can be searched above.



This is OpenWayback. Any URL in ARC or WARC files accessible to this service can be searched above.


Collection pywb Search Page
Search the pywb collection by url:
Enter a URL to search for
□ Open results in new window
Advanced Sear

7.4 Configuration

Please note: From here forward any reference to Wayback applies to OpenWayback and Wayback. This does not cover PyWB configuration.

The easiest configuration to get WCT and Wayback working together is to leave Wayback with its default setting of indexing using BDB (instead of CDX). As Wayback indexes by watching a folder for new files, we need to configure WCT to copy new harvests to a common location between the two. *Note you don't have to move where WCT is storing your harvests, this is an extra location common to WCT and Wayback.*

In this example our common location will be */wct/wayback/*.

- WCT will copy our warc/arc files to */wct/wayback/store/*, and Wayback will be watching this folder for any new files to index.
- The indexes that Wayback creates will be in */wct/wayback/index-data/merged/*.
- Shut down Tomcat
- Open your *wct-das.properties* file and make the following changes. (*wct-das.properties* is located in */<path to tomcat>/webapps/wct-store/WEB-INF/classes/*):

```
#WaybackIndexer
# Enable this indexer
waybackIndexer.enabled=true
# Frequency of checks on the merged folder (milliseconds)
waybackIndexer.waittime=1000
# Time to wait for the file to be indexed before giving up (milliseconds)
waybackIndexer.timeout=300000
# Location of the folder Wayback is watching for auto indexing
waybackIndexer.waybackInputFolder=/wct/wayback/store
# Location of the folder where Wayback places merged indexes
waybackIndexer.waybackMergedFolder=/wct/wayback/index-data/merged
# Location of the folder where Wayback places failed indexes
waybackIndexer.waybackFailedFolder=/wct/wayback/index-data/failed
```

- Open your *wayback.xml* file and change the *wayback.basedir* path. (*wayback.xml* is located in */<path to tomcat>/webapps/wayback/WEB-INF/*):

```
<bean class="org.springframework.beans.factory.config.
↪PropertyPlaceholderConfigurer">
    <property name="properties">
      <value>
```

(continues on next page)

(continued from previous page)

```
        wayback.basedir=/wct/wayback
        wayback.urlprefix=http://localhost:8080/wayback/
      </value>
    </property>
</bean>
```

- Open your *BDBCollection.xml* file and change the prefix property. (*BDBCollection.xml* is located in */<path to tomcat>/webapps/wayback/WEB-INF/*):

```
<bean id="datadirs" class="org.springframework.beans.factory.config.
↪ListFactoryBean">
    <property name="sourceList">
      <list>
        <bean class="org.archive.wayback.resourcestore.resourcefile.
↪DirectoryResourceFileSource">
          <property name="name" value="files1" />
          <property name="prefix" value="${wayback.basedir}/store/" />
          <property name="recurse" value="false" />
        </bean>
      </list>
    </property>
</bean>
```

Inside our common location Wayback will create the following folder structure. (*/index-data/merged/* is where the completed indexes are stored. Their file names exactly match the name of their corresponding warc/arc file, including the extension):

```
file-db/db
file-db/incoming
file-db/state
index
index-data/failed
index-data/incoming
index-data/merged
index-data/queue
index-data/tmp
```

7.5 PyWB Configuration

PyWB is different to OpenWayback and Wayback in that it requires a collection to be initialised, it uses .cdxj as index files, and it runs on a separate Gevent server. *If you intend to use PyWB along with another Wayback tool you might want to either configure the waybackIndexer.waybackInputFolder within wct-das.properties to the initialised PyWB archive directory collections/collectionName/archive or symlink the initialised PyWB archive directory with the directory you have used for waybackIndexer.waybackInputFolder.* This way PyWB will always get a copy of the warc files that are being generated.

When you run the PyWB server you can specify the port using -p. Using -a will ensure that the initialised PyWB archive directory is checked every 30 seconds for new warcs to index. Any warc files that are manually added in will be indexed within *indexes/index.cdxj* and any warc files that are indexed using the autoindex setting will be indexed within *indexes/autoindex.cdxj*.

7.6 Wayback as a Review Tool in WCT

In order to use Wayback as a review tool inside WCT, there are some more configuration changes.

First take note of the url that Wayback is running from inside Tomcat. This should match the *wayback.urlprefix* property we saw above in *wayback.xml*. In our example it is http://localhost:8080/wayback/.

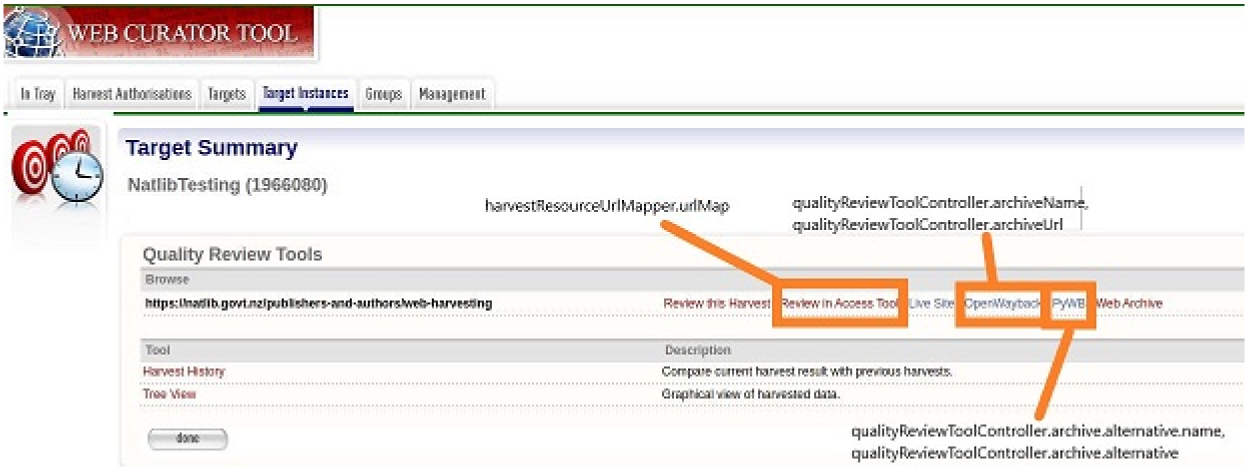
Open your *wct-core.properties* file and make the following changes. (*wct-core.properties* is located in */<path to tomcat>/webapps/wct/WEB-INF/classes/*):

```
harvestResourceUrlMapper.urlMap=http://localhost:8080/wayback/{$ArcHarvestResource.
↪FileDate}/{$HarvestResource.Name}
qualityReviewToolController.enableBrowseTool=true
qualityReviewToolController.enableAccessTool=true
qualityReviewToolController.archiveUrl=http://localhost:8080/wayback/*/
```

7.7 Using Multiple Review Tools in WCT

Within the Target Summary for the harvest you will have options for different Quality Review Tools. There will be a link to Review in Access Tool plus other links to other archives which you can specify the name of. All of these links are configurable via wct-core.properties.

- Review in Access Tool uses the value set in harvestResourceUrlMapper.urlMap
- qualityReviewToolController.archiveName uses the value set in qualityReviewToolController.archiveUrl
- qualityReviewToolController.archive.alternative.name uses the value set in qualityReviewToolController.archive.alternative



Target Summary

NatlibTesting (1966080)

harvestResourceUrlMapper.urlMap

qualityReviewToolController.archiveName,
qualityReviewToolController.archiveUrl

Quality Review Tools

Browse

https://natlib.govt.nz/publishers-and-authors/web-harvesting Review this Harvest Review in Access Tool Live Site OpenWayback PyWB Web Archive

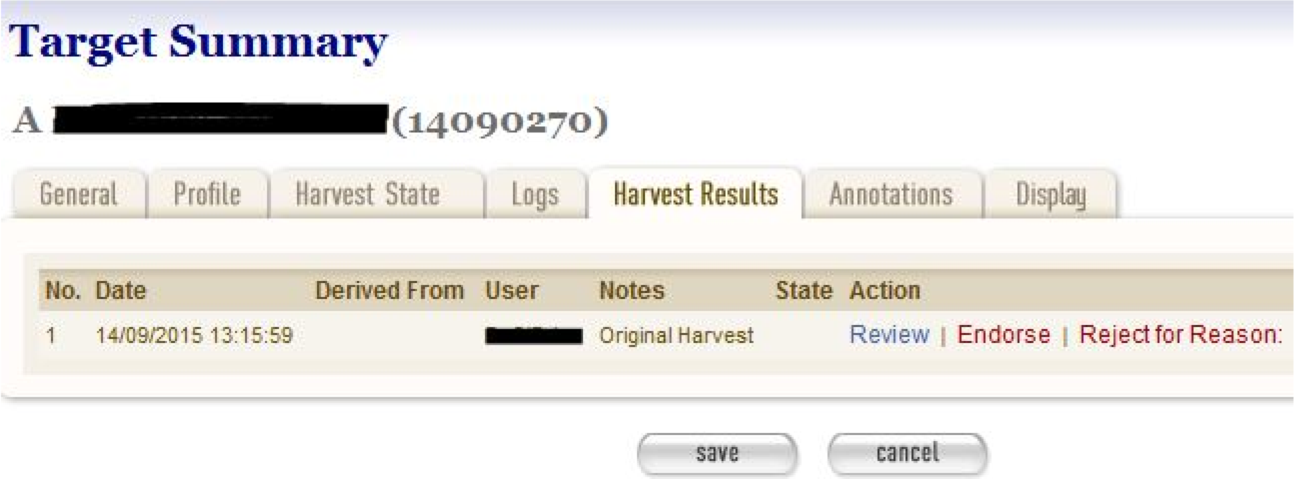
Tool	Description
Harvest History	Compare current harvest result with previous harvests.
Tree View	Graphical view of harvested data.

qualityReviewToolController.archive.alternative.name,
qualityReviewToolController.archive.alternative

7.8 Testing

Once you have restarted Tomcat, schedule a harvest to test the integration.

- When the harvest is completed, you should see it's warc/arc file copied to */wct/wayback/store*
- When the indexing is complete, you should see the index file in */wct/wayback/index-data/merged*
- Inside WCT - Under the *Harvest Results* tab for a Target Instance, *Review* your completed harvest.

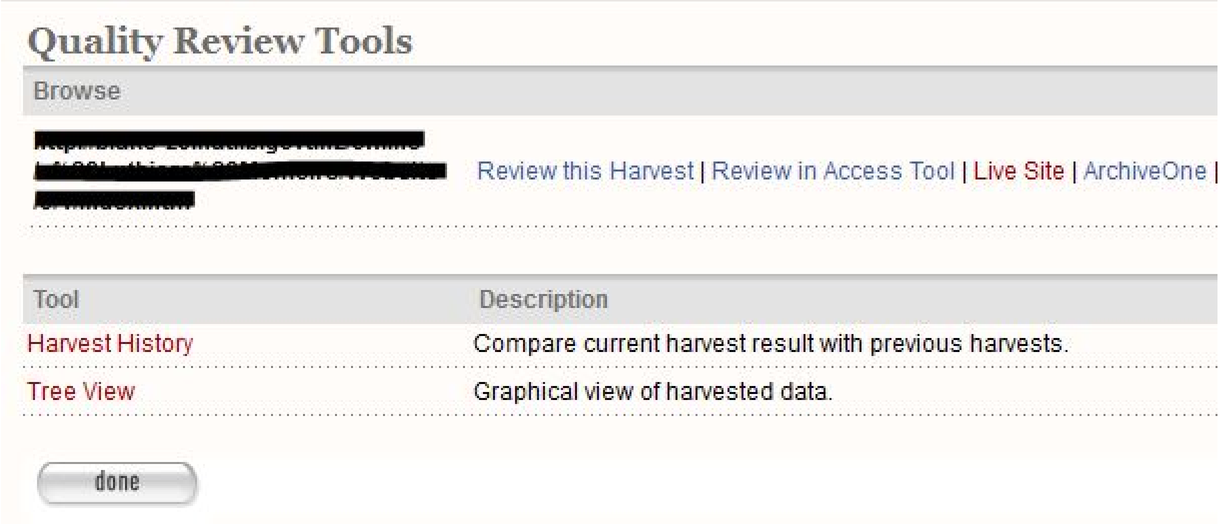


Target Summary

A [redacted] (14090270)

No.	Date	Derived From	User	Notes	State	Action
1	14/09/2015 13:15:59		[redacted]	Original Harvest		Review | Endorse | Reject for Reason:

- Choose the option to 'Review in Access Tool' to view the harvest in Wayback.



Quality Review Tools

Browse

[redacted] Review this Harvest | Review in Access Tool | Live Site | ArchiveOne |

Tool	Description
Harvest History	Compare current harvest result with previous harvests.
Tree View	Graphical view of harvested data.

7.9 More information

The following guides can provide additional information:

- *System Administrator Guide*
- *Developer Guide*
- Troubleshooting Guide
- *FAQ*

CHAPTER 8

Rosetta DPS Configuration Guide

8.1 Introduction

The Web Curator Tool is able to archive harvests to the Rosetta Digital Preservation System (DPS). The National Library of New Zealand currently uses Rosetta DPS for archiving their harvests from WCT.

This guide shows how to deploy and configure an instance of Web Curator Tool to work with Rosetta DPS.

8.1.1 Contents of this document

Following this introduction, the Rosetta DPS Configuration Guide includes the following sections:

- **Wayback Vs OpenWayback** - Covers the Wayback options.
- **Installation** - Covers installing Wayback.
- **Configuration** - Covers configuring Wayback.
- **Wayback as a Review Tool in WCT** - Covers configuring Wayback for use as a review tool in the Web Curator Tool.
- **Testing** - Covers testing the Wayback installation.
- **More information** - Provides some links for more information.

All configuration for this integration is inside 'wct-das.properties'. (This file is located in '/<path to tomcat>/webapps/wct-store/WEB-INF/classes/'.

8.2 Configuration steps

8.2.1 Enable Rosetta DPS archiving

```
# The archive type to use for this installation (one of: fileArchive, omsArchive,␣
↪dpsArchive).
arcDigitalAssetStoreService.archive=dpsArchive
```

8.2.2 Configure the Rosetta Server

```
dpsArchive.pdsUrl=http://xxxserverxxx.xxx.xxx.xx/pds
dpsArchive.ftpHost=xxxftpserverxxx.xxx.xxx.xx
dpsArchive.ftpUserName=<ftp_username>
dpsArchive.ftpPassword=<ftp_password>
dpsArchive.dpsUserInstitution=INS00
dpsArchive.dpsUserName=<rosetta_username>
dpsArchive.dpsUserPassword=<rosetta_password>
dpsArchive.materialFlowId=<rosetta_material_flow_ID>
dpsArchive.producerId=<rosetta_producer_ID>
dpsArchive.depositServerBaseUrl=http://xxxserverxxx.xxx.xxx.xx
dpsArchive.producerWsdlRelativePath=/dpsws/deposit/ProducerWebServices?wsdl
dpsArchive.depositWsdlRelativePath=/dpsws/deposit/DepositWebServices?wsdl
```

8.2.3 Set your access restriction codes

```
#OMS Codes (Rosetta)
dpsArchive.dnx_open_access=1020
dpsArchive.dnx_published_restricted=1021
dpsArchive.dnx_unpublished_restricted_location=1022
dpsArchive.dnx_unpublished_restricted_person=1023
```

8.2.4 Custom deposit form configuration

DPSArchive uses the following two parameters to determine whether a custom deposit form needs to be displayed before submitting an HTML Serial harvest. Configure the following parameters to reflect:

- The name of the agency that would normally harvest/ingest HTML serials
- The Dublin Core *Type* that would represent the target for an HTML serial

If there are more than one value for each of these, separate them using comma.

```
dpsArchive.htmlSerials.agencyNames=Electronic Serials Harvesting
dpsArchive.htmlSerials.targetDCTypes=eSerial,eMonograph
```

URLs that WCT Core would use to display the custom deposit form for each of the target types, separated by comma. A note on the format of this URL:

- If WCT Core and WCT Digital Asset Store are deployed in the same Tomcat instance, use a relative URL.
- If they are deployed in different machines or Tomcat instances, use absolute URL based on WCT DAS' host/port.

```
dpsArchive.htmlSerials.customDepositFormURLs=/wct-store/customDepositForms/
↪rosetta_custom_deposit_form.jsp
```

- The material flow ID for each of the target types, separated by comma. There should be one entry for each target type defined above.

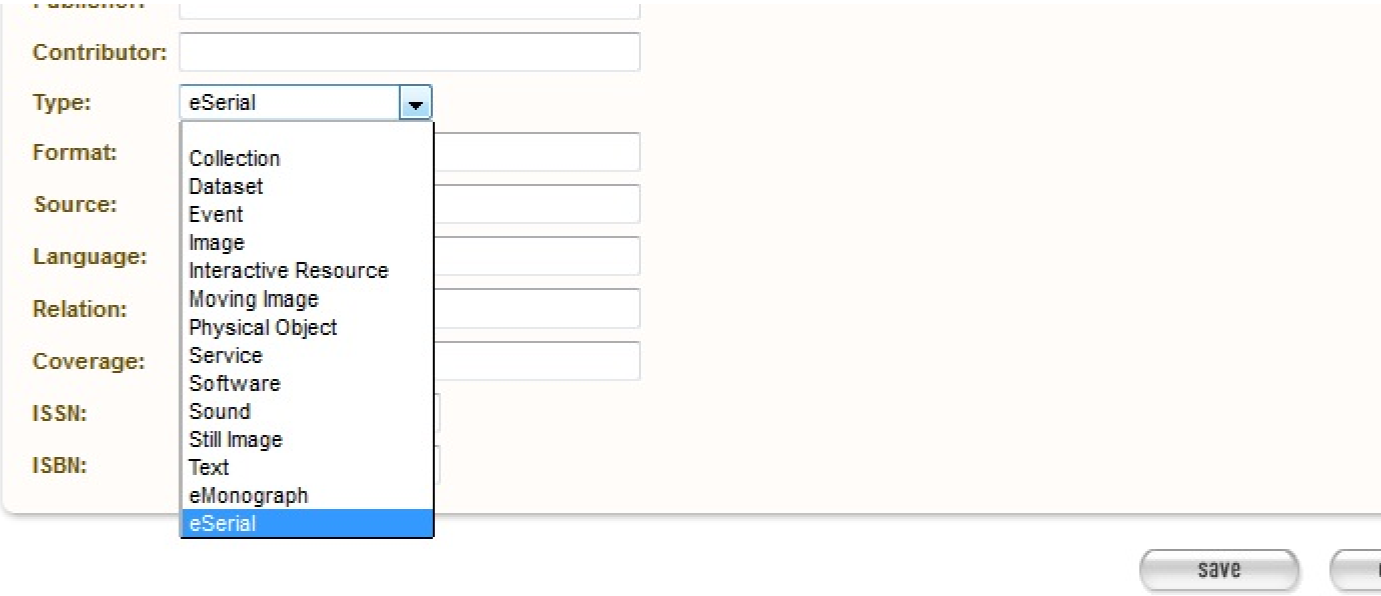
```
dpsArchive.htmlSerials.materialFlowIds=52063,52073
```

- The IE Entity Type for each of the target types, separated by comma. There should be one entry for each target type defined above.

```
dpsArchive.htmlSerials.ieEntityTypes=HTMLSerialIE,HTMLMonographIE
```

8.3 User Interface adjustment

In the event that multiple targetDCTypes are added (as per above), then they need to be made available through the user interface.



Contributor:

Type: eSerial

Format:

Source:

Language:

Relation:

Coverage:

ISSN:

ISBN:

- Configuration for this list of types is inside *wct-core-lists.xml*. (This file is located in */<path to tomcat>/webapps/wct/WEB-INF/classes/*).
- The value should match the *targetDCType* set in *wct-das.properties*.

```
<bean id="dublinCoreTypesList" class="org.webcurator.core.common.WCTTreeSet"␣
↪abstract="false" singleton="true" lazy-init="default" autowire="default" ␣
↪dependency-check="default">
    <constructor-arg index="0" type="java.util.List">
      <list>
        <value></value>
        <value>Collection</value>
        <value>Image</value>
        <value>Interactive Resource</value>
        <value>Moving Image</value>
        <value>Software</value>
```

(continues on next page)

(continued from previous page)

```
        <value>Sound</value>
        <value>Text</value>
        <value>eSerial</value>
        <value>eMonograph</value>
      </list>
    </constructor-arg>
    <constructor-arg index="1" type="int">
      <value>50</value>
    </constructor-arg>
</bean>
```

8.4 More information

The following guides can provide additional information:

- *System Administrator Guide*
- *Developer Guide*
- Troubleshooting Guide
- *FAQ*

CHAPTER 9

Developer Guide

9.1 Introduction

This guide, designed for a Web Curator Tool developer and contributor, covers how to develop and contribute to the Web Curator Tool. The source for both code and documentation can be found at: http://dia-nz.github.io/webcurator/

For information on how to install and setup the Web Curator Tool, see the Web Curator Tool System Administrator Guide. For information on using the Web Curator Tool, see the Web Curator Tool Quick Start Guide and the Web Curator Tool online help.

9.1.1 Contents of this document

Following this introduction, the Web Curator Tool Developer Guide includes the following sections:

- **Contributing** - Covers how to contribute to the project.
- **Basic architecture** - Covers the basic Web Curator Tool architecture.
- **Building** - Covers building the Web Curator Tool from source.
- **Configuration** - Some configuration information.
- **Developer guidelines** - Covers coding practice and development workflow.
- **Future milestones** - Covers plans for future development.

9.2 Contributing

This describes how to contribute to the Web Curator Tool project.

9.2.1 Source Code Repository

Source code for the Web Curator Tool is stored in github at: http://dia-nz.github.io/webcurator/ Contributors to the codebase will require a github account.

9.2.2 Issue tracking

Issues are tracked via Github's issue tracking. The current set of issues can be viewed on the project's *Issues* tab. The issue state (*To do*, *In progress* and *Done*) are also tracked through the *WCT Development* project (go to the *Projects* tab and select *WCT Development* project.

When creating issues please include as much information as possible. The more information that you include, the easier it is for the issue resolver to solve the problem. Useful information includes the following:

Background information

- The version of the Web Curator Tool you are using
- The database type and version (for example, MySql 8.0.12)
- The servlet container type and version (for example, Tomcat 9.0.13)
- The operating system type and version (for example, RHEL 7.6)
- The Java type and version (for example OpenJDK 8u192)
- The version of Heritrix 3 (if applicable)
- The web browser and version (for example Chrome 69.0.3497.100 (64-bit))

Specific issue information

- Mention precisely what went wrong, including the steps you took to get to point where things didn't work as expected. Describing the steps you took can help us reproduce the issue.
- Describe what you expected to have happen.
- Include any relevant log messages.

9.2.3 Pull requests

Pull requests are managed with Github's pull request process. For pull requests, see the *Pull requests* tab in the Github project.

9.2.4 License

All contributions to the Web Curator Tool must be under the Apache 2.0 License, which can be found at: https://www.apache.org/licenses/LICENSE-2.0

9.2.5 Copyright

In general copyright is assumed to belong to either the person who committed a change or the institution employing that person.

Please do not put copyright notices in files.

9.2.6 Major Contributors

Major contributors to the Web Curator Tool are NLNZ (The National Library of New Zealand) (https://natlib.govt.nz/) and KB (Koninklijke Bibliotheek or The National Library of the Netherlands) (https://www.kb.nl). These two institutions currently drive most development. All contributors are welcome. Making your interest in the Web Curator Tool known can help to ensure that the Tool meets your institution's needs.

9.2.7 Development discussion

Slack channels are used to discuss current Web Curator Tool development. The slack channels can be found at https://webcurator.slack.com. The *#development* and *#general* channels are two places to discuss issues.

9.2.8 Getting help

If the documentation isn't sufficiently clear, please use the slack channel *#general* at https://webcurator.slack.com to request assistance. You can also create github issues for specific problems with the tool or its documentation.

9.2.9 We want to know who you are

Part of what makes a community-driven open-source project successful is the relationships between the participants. We want to know who you are. Take the time to announce yourself on the *#community* channel at https://webcurator.slack.com.

9.3 Basic architecture

The following diagram illustrates the basic architecture and its components.

9.3.1 Some important notes

- The Harvest Agents and the Store talk to the Web Curator Tool WebApp, but the WebApp does not talk to the Harvest Agents or Store. This means that Harvest Agents can come and go.
- The Harvest Agents and Store signal to the WebApp that they exist by sending heartbeat messages.
- The Heritrix H1 agent/crawler contains its Heritrix1 crawler.
- The Heritrix H3 crawlers are not aware of their agents. Instead the Heritrix H3 agent tracks the Heritrix3 crawler. Each Heritrix H3 agent runs as a web application (war). Each Heritrix3 crawler (jar) runs in own JVM.
- The WebCurator Store runs as a web application (war).
- The Web Curator Tool WebApp is the only component that communicates with the SQL database.

9.4 Building

9.4.1 Requirements

Build requirements

Building the Web Curator Tool from source requires the following:

- Java 8 (1.8) JDK or above (64bit recommended). Current development assumes using the Oracle JDK, but long-term it may be better to switch to OpenJDK.
- Maven 3+ or later.
- Git (required to clone the project source from Github).

As the artifact targets are Java-based, it should be possible to build the artifacts on either Linux, Solaris or Windows targets.

Development platforms

The following platforms have been used during the development of the Web Curator Tool:

- Sun Solaris 10
- Red Hat Linux EL3.
- Ubuntu GNU/Linux 16.04 LTS and later
- Windows 7 Ultimate, Windows 2000, Windows XP Pro, Windows Server 2003

Web Application Server platforms

The Web Curator Tool currently requires that its *.war* components run in a Web Application Server.

Development has used Tomcat (currently version 8.x) Web Application Server for development. Testing has also taken place using jetty.

Database platforms

The Web Curator Tool requires a backend database for persistent storage.

Development and testing has taken place using MySQL, Postgres and Oracle. See the *System Administrator Guide* for more details. Testing has also used the *H2* database.

9.4.2 Build commands

Installing maven dependencies

While maven generally will pull in dependencies as required from Maven Central, some of the dependencies that different Web Curator Tool components require do not exist in Maven Central. These dependencies have been checked into the codebase and must be installed in the local maven repository so they are available to maven when it builds the different components.

Install the maven dependencies by running from the root project folder: For Windows operating system:

```
install_maven_dependendencies.bat
```

For *nix-based operating systems:

```
install_maven_dependencies.sh
```

Building with unit tests

This can be run from the root project folder, or from a specific subproject folder, such as *wct-core*, *harvest-agent-h1*, *harvest-agent-h3* or *wct-core*.

```
mvn clean install -P<database-type>
```

The *-P<database-type>* parameter is one of *mysql*, *oracle*, *postgres*, as applicable. The *-Ph2* option, if used, is only intended for use with Jetty, and cannot be used to create the .war file with the current version of Hibernate.

The digital asset store (*wct-store*) and harvest agents (*h1-harvest-agent* and *h3-harvest-agent*) do not need a database, so there is no need to specify anything database-related when building or running those specific components.

The artifacts produced by the build (in general these will be *.jar* and *.war* files) will be found in the *target* subfolders of each subproject. The *.war* files are generally copied to the Tomcat *webapps* folder for deployment.

Building and skipping unit tests

This can be run from the root project folder, or from a specific subproject folder, such as *wct-core*, *harvest-agent-h1*, *harvest-agent-h3* or *wct-core*.:

```
mvn clean install -P<database-type> -DskipTests=true
```

Running with jetty

Jetty is an inbuilt lightweight web application server than eliminates the need to run an Web Curator Tool component under Tomcat. It is not production capable but is useful for development. *wct-core*, *harvest-agent-h1*, *harvest-agent-h3* and *wct-store* can all be run using Jetty.

Note that for ‘wct-harvest-agent‘ and ‘wct-store‘, you will see a warning that a profile couldn’t be activated. This is not important.

To run the component under jetty use the following command:

```
mvn jetty:run <command-line-parameters> -P<database-type>
```

Note that the command line parameters will vary based on the different components. If the command line parameter is not specified, a default is used.

For these examples, *core-host* is *localhost*, *core-port* is *8080*, *h1-agent-port* is *8081*, *h3-agent-port* is *8086* and *das-port* is *8082* but any valid port can be used.

***wct-core* under Jetty and H2 first time** *wct-core* can run with a H2 database (as specified with the *Ph2* parameter, which removes the need to run against MySQL, Postgres or Oracle. The first time this is run, the *-Dhbm2ddl.auto=create* creates a new instance of this database.

The H2 database is stored in the user’s home directory (for *nix systems this would be ‘~/DB_WCT.*db‘). Unfortunately, it appears that the *-Dhbm2ddl.auto=create* option doesn’t entirely clear a pre-existing database: In testing we found that the tables *ABSTRACT_TARGET*, *PERMISSION*, *SITE*, *URL_PATTERN* and *URL_PERMISSION_MAPPING* were not cleared. For this reason, before running with the *-Dhbm2ddl.auto=create* option, we recommend that the user deletes the H2 database (if it already exists), by deleting the files *DB_WCT.*.db* found in the user's home directory.

```
mvn jetty:run -Ph2 -Dhbm2ddl.auto=create \
    -Dcore.host="<core-host>" -Dcore.port="<core-port>" -Ddas.port="<das-port>" \
    -Darc.store.dir="<arc-store-directory>" \
    -DarchiveType=fileArchive \
    -Dfile.archive.repository="<file-archive-repository>" \
    -Dfile.archive.files="<file-archive-files>" \
    -Dlog4j.log.dir="<log4j-log-dir>" \
    -Dattach.dir="<attachments-directory>"
```

In this scenario the bootstrap user will be created. Note that the tables are cleared using this command.

***wct-core* under Jetty and H2 subsequent times (when the h2 database already exists)**

```
mvn jetty:run -Ph2 \
    -Dcore.host="<core-host>" -Dcore.port="<core-port>" -Ddas.port="<das-port>" \
    -Darc.store.dir="<arc-store-directory>" \
    -DarchiveType=fileArchive \
    -Dfile.archive.repository="<file-archive-repository>" \
    -Dfile.archive.files="<file-archive-files>" \
    -Dlog4j.log.dir="<log4j-log-dir>" \
    -Dattach.dir="<attachments-directory>"
```

***wct-core* under Jetty and oracle** If using the Oracle database profile, the Oracle driver is required to run Jetty. This driver is not availabe via Maven repositories for licensing reasons - it needs to be downloaded and manually installed.

In general the steps are:

1. Obtain the appropriate driver for your installation (see Oracle documentation).
2. Install it into your maven repository. This is generally done by using a command like:

```
mvn install:install-file -DgroupId=com.oracle -DartifactId=ojdbc14 -Dversion=
↪<version> -Dpackaging=jar -Dfile=<jar-location>
```

3. Change the relevant *pom.xml* to reflect the Oracle jar version in use.
4. Add a dependency in the pom.xml for the jetty plugin (refer to the mysql profile as a reference).

More detailed instructions can be found via internet search engines.

Note also that if you are installing a new database, you will need to create a tablespace called *WCT_DATA* in order for database creation scripts to function as expected. Since this is a database specific configuration, it cannot be defaulted easily.

```
mvn jetty:run \
    -Dcore.host="<core-host>" -Dcore.port="<core-port>" -Ddas.port="<das-port>" \
    -Darc.store.dir="<arc-store-directory>" \
    -DarchiveType=fileArchive \
    -Dfile.archive.repository="<file-archive-repository>" \
    -Dfile.archive.files="<file-archive-files>" \
    -Dlog4j.log.dir="<log4j-log-dir>" \
    -Dattach.dir="<attachments-directory>"
```

***harvest-agent-h1* under Jetty**

```
mvn jetty:run \
    -Dcore.host="<core-host>" -Dcore.port="<core-port" \
    -Dagent.port="<h1-agent-port>" \
    -Ddas.host="<das-host>" -Ddas.port="<das-port>" \
    -Dharvest.tmp.dir="<harvest-temp-directory>" \
    -Dlog4j.log.dir="<log4j-directory>" \
    -Dattach.dir="<attachments-directory>"
```

***harvest-agent-h3* under Jetty** *harvest-agent-h3* requires a separate instance of Heritrix3 to run. See the *System Administrator Guide* for details on how to setup and run Heritrix3.

There may be conflicts with the JMX port of other components. You can change the port used by editing the *build/jetty/jetty-jmx.xml* and changing the port from *localhost:9004* to another unused port.

```
mvn jetty:run \
    -Dcore.host="<core-host>" -Dcore.port="<core-port" \
    -Dagent.port="<h3-agent-port>" \
    -Ddas.host="<das-host>" -Ddas.port="<das-port>" \
    -Dharvest.tmp.dir="<harvest-temp-directory>" \
    -Dlog4j.log.dir="<log4j-directory>" \
    -Dattach.dir="<attachments-directory>"
```

***wct-store* under Jetty**

```
mvn jetty:run \
    -Dcore.host="<core-host>" -Dcore.port="<core-port>" -Ddas.port="<das-port>" \
    -Darc.store.dir="<arc-store-directory>" \
    -DarchiveType=fileArchive \
    -Dfile.archive.repository="<file-archive-repository>" \
    -Dfile.archive.files="<file-archive-files>" \
    -Dlog4j.log.dir="<log4j-log-directory>" \
    -Dattach.dir="<attachments-directory>"
```

XDoclet

XDoclet is still used to generate *hibernate.cfg.xml* and the *.hbm.xml* files. This is configured via the *xdoclet-maven-plugin* and the antrun plugin.

Future development that includes a Hibernate upgrade will remove the dependency on XDoclet.

9.5 Configuration

9.5.1 Configuration details

The *System Administrator Guide* contains detailed information about configuring the Web Curator Tool.

The configuration files are generally found in the *build* subfolder of each subproject.

You may need to change various configuration settings in one of these files to make them work for your specific environment. The MySQL configuration should require minimal/no changes if using the default installations. The H2 configuration should require no changes to start.

9.5.2 Maven filtering

Maven has a feature called *filtering* where it tries to replace placeholders like *${core.port}* with a property value that has been configured. This is an optional feature which is off by default, however WCT makes use of it for some of the build resources. Any *<resource>* with a *<filtering>* value of *true* is filtered, and the properties are supplied in two places: the *<properties>* tag, and via the *properties-maven-plugin*. These properties are also used to resolve these placeholders inside the *pom.xml* itself, e.g. *${databaseType}*.

9.6 Developer Guidelines

9.6.1 Coding practice

- We assume common good coding practices. Consider following the principles outlined in Robert C. Martin's book *Clean Code* (https://www.oreilly.com/library/view/clean-code/9780136083238/).
- New functionality changes have a reasonable set of unit tests included. This can be enforced through minimal code coverage tests as part of the build process.
- Code contains robust instrumentation, which means extensive and detailed logging about the state of operations at significant processing points.

9.6.2 Code style

While coding style can be idiosyncratic and personal, consider following established coding styles enforced through Checkstyle. This ensures that all code has a similar look and feel while also preventing wasted effort in code reviews and pull requests discussing formatting. Candidates for a consistent coding style include:

- Google Java Style Guide - https://google.github.io/styleguide/javaguide.html which is a subset of the Google style guide https://github.com/google/styleguide
- OpenJDK Java Style Guide - http://cr.openjdk.java.net/~alundblad/styleguide/index-v6.html
- Spring framework code style - https://github.com/spring-projects/spring-framework/wiki/Code-Style
- 47deg coding guide - https://github.com/47deg/coding-guidelines/tree/master/java/spring
- Oracle's coding conventions - https://www.oracle.com/technetwork/java/codeconventions-150003.pdf Note that this guide is significantly out of date and is only included here for historical purposes.

9.6.3 Definition of Done

Code is considered done and can be merged into the master branch when the following conditions have been met:

- The requirements driving the change have been satisfied by the change.
- The code builds without errors.
- All unit tests pass.
- Unit test code coverage remains the same or is increasing.
- Functional tests have all passed.
- Non functional requirements met.
- Significant user journeys all work.
- Code and other changes have been peer reviewed and approved.

- New code has instrumentation (logging points) that conveys accurate and helpful information about the state of the application.
- The documentation has been updated to reflect changes in functionality. Some documents that could be updated include: - The *Release Notes release-notes.rst*, especially for new features. - If there are any database changes, update the *Data Dictionary data-dictionary.rst*. - If there are changes related to installing and running the WCT, update the *System Administrator Guide system-administrator-guide.rst*. - If there are any changes that would require steps to upgrade from a previous version, update the *Upgrade Guide upgrade-guide.rst*. - If there is any helpful advice regarding troubleshooting, update the *Troubleshooting Guide troubleshooting-guide.rst*. - If there is helpful information that can be include in the FAQ, update the *FAQ faq.rst*.
- The Product Owner accepts the changes.

9.6.4 Semantic versioning

Use semantic versioning as described in https://semver.org/ . This means having a version number composed of major, minor and patch versions. For current development this means changing the maven *pom.xml* associated with each build artifact and tagging the associated git commit with the version.

TODO Make the steps to change version number is maven and git more explicit, perhaps as part of the **Git workflow**.

9.7 Git Workflow

This workflow is a hybrid of several popular git workflows (Github Flow, Atlassian Simple Git, Cactus Model, Stable Mainline Model), designed to fit the needs of the NLNZ and KB collaborative development of WCT. It will use a shared repository model via Github using the https://github.com/DIA-NZ/webcurator repository.

9.7.1 Commit Messages

Prefix commit messages with a ticket number (when applicable). This information comes in handy when reviewing git history, or when cherry-picking individual commits (e.g. when cherry-picking a bug-fix commit from master into a release branch, the resulting history will be more informative).

TODO Consider more detail in the commit message, limiting line length.

Commit message example

```
D1.1: Add a unit test for dynamic reflow
```

9.7.2 Master Branch

The master branch is the default development branch for this project. For most purposes, the master branch is considered stable. In other words, if you check out the master branch you can expect that:

- It builds on all supported platforms/targets.
- All unit tests pass (as well as static tests, linter checks and the like).
- A "standard run" of the software works (WCT should start up).

However, the master branch might not pass a comprehensive QA test at all times.

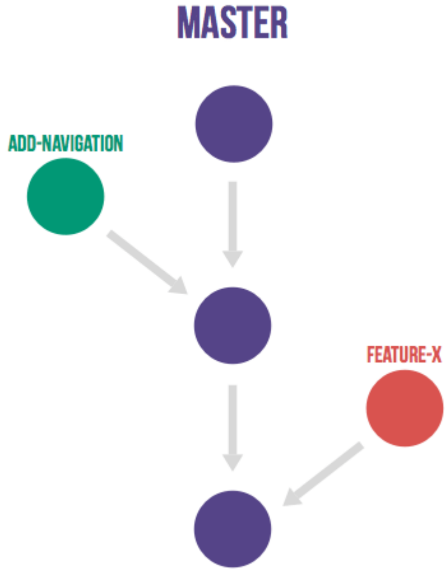
9.7.3 Feature Development

Feature branch purpose

All development is done in dedicated (relatively short lived) feature branches. This is where most of the action takes place, including:

- Feature development.
- Code review.
- Integration testing.

A feature branch branches off from master, and once development is finished and all the integration criteria have been met, code review completed, it is merged back to the master branch using a pull request.



Feature branch naming

This project will use the following convention for naming feature branches:

```
“feature/<ticket>_description_separated_by_underscores”
```

where ticket is a reference to the corresponding ticket in the project issue tracker (or work plan), and description is a very short description (up to five words or so) of the purpose of the branch.

Feature branch naming example:

```
feature/D1.1_new_harvestagent_h3_impl
```

If a feature branch is running for an extended period of time, consider breaking the issue/story into smaller components that can be integrated more frequently with the master branch.

Updating Feature Branches

To keep feature branches up to date with changes in the master branch, it is a good idea to merge regularly from master to minimize merge conflicts later on when it is time for a feature to be merged back into master.

While rebasing is considered common practice in keeping feature branches up to date, in most situations it won't be appropriate in this project due to sharing remote branches for pull requests and code review/testing. Rebasing rewrites the history of a branch and has potential for history breakage when sharing branches.

There are some distinct advantages for rebasing, but it's not recommended given the current nature of a large codebase in a single repository. When the codebase gets split into multiple repositories based on functional components the use of rebasing might be more appropriate.

To update feature branches use merging.

Checking out a branch example:

```
git checkout feature_branch
git pull origin master
```

Reasons for using 'Always Merge' convention

- Pull Requests won't contain rebased commits from master that have already been reviewed. You will just see the changes relating to the feature branch.
- Merging changes from master, 'rework' commits, should mean you will only need to fix merge conflicts once. Whereas merge conflicts need to be resolved every time a rebase is done.
- Rebasing can be dangerous when used on shared remote branches, as the history of the branch is being rewritten.
- No need to worry about using force push for a branch that has been rebased.
- Rebasing is generally considered a complex and advanced feature of git. In order to make it easier for the community to engage with Web Curator Tool developement, it would be wise to keep the project workflow as simple as possible.

9.7.4 Code Review and Pull Requests

Pull Requests are to be used to initiate code reviews and discussions about the code implementation in a dedicated branch that does not interfere with the main development branch. This review/testing can done at any stage in the development of that branch. As a rule, all feature branches must be peer reviewed via Github before being merged into the master branch.

Sharing a feature branch remotely

1. Ensure your feature branch is up to date with latest changes from master.
2. Push the latest commit from your feature branch to the shared github repository.
3. Fetch remote feature branch into local repository.

Initiating a code review via Github

1. Ensure your feature branch is up to date with latest changes from master.
2. Push the latest commit from your feature branch to the shared github repository.
3. Navigate to that branch in Github, and open a Pull Request.
4. Use WIP if not ready to be merged into master.
5. Use assigning and mentions to ensure the right people are notified of the Pull Request.

After the initial push of a feature branch you can keep pushing updates to the remote branch multiple times throughout. This can happen in response to feedback, or because you're not done with the development of the feature.

9.7.5 Merging into Master

Merging feature branches into master will use the no fast forward method. This forces the creation of merge commits to preserve the notion of the feature branches in the git history, and also makes it easier to revert a merge if necessary.

TODO Shouldn't all merges to Master be done via Github pull request? In fact, the Github master branch should be locked down so that merges are done ONLY by pull request.

```
git checkout master
git merge --no-ff branch
```

Example of merging with fast forward:

```
git merge --no-ff feature/DX.Y_desc
```

If merging a major feature that includes a large number of commits then add the *–log* flag to the merge command to include a brief description of the commits that were merged.

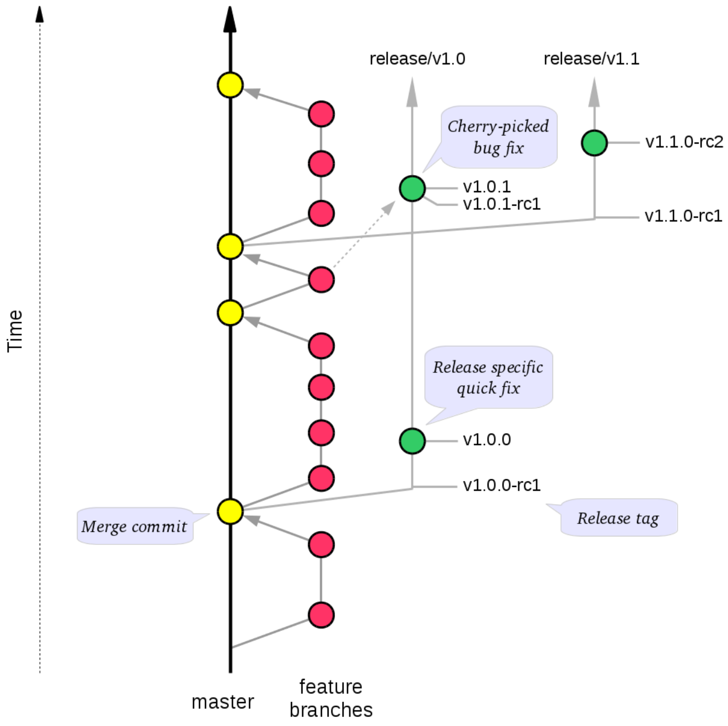
Example of merging with log flag:

```
git merge --no-ff --log feature/DX.Y_desc
```

9.7.6 Releases

Release branch criteria

This project will use release branches and tags to designate releases. Once it has been decided what version number to use and what commit to base a release on, a new release branch is created from the selected commit, and it is never merged back into master.



Changes to the release branch

After a release branch is announced, only serious bug fixes are included in the release branch. If possible these bug fixes are first merged into master and then cherry-picked into the release branch. This way you can't forget to cherry-pick them into master and encounter the same bug on subsequent releases.

Release branch naming

Given a regular major.minor.patch version numbering scheme (e.g. semantic versioning), a release branch should be named *release/vX.Y*, where *X* is the major version number and *Y* is the minor version number.

Example of release branch naming:

```
release/v1.3
```

Git release tags

In addition to release branches, release tags are created for each actual release (this may include release candidates that are intended for QA or beta testing, as well as public releases). The release tags are made in the corresponding release branch.

The commit that represents a specific release is tagged with a tag named *vX.Y.Z*, optionally suffixed with a textual identifier, such as *-alpha*, *beta*, *-rc1*.

Example of release tag:

```
v1.3.2-rc1
```

Patch versions

The first release version from the *vX.Y* release branch, is tagged with the patch version *0*, eg. *vX.Y.0*. Every time a bug-fix is included in a release branch the patch version is raised (to comply with Semantic Versioning) by setting a new tag.

When no more bugs are found, tag the tip of the release branch with *vX.Y.Z* (it is no longer a release candidate), and if necessary make a final build (e.g. to get the release number correct in the release package etc).

9.7.7 Continuous Integration (placeholder)

TODO Write notes and instructions for continuous integration.

9.8 Future milestones

Future milestones are divided into several different phases, some of which can be pursued independently.

9.8.1 Audit usage

Future development work may involve restructuring the application code and applying technical upgrades to underlying frameworks. The technical direction of code changes also relies on ensuring that the Web Curator Tool meets the needs of its current and potential future users. Application functionality needs verification throughout all steps of restructuring, uplift and functional enhancement. For this reason, developers need to understand and duplicate current usage by:

1. Understanding who uses the Web Curator Tool and how they use it.
2. Provide a set of key user journeys. These user journeys cover all essential functionality in how the tool is used.
3. Write unit and/or integration tests that cover those essential user journeys. These tests are used to ensure that all essential functionality remains through all development changes.

9.8.2 Containerization and continuous integration

Containerization

Containerization ensures that each Web Curator Tool can run in its own container connected to other containers. (TODO Describe the advantages of containerization and what it means for the WCT).

Repository split

Splitting the single Web Curator Tool into multiple repositories means that the each component can be developed, built, versioned and released independently from the other components. This provides the advantage of decoupling the components. Decoupling is predicated on reliable interfaces connecting each component.

Continuous integration through build and deploy pipeline

A preconfigured build and deploy pipeline (or pipeline template) allows developers to quickly build and test changes and put new releases into production.

Ease of installation

Part of the reason to move to a containerisation approach with a build and deploy pipeline is to make it easier for users to easily build, stand up and run the Web Curator Tool in a production environment. It also means that component upgrades are much easier to roll out by component (so one component can receive an upgrade/code change without requiring all components be changed).

9.8.3 Quality assurance improvements

In addition to providing a testable set of user journeys and an easy-to-use build and deploy pipeline, additional changes that ensure code quality, including:

- More comprehensive logging at all API points.
- Better enforcement of coding quality and standards through build-time enforcement using such things as PMD static code analysis (https://pmd.github.io/), Jacoco code coverage (https://www.eclemma.org/jacoco/), FindBugs (http://findbugs.sourceforge.net/), Checkstyle for coding style (http://checkstyle.sourceforge.net/), SonarQube for code quality (https://www.sonarqube.org/) and others.
- Switch to Test-Driven Development.
- Consistently applied coding and development standards.

9.8.4 Component based REST API

APIs ensure that the different components can talk to each other through standard interfaces. Currently communication between components is handled via SOAP interfaces. The technical uplift would move the API interfaces to REST. The API would allow for decoupling of the components and more flexibility in how the Web Curator Tool is structured for use in production. Several potential API candidates exist:

1. Agent API - A generic wrapper supporting different crawlers, such as Heritrix1, Heritrix3 and other potential crawlers, such as WebRecorder. Re-develop WCT Core and Harvest Agent to be crawler agnostic, allowing other crawl tools to be utilised by WCT. Harvest Agent pooling/grouping also required to allocate scheduled Targets to different crawl tools.
2. Workflow API - This would separate out the workflow into a separate component to allow easier integration with other systems.
3. Administration API - For management of users, roles and other administrative components.
4. Configuration API - For easier management of configuration so that run time values are contained in a single location instead of being spread across properties files, xml files and hard-coded in the codebase.

9.8.5 Technical uplift

Upgrade the frameworks and technologies that underpin the Web Curator Tool. This technical shift is aimed at ensuring that the technologies used are supported and effective.

Uplift components

Technology	Reasoning	Alternatives
Java 11 (OpenJDK)	Java 11 is the latest version. Containerization helps limit exposure of co-located applications.	Possibly Java 8. But long-term support ends in 2023.
Spring 5.x	Latest version.	None.
Spring boot	Simplify deployment. Light- weight and more compatible with microservice approach.	Deploy as war.
REST (API)	More universally supported and compatible with micro- service approach.	
jQuery 3.3.1	Use the latest version of jQuery, with its security fixes, speed improvements and modern browser support.	Keep using jQuery 1.7.2 which dates from 2012.
Quartz 2.3.0	Latest version.	Spring scheduler, which is a bit simpler.
GUI framework	Major upgrade. Decoupled from back-end services via REST API (REST API allows for custom clients). No specific technology has been proposed.	None. Struts 1.2.7 is long-unsupported, difficult to use and maintain.
JPA (Hibernate)	Standard way of interfacing with database. This would include an upgrade to latest Hibernate (currently 5.3.x).	Straight JDBC or MyBatis, which allows for writing SQL instead of a generic wrapper like Hibernate.
Microservices	Decouple application into focused components.	Keep as monolith.
Gradle builds	A more flexible build tool that makes build and deploy pipelines easier to write.	Keep using maven.

Additional uplift notes

- Java 11 - OpenJDK has moved from version 8 to 11, so it makes sense to make the same leap. If the Web Curator Tool is a monolith, this could cause issues because it means that all co-located applications (as in, those applications running on the same machine) would need to upgrade as well. However, running the Web Curator Tool components in containers means that the container itself would have the Java version required.
- Spring boot - Spring boot applications are deployed as Java jars. This can simplify application deployment.
- REST (API) - In order to maintain a working Web Curator Tool throughout the upgrade process, the REST API would be incorporated into the existing codebase as upgraded component by component.
- GUI framework - Exposing all Web Curator Tool functionality through REST API services allows for different GUI frameworks to run against the same API. Some research is necessary to determine a suitable technology, one that is well supported, easy to work with and having a large user base.
- JPA (Hibernate) - Hibernate tends to obscure the underlying SQL. It may be more effective to write all SQL queries in ANSI SQL so they run across all database flavours without change. Using straight JDBC or MyBatis could make development and maintenance much easier to understand, allowing less experienced (and not Hibernate savvy) developers participate. There doesn't seem to be an inherent requirement for using SQL, so consider whether NoSQL might work better.

CHAPTER 10

Data Dictionary

10.1 Additional TODO

- Do a git comparison between version 1.6.2 and version 2.0.0 and document all changes between the two versions.

10.2 Introduction

This guide, designed for a Web Curator Tool developer and contributor, explains and documents the database for the Web Curator Tool. The source for both code and documentation for the Web Curator Tool can be found at: http://dia-nz.github.io/webcurator/

For information on how to install and setup the Web Curator Tool, see the Web Curator Tool System Administrator Guide. For information about developing and contributing to the Web Curator Tool, see the Developer Guide. For information on using the Web Curator Tool, see the Web Curator Tool Quick Start Guide and the Web Curator Tool online help.

10.2.1 Contents of this document

Following this introduction, the Web Curator Tool Developer Guide includes the following sections:

- **Changes** - Covers changes between different versioned releases.
- **Data model diagram** - provides a diagram of the WCT data model.
- **Data descriptions** - Data descriptions for the data fields.
- **Database descriptions** - Descriptions for the tables and their fields.
- **Generating primary keys** - How to generate primary keys.

10.3 Changes

10.3.1 Changes since 2.0.0

- Placeholder for changes since version 2.0.0. This list should be updated after every feature/bug fix is merged into the master branch.

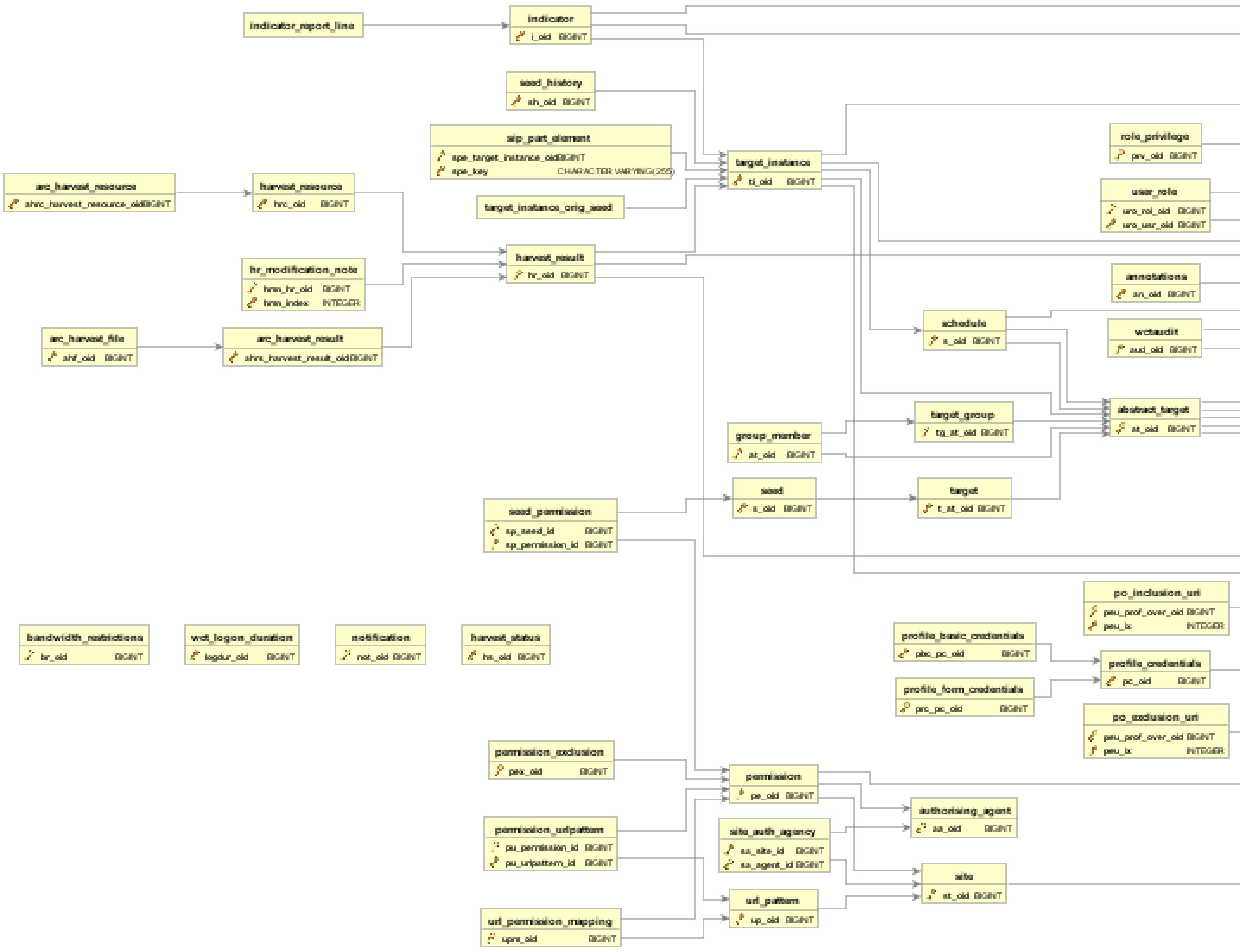
10.3.2 1.6.2 to 2.0.0

- Placeholder for changes between version 1.6.2 and 2.0.0.

10.4 Data model diagram

The data model diagram shows the relationships between the different tables.

Note that this diagram cannot be updated since we don't have the original source file. Any significant updates to the tables and/or their relationships should result in the commissioning of a new diagram.



10.5 Data descriptions

10.5.1 Overview

This section describes the tables in the WCT database.

10.5.2 Field types

Field types in this document are indicative only, and may depend on the implementation.

The types used are:

Boolean A Boolean value (true or false, or 0 or 1, depending on implementation).

Text A free text field.

Constrained text A text field whose contents are constrained to a limited set of values by the application (see *Constrained text fields* below).

Timestamp A timestamp encoding a date and time.

Primary key A unique internal identifier (see *Generating primary keys*).

Secondary key A key from another table.

Float A floating-point number.

Number An integer number.

10.5.3 Constrained text fields

Some tables have fields that are constrained to a fixed set of values.

These fields will be implemented in the database as Text fields, but will appear to users as enumerations (usually in a drop-down menu).

In most cases, the set of possible values can be set in a configuration file (to support different requirements at different institutions).

In each case, a single value can be assigned.

10.6 Database descriptions

10.6.1 Targets, Groups and Schedules

ABSTRACT_TARGET

The *ABSTRACT_TARGET* table is used to store information that is common to both Targets and Groups.

The table is needed because the WCT can be instructed to “harvest” an entire Group at once, as though it were a Target. This means that the *ABSTRACT_TARGET* is used to contain or manage all profile and scheduling information.

Name	Type	Description
AT_OID	Primary key	
AT_DESC	Text	An internal description of the Target or Group.
AT_NAME	Text	The name of the Target or Group.
AT_OWNER_ID	Foreign key	The owner of the Target or Group.
AT_PROF_OVERRIDE_OID	Foreign key	The key of the profile override information for this Target or Group.
AT_STATE	Integer	The state of the Target or Group. Values will be different for Targets than for Groups. Target values correspond to: Pending, Nominated, Rejected, Approved, Completed, Cancelled, Reinstated.
AT_PROFILE_ID	Foreign key	Reference to the profile information for this Target.
AT_OBJECT_TYPE	Integer	Identifies whether this is a Target (1) or a Group (0).
AT_CREATION_DATE	Timestamp	The date and time the *ABSTRACT_TARGET* was created.
AT_REFERENCE	Text	An external reference number (e.g. catalogue number).
AT_PROFILE_NOTE	Text	Records notable aspects of the site that relate to the choice of harvest profile and overrides.
AT_DUBLIN_CORE_OID	Foreign key	Reference to the Dublin Core metadata for this Target.
AT_ACCESS_ZONE	Integer	Access Zone (enumerated field): 0 – Public (default), 1 – On Site, 2 - Restricted.
AT_DISPLAY_TARGET	Boolean	Display this Target.
AT_DISPLAY_NOTE	Text	Records an explanation of the Access Zone and Display Target choices.
AT_DISPLAY_CHG_REASON	Text	Records the reason the *AT_DISPLAY_TARGET* Boolean was last changed.
AT_RR_OID	Foreign key	Reference to the rejection reason for this Target.

TARGET

TARGET contains information specific to Target objects.

Each Target is based on an *ABSTRACT_TARGET*, and takes its primary key from the *ABSTRACT_TARGET* primary key.

Name	Type	Description
***T_AT_OID* | Primary key** (Foreign key)		Reference to *ABSTRACT_TARGET* corresponding to the Target.
T_RUN_ON_APPROVAL	Boolean	If true, then an additional Target Instance will be scheduled to begin one minute after the Target state is set to Approved.
T_EVALUATION_NOTE	Text	Records notable aspects of the site that relate to its evaluation.
T_SELECTION_DATE	Timestamp	The date the Target was formally selected. This should be set automatically to the date and time the Target state first changed to Approved.
T_SELECTION_NOTE	Text	Records information relating to the selection process, in particular reasons for the selection decision.
T_SELECTION_TYPE	Constrained text	Records the type of schedule that has been applied to the site. Example values: one-off, ad hoc, regular.
T_HARVEST_TYPE	Constrained	Records type of selective harvest approach has been applied to the site. Example values: subject, event, theme.
T_USE_AQA	Boolean	Records whether TIs derived from this Target should be marked for inclusion in the automated quality assurance (AQA) post harvest processes.
T_ALLOW_OPTIMIZE	Boolean	Flag to indicate whether harvest optimization is permitted for this target's harvests.

SEED

SEED contains the set of seed URLs corresponding to a Target.

Name	Type	Description
S_OID	Primary key	
S_SEED	URL	The seed URL.
S_TARGET_ID	Foreign Key	The key of the Target the key belongs to.
S_PRIMARY	Boolean	Records whether the URL is marked as a primary URL in the user interface.

SEED_HISTORY

SEED_HISTORY contains the set of seed URLs corresponding to a Target Instance when harvested. Population of this table can be turned off in *wct_core.xml*. Once written the content is not used again by WCT.

Name	Type	Description
SH_OID	Primary key	
SH_TI_OID	Foreign Key	The key of the Target Instance the key belongs to.
SH_SEED	URL	The seed URL.
SH_PRIMARY	Boolean	Records whether the URL is marked as a primary URL in the user interface.

TARGET_GROUP

TARGET_GROUP contains information specific to Group objects.

Each Group is based on an *ABSTRACT_TARGET*, and takes its primary key from the *ABSTRACT_TARGET* primary key.

Groups can usually act as logical groupings that indicate that a set of Targets share some property. For example, a set of Targets in the *Elections 2005* Group might all be relevant to a particular general election. They can also act as functional groupings that simplify the management of Targets by allowing all the Targets in a Group to have a crawl scheduled for specific time. This means they share much of the functionality of a Target (specifically, the ability to schedule a harvest, with all the profile and scheduling data required).

Group membership is recorded in the *GROUP_MEMBER* table.

Name	Type	Description
TG_AT_OID	Primary key (Foreign key)	Reference to *ABSTRACT_TARGET* corresponding to the Group.
TG_SIP_TYPE	Boolean	Controls whether the members are crawled as separate jobs or as a single combined job when the Group is crawled.
TG_START_DATE	Date	The date on which the Group becomes relevant to its members.
TG_END_DATE	Date	The date after which the Group ceases to be relevant to its members.
TG_OWNERSHIP_METADATA	Text	Additional information describing the ownership of a Group, particularly for Groups that have multiple owners.
TG_TYPE	Constrained text	Records the type of Group. Example values: collection, subject, thematic, event, functional.

GROUP_MEMBER

GROUP_MEMBER records Group membership information.

Name	Type	Description
AT_OID	Primary key (Foreign key)	
GM_CHILD_ID	Foreign key	The key of the child (member) Target or Group.
GM_PARENT_ID	Foreign key	The key of the parent (containing) Group.

SCHEDULE

A SCHEDULE contains information about the times that a harvest will be run.

Name	Type	Description
S_OID	Primary key	
S_CRON	Text	The cron pattern this schedule is based on.
S_START	Timestamp	The date the harvests are to commence.
S_END	Timestamp	The date the harvests are to end.
S_ABSTRACT_TARGET_ID	Foreign key	ID of the AbstractTarget to which this schedule belongs.
S_TYPE		The type of the schedule. This is 0 for a custom schedule, or the ID number of a SchedulePattern from the wct-core.xml.
S_OWNER_OID	Foreign key	The key of the User who is the owner of this schedule.
S_NEXT_SCHEDULE_TIME	Timestamp	The date of the next harvest initiated by this schedule.
S_ABSTRACT_TARGET_ID		The key of the Target or Group this schedule is part of.
S_LAST_PROCESSED_DATE	Timestamp	The date that the background batch scheduling processing last processed this record – used to optimise batch processing.

10.6.2 Target Instances and Harvest Results

TARGET_INSTANCE

TARGET_INSTANCE contains information specific to the Target Instances. Target Instances represent the harvests that have occurred, are occurring, or will occur for a Target

Name	Type	Description
TI_OID	Primary key	
TI_VERSION	Number	Internal version number for optimistic locking.
TI_SCHEDULE_ID	Foreign key	The key of the schedule that initiated this harvest.
TI_TARGET_ID	Foreign key	The key of the *ABSTRACT_TARGET* that this Target Instance is derived from.
TI_PRIORITY	Number	0 = High Priority; 100 = Normal Priority; 1000 = Low priority.
TI_SCHEDULED_TIME	Timestamp	The date and time the harvest is (or was) scheduled to begin.
TI_STATE		The current state of the Target Instance. Values correspond to: Scheduled, Running, Paused, Aborted, Harvested, Rejected, Endorsed, Archived.
TI_BANDWIDTH_PERCENT		The proportion of the total available bandwidth that has been manually assigned to this crawl job (empty if the default bandwidth allocation has not been overridden).
TI_ALLOCATED_BANDWIDTH	Number	The actual amount of bandwidth assigned in Kilobytes per second.
TI_START_TIME	Timestamp	For harvests that have started, the date and time the harvest actually did begin.
TI_OWNER_ID	Foreign key	The key of the User who is the owner of this schedule.
TI_DISPLAY_ORDER	Number	A number to assist with the ordering of results in the Target Instance search results screen. This number is tied to the state of the target instance.
TI_PROF_OVERRIDE_OID	Foreign key	The key of the profile override information for this harvest.
TI_PURGED	Boolean	True if the Harvest Results have been purged from the Digital Asset Store.
TI_ARCHIVE_ID	Text	The ID returned by the Archive when the Harvest Result is “Submitted to Archive”, if any.
TI_REFERENCE	Text	Duplicate of the *TI_ARCHIVE_ID* field.
TI_HARVEST_SERVER	Text	The name of the harvest agent that ran this Target Instance.
TI_DISPLAY_TARGET_INSTANCE	Boolean	Display this Target Instance.
TI_DISPLAY_NOTE	Text	Records an explanation of the Display Target Instance choice.
TI_FLAGGED	Boolean	Flag this target instance.
TI_PROFILE_ID	Number	If this target instance is in a running state or later, this is the ID of the locked profile used to run the target instance.
TI_ARCHIVED_TIME	Timestamp	The time that this target instance was archived or rejected.
TI_FIRST_FROM_TARGET	Boolean	Is this the first TI created from a particular Target?
TI_DISPLAY_CHG_REASON	Text	The reason the *TI_DISPLAY_TARGET_INSTANCE* Boolean was last changed.
TI_USE_AQA	Boolean	Records whether the TI is marked for inclusion in the automated quality assurance (AQA) post harvest processes.

HARVEST_RESULT

A *HARVEST_RESULT* is a set of files that represent the result of a harvest of a Target Instance. Note there can be several harvest results for each Target Instance (the first created by the crawler, the rest by QR tools).

Name	Type	Description
HR_OID	Primary key	
HR_HARVEST_NO	Number	The sequence number of the result. Harvest Result #1 is always the original harvest. Harvest Result #2 can be created through the prune tool.
HR_TARGET_INSTANCE_ID	Foreign key	The key of the Target Instance this harvest result belongs to.
HR_PROVENANCE_NOTE		The provenance note of this Harvest Result.
HR_CREATED_DATE	Timestamp	The date the harvest result was created.
HR_CREATED_BY_ID	Foreign key	The key of the User who created the Harvest Result.
HR_STATE	Number	The endorsement state of the Harvest Result. Values correspond to: 1 = Endorsed; 2 = Rejected
HR_INDEX	Number	An internal number for list management, this is mandatory for a Hibernate List.
HR_DERIVED_FROM	Number	The list index of the harvest result that this harvest result is derived from. This is used in the case of a pruned harvest result.
HR_RR_OID	Foreign key	Reference to the rejection reason for this Harvest Result.

ARC_HARVEST_RESULT

ARC_HARVEST_RESULT associates each ARC file (*ARC_HARVEST_FILE*) with a Harvest Result (HARVEST_RESULT). This allows for flexibility in the future, despite having no data at present.

Name	Type	Description
AHRS_HARVEST_RESULT_OID	Primary key	
HR_MODIFICATION_NOTE		This table holds a record of the modifications made to a harvest through the Prune Tool.
HMN_HR_OID	Foreign key	The key of the Harvest Result that this belongs to.
HMN_INDEX	Number	The list index number (used to keep the order of the list).
HMN_NOTE	Text	The text describing the modification.

ARC_HARVEST_FILE

ARC_HARVEST_FILE contains information describing a single ARC file that is part of an *ARC_HARVEST_RESULT*.

Name	Type	Description
AHF_OID	Primary key	
AHF_COMPRESSED	Boolean	Specifies whether the ARC file is compressed.
AHF_NAME	Text	The ARC file name.
AHF_ARC_HARVEST_RESULT_ID	Foreign key	The key of the *ARC_HARVEST_RESULT* this file belongs to.

HARVEST_RESOURCE

HARVEST_RESOURCE contains information about each resource harvested.

Name	Type	Description
HRC_OID	Primary key	
HRC_LENGTH	Number	The length of the resource in bytes.
HRC_NAME	Text	The URI of the resource.
HRC_HARVEST_RESULT_OID	Foreign key	The key of the *HARVEST_RESULT* this file belongs to.
HRC_STATUS_CODE	Number	The HTTP status code of the resource (e.g. 200 = OK, 500 = Internal Server Error, etc.).

ARC_HARVEST_RESOURCE

ARC_HARVEST_RESOURCE contains information about a harvested resource that is particular to the ARC format.

Name	Type	Description
AHRC_HARVEST_RESOURCE_OID	Primary key	
AHRC_RESOURCE_LENGTH	Number	Not used – we currently rely on the HarvestResource's length attribute.
AHRC_RESOURCE_OFFSET	Number	The offset of this resource in the ARC file.
AHRC_ARC_FILE_NAME	Text	The ARC file that contains this resource.
AHRC_COMPRESSED_YN	Boolean	True if the ARC file is compressed; otherwise false.
SIP_PART_ELEMENT		The SIP_PART_ELEMENT table is used internally to store parts of the SIP that must be created when a target instance's harvest is started. This ensures that the details in the SIP remain consistent, even if the target instance's data is changed between harvest and archive.
SPE_KEY	Text	A key indicating what part of the SIP this row represents.
SPE_TARGET_INSTANCE_OID	Foreign Key	The key of the Target Instance to which this belongs.
SPE_VALUE	Text / CLOB	The value of this part of the SIP.
TARGET_INSTANCE_ORIG_SEED		This table holds the seeds of a target instance at the time the harvest was started. This is used internally to the WCT to ensure that the seeds stated in the SIP represent those at the time of the harvest, rather than those at the time of archiving (for example, if the seeds of the Target were changed after the harvest had started).
TIOS_TI_OID	Foreign key	The key of the Target Instance to which this belongs.
TIOS_SEED	Text	The seed at the time of harvest.

REJECTION_REASON

This table holds the reason for rejection that may be assigned to a Target or Harvest Result when it is rejected by the user. An administration page within WCT allows system administrators to set these up on a per agency basis.

Name	Type	Description
RR_OID	Primary key	
RR_NAME	Text	A description of the reason for rejection.
RR_AVAILABLE_FOR_TARGET	Boolean	Should this reason be applicable to Targets?
RR_AVAILABLE_FOR_TI	Boolean	Should this reason be applicable to TIs?
RR_AGC_OID	Foreign key	The owning Agency that this rejection reason belongs to.

10.6.3 Harvest Authorisations

SITE

The *SITE* table contains high-level information about a Harvest Authorisation, and is used to group all the information applying to a specific harvest authorisation.

Note that the *SITE* table is badly named through historical accident.

Name	Type	Description
ST_OID	Primary key	
ST_TITLE	Text	The name of the Harvest Authorisation record.
ST_DESC	Text	A description of the authorisation record.
ST_LIBRARY_ORDER_NO	Text	An external Order Number (e.g. Library Order Number).
ST_NOTES	Text	
ST_PUBLISHED	Boolean	Records whether the "Published" checkbox is ticked.
ST_ACTIVE	Boolean	Records whether the harvest authorisation (and all associated permissions) is enabled or disabled.
ST_OWNING_AGENCY_ID	Foreign Key	The owning agency for this site.

URL_PATTERN

The *URL_PATTERN* table contains a URL or URL pattern.

The scope of each harvest authorisation (*SITE*) is defined by a set of URL patterns.

Name	Type	Description
UP_OID	Primary key	
UP_PATTERN	Text	The URL or URL pattern.
UP_SITE_ID	Foreign key	The key of the *SITE* this *URL_PATTERN* belongs to.

AUTHORISING_AGENT

The *AUTHORISING_AGENT* table contains information about an entity contacted in relation to harvesting a website.

Name	Type	Description
AA_OID	Primary key	
AA_NAME	Text	The name of the authorising agent.
AA_ADRESS	Text	The full address of the authorising agent.
AA_CONTACT	Text	The name of the individual contact for an organisation.
AA_EMAIL	Text	The email address of the authorising agent.
AA_PHONE_NUMBER	Text	The phone number of the authorising agent.
AA_DESC	Text	A description of the authorising agent.

SITE_AUTH_AGENCY

The *SITE_AUTH_AGENCY* table links each site with its list of authorising agencies. (Note this is a many-to-many relationship.)

Name	Type	Description
SA_SITE_ID	Primary key, Foreign key	The key of the *SITE*.
SA_AGENT_ID	Primary key, Foreign key	The key of the *AUTHORISING_AGENT*.

PERMISSION

The PERMISSION table contains information about a single permission that has been granted by an *AUTHORISING_AGENT* for a *SITE*.

Name	Type	Description
PE_OID	Primary key	
PE_ACCESS_STATUS	Constrained	The access status of the permission. This value is constrained by the accessStatusList list in wct-core-lists.xml.
PE_APPROVED_YN	Boolean	Not used.
PE_AVAILABLE_YN	Boolean	Not used.
PE_COPYRIGHT_STATEMENT	Text	A passage of text that the publisher requires be displayed with the harvested material.
PE_COPYRIGHT_URL		A URL (linking to a copyright statement) that the publisher requires to be displayed with the harvested material.
PE_CREATION_DATE	Timestamp	The date and time the permission record was created.
PE_END_DATE	Timestamp	The date the permission information stored in this record expires (i.e. this permission only applies to harvests that occur between *PE_START_DATE* and *PE_END_DATE*).
PE_NOTES	Text	As of release 1.6.0 used to hold Auth Agency Response.
PE_OPEN_ACCESS_DATE	Timestamp	The date the rights over the harvested material expire and the material can be freely distributed.
PE_PERMISSION_GRANTED_DATE	Timestamp	The date the permission was granted (or rejected).
PE_PERMISSION_REQUESTED_DATE	Timestamp	The date the permission was requested.
PE_SPECIAL_REQUIREMENTS	Text	A passage of text describing any special requirements for the use of the harvested material.
PE_START_DATE		The date the permission information stored in this record expires (i.e. this permission only applies to harvests that occur between *PE_START_DATE* and *PE_END_DATE*).
PE_STATUS	Number	The current state of the Target Instance. Values correspond to: Pending, Requested, Approved, Rejected.
PE_AUTH_AGENT_ID	Foreign key	The key of the *AUTHORISING_AGENT* who has authorised this permission record.
PE_SITE_ID	Foreign key	The key of the Harvest Authorisation (i.e. *SITE*) that this permission applies to.
PE_QUICK_PICK	Boolean	Records whether this permission appears in the *Authorisation* drop-down menu in the Seeds tab in the Target editing interface.
PE_DISPLAY_NAME	Text	Label to use in the “Authorisation” drop-down menu in the Seeds tab in the Target editing interface (if *PE_QUICK_PICK* is set).
PE_OWNING_AGENCY_ID	Foreign key	The key of the Agency that has been granted authorisation by this permission record.
PE_FILE_REFERENCE	Text	An external reference number relating to this permission record (e.g. the file number of a permission letter).

PERMISSION_URLPATTERN

The *PERMISSION_URLPATTERN* table links *PERMISSION* records to the *URL_PATTERN* records that apply to them. Each permission will apply to one or more URL Patterns.

Name	Type	Description
PU_URLPATTERN_ID	Primary key, Foreign key	The key of the URL Pattern.
PU_PERMISSION_ID	Primary key, Foreign key	The key of the Permission record.

PERMISSION_EXCLUSION

The *PERMISSION_EXCLUSION* table contains information about a URL pattern that has been excluded from a *PERMISIION*.

Name	Type	Description
PEX_OID	Primary key	
PEX_REASON	Text	The reason for the exclusion.
PEX_URL	Text	The URL or URL Pattern that has been excluded.
PEX_PERMISSION_OID	Foreign key	The key of the permission that this is an exclusion to.
PEX_INDEX	Number	Internal number for maintaining the order of elements in a list.

SEED_PERMISSION

SEED_PERMISSION contains information about the associations between Seed URLs and the permission records that apply to them.

Name	Type	Description
SP_SEED_ID	Primary key, Foreign key	The key of a Seed URL.
SP_PERMISSION_ID	Primary key, Foreign key	The key of a permission record that is linked to the Seed URL.

URL_PERMISSION_MAPPING

URL_PERMISSION_MAPPING contains information about the associations between *URL_PATTERNS* and the permission records they apply to.

Name	Type	Description
UPM_OID		
UPM_PERMISSION_ID		The key of the permission record.
UPM_URL_PATTERN_ID		The key of a URL Pattern that is linked to this permission record.
UPM_DOMAIN		The most specific part of the domain, used for quick matching of seeds to permissions. For *global* patterns, this will be *.

Profiles and profile overrides

PROFILE

PROFILE contains information describing a single Heritrix profile.

Name	Type	Description
P_OID	Primary key	
P_VERSION	Number	Internal version number for optimistic locking.
P_DESC	Text	A textual description of the profile.
P_NAME	Text	The name of the profile.
P_PROFILE_STRING	Text	The profile itself, stored as an XML document.
P_PROFILE_LEVEL	Number	The level of the profile (controls which users may use the profile).
P_STATUS	Number	The current status of the profile.
P_DEFAULT	Boolean	Records whether this profile is the default profile for the Agency.
P_AGENCY_OID	Foreign key	The key of the Agency that this profile belongs to.
P_ORIG_OID	Number	The oid of the profile that this is a (usually locked) copy of.

PROFILE_OVERRIDES

PROFILE_OVERRIDES contains information describing the overrides to a profile pertaining to a specific *ABSTRACT_TARGET* (or its Target Instances).

Name	Type	Description
PO_OID	Primary key	
PO_EXCL_MIME_TYPES	Text	A list of MIME types to exclude from the harvest.
PO_MAX_BYES	Number	The maximum quantity of data to download (in bytes).
PO_MAX_DOCS	Number	The maximum number of documents to download.
PO_MAX_HOPS	Number	The maximum distance to crawl (in Heritrix "hops").
PO_MAX_PATH_DEPTH	Number	The maximum distance to crawl (in path depth from the website root).
PO_MAX_TIME_SEC	Number	The maximum time to spend on the harvest (in seconds).
PO_ROBOTS_POLICY	Text	Specifies whether the obots.txt file should be consulted or ignored. Either *ignore* or *classic*.
PO_OR_CREDENTIALS	Boolean	Specifies whether the Target has any credentials (i.e. usernames and passwords) stored in the *PROFILE_CREDENTIALS* and related tables.
PO_OR_EXCL_MIME_TYPES	Boolean	Specifies whether the *PO_EXCL_MIME_TYPES* override is activated.
PO_OR_EXCLUSION_URI	Boolean	Specifies whether the Target has any URL exclusions stored in the *PO_EXCLUSION_URI* table.
PO_OR_INCLUSION_URI	Boolean	Specifies whether the Target has any URL inclusions stored in the *PO_INCLUSION_URI* table.
PO_OR_MAX_BYTES	Boolean	Specifies whether the *PO_MAX_BYES* override is activated.
PO_OR_MAX_DOCS	Boolean	Specifies whether the *PO_MAX_DOCS* override is activated.
PO_OR_MAX_HOPS	Boolean	Specifies whether the *PO_MAX_HOPS* override is activated.
PO_OR_MAX_PATH_DEPTH	Boolean	Specifies whether the *PO_MAX_PATH_DEPTH* override is activated.
PO_OR_MAX_TIME_SEC	Boolean	Specifies whether the *PO_MAX_TIME_SEC* override is activated.
PO_OR_ROBOTS_POLICY	Boolean	Specifies whether the *PO_ROBOTS_POLICY* override is activated.

PO_EXCLUSION_URI

The *PO_EXCLUSION_URI* table contains information about a URL patterns that have been excluded from a *PROFILE_OVERRIDE*.

Name	Type	Description
PEU_IX	Primary key	
PEU_PROF_OVER_OID	Foreign key	The key of the *PROFILE_OVERRIDES* that this exclusion applies to.
PEU_FILTER	Text	The URL pattern excluded (a PERL regular expression).

PO_INCLUSION_URI

The *PO_INCLUSION_URI* table contains information about a URL patterns that have been un-excluded from a *PROFILE_OVERRIDE* (i.e. patterns that are exceptions to exclusions in *PO_EXCLUSION_URI*).

Name	Type	Description
PEU_IX	Primary key	
PEU_PROF_OVER_OID	Foreign key	The key of the *PROFILE_OVERRIDES* that this un-exclusion applies to.
PEU_FILTER	Text	The URL pattern included (a PERL regular expression).

PROFILE_CREDENTIALS

PROFILE_CREDENTIALS contains shared credential information used by both basic and form credentials.

Name	Type	Description
PC_OID	Primary key	
PC_DOMAIN	Text	The Internet domain this credential applies to.
PC_PASSWORD	Text	The password for this credential.
PC_USERNAME	Text	The username for this credential.
PC_PROFILE_OVERIDE_OID	Foreign key	The key of the *PROFILE_OVERRIDES* that these credentials apply to.
PC_INDEX	Number	Internal number for maintaining the order of elements in a list.

PROFILE_BASIC_CREDENTIALS

PROFILE_BASIC_CREDENTIALS is an extension of *PROFILE_CREDENTIALS* that contains credential information in *basic* credential format.

Name	Type	Description
PBC_PC_OID	Primary key, Foreign key	The key of the *PROFILE_CREDENTIALS* that this credential extends.
PBC_REALM	Text	The realm this credential applies to.

PROFILE_FORM_CREDENTIALS

PROFILE_FORM_CREDENTIALS is an extension of *PROFILE_CREDENTIALS* that contains credential information in "form" credential format.

Name	Type	Description
PRC_PC_OID	Primary key, Foreign key	The key of the *PROFILE_CREDENTIALS* that this credential extends.
PFC_METHOD	Text	The method for submitting the form.
PFC_LOGIN_URI	Text	The URL of the login form to use this credential against.
PFC_PASSWORD_FIELD	Text	The name of the password field used in the form.
PFC_USERNAME_FIELD	Text	The name of the username field used in the form.

10.6.4 Audit trail

WCTAUDIT

WCTAUDIT records all auditable events.

Each row in the table records a single auditable action, including the user who performed the action, the date and time, the object the action was performed on (i.e. the subject), and any message.

Name	Type	Description
AUD_OID	Primary key	
AUD_ACTION	Action	The auditable action performed.
AUD_DATE	Timestamp	The date and time the action was performed.
AUD_FIRSTNAME	Text	The first name of the user who performed the action.
AUD_LASTNAME	Text	The last name of the user who performed the action.
AUD_MESSAGE	Text	Additional text describing the action.
AUD_SUBJECT_TYPE	Text	The type of the object that was acted on.
AUD_USERNAME	Text	The username of the user who performed the action.
AUD_USER_OID	Foreign key	The key of the user who performed the action.
AUD_SUBJECT_OID	Foreign key	The key of the object that was acted on.
AUD_AGENCY_OID	Foreign key	The key of the agency that the user who performed the action belongs to.

WCT_LOGON_DURATION

WCT_LOGON_DURATION records the time and duration of all user sessions.

Each row in the table records a single user session.

Name	Type	Description
LOGDUR_OID	Primary key	
LOGDUR_DURATION	Number	The duration of the user session in seconds.
LOGDUR_LOGON_TIME	Timestamp	The date and time the user logged on to the WCT.
LOGDUR_LOGOUT_TIME	Timestamp	The date and time the user logged out of the WCT.
LOGDUR_USERNAME	Text	The username of the user.
LOGDUR_USER_OID	Foreign key	The key of the user.
LOGDUR_USER_REALNAME	Text	The full name of the user.
LOGDUR_SESSION_ID	Text	The Apache Tomcat Session ID.

10.6.5 Agencies, Roles and Users

AGENCY

AGENCY contains information describing an agency.

Name	Type	Description
AGC_OID	Primary key	
AGC_NAME	Text	The name of the agency.
AGC_ADDRESS	Text	The address of the agency.
AGC_LOGO_URL	Text	A URL for the logo of the agency.
AGC_URL	Text	The URL of the Agency
AGC_EMAIL	Text	The agency email address.
AGC_FAX	Text	The agency fax number.
AGC_PHONE	Text	The agency phone number.
AGC_SHOW_TASKS	Boolean	Whether the tasks list is shown on the notifications page for users in this agency. Default is true.

WCTROLE

WCTROLE contains information about a role.

Each role is associated with a single agency. The privileges attached to the role are stored in the ROLE_PRIVILEGE table.

Name	Type	Description
ROL_OID	Primary key	
ROL_DESCRIPTION	Text	Description of the role.
ROL_NAME	Text	Name of the role.
ROL_AGENCY_OID	Foreign key	The key of the agency that this role belongs to.

ROLE_PRIVILEGE

ROLE_PRIVILEGE records the privileges, and the scope of privileges, associated with each role.

Each role can have any number of privileges associated with it. Privileges are identified by the *PRV_CODE*, a unique code used by the WCT to represent each privilege. These are codes are hard-coded in the WCT, where they are used to determine whether users can perform particular actions.

Name	Type	Description
PRV_OID	Primary key	
PRV_CODE	Text	The code identifying the privilege being set.
PRV_ROLE_OID	Foreign key	The key of the role this privilege is associated with.
PRV_SCOPE	Number	The scope of the privilege as it applies to this role (i.e. whether the privilege applies to all data, agency data, or only the data owned by the user). 0 = All; 100 = Agency; 200 = Owner; 500 = None.

WCTUSER

WCTUSER contains information describing the WCT users.

Name	Type	Description
USR_OID	Primary key	
USR_ACTIVE	Boolean	Specifies whether the user is currently active or disabled.
USR_ADDRESS	Text	The user’s physical or postal address.
USR_EMAIL	Text	The user’s email address.
USR_EXTERNAL_AUTH	Boolean	Specifies whether the user should be authenticated using an external LDAP service, or using the internal authentication system.
USR_FIRSTNAME	Text	The user’s first name.
USR_FORCE_PWD_CHANGE	Boolean	Specifies whether the user should be forced to reset their password next time they log on to the WCT.
USR_LASTNAME	Text	The user’s last name.
USR_PASSWORD	Text	The user’s (encrypted) password.
USR_PHONE	Text	The user’s phone number.
USR_TITLE	Text	The user’s title.
USR_USERNAME	Text	The unique username identifying the user.
USR_AGC_OID	Foreign key	The key of the Agency that the user belongs to.
USR_DEACTIVATE_DATE	Timestamp	The date when the user was deactivated.
USR_NOTIFICATIONS_BY_EMAIL	Boolean	True if the user wants to receive notifications by emails as well as to their WCT in-tray.
USR_TASKS_BY_EMAIL	Boolean	True if the user wants to receive tasks by email as well as to their WCT in-tray.
USR_NOTIFIY_ON_GENERAL	Boolean	True if the user wants to receive general notifications.
USR_NOTIFY_ON_WARNINGS	Boolean	True if the user wants to receive notifications for warnings (such as memory warnings from the Harvest Agent).

USER_ROLE

USER_ROLE contains information linking users and roles.

Each row contains a user key and a role key, indicating that the specified user has been assigned the specified role.

Name	Type	Description
URO_USR_OID	Foreign key	The key of the user.
URO_ROL_OID	Foreign key	The key of the role.

PERMISSION_TEMPLATE

PERMISSION_TEMPLATE contains information describing a permission request template.

Name	Type	Description
PRT_OID	Primary key	
PRT_AGC_OID	Foreign key	The key of the Agency this template belongs to.
PRT_TEMPLATE_TEXT	Text	The text of the permission letter template.
PRT_TEMPLATE_NAME	Text	The name of the template.
PRT_TEMPLATE_TYPE	Text	The type of template (either *Print Template* or *Email Template*).
PRT_TEMPLATE_DESC	Text	The description of the template.
PRT_TEMPLATE_SUBJECT	Text	The subject of the Email.
PRT_TEMPLATE_OVERWRITE_FROM	Boolean	A flag used to control if the from field of the email is overwritten by the *PRT_TEMPLATE_FROM*.
PRT_TEMPLATE_FROM	Text	The email address used in the sent from field.
PRT_TEMPLATE_CC	Text	Email address(s) the emails are cc'd to.
PRT_TEMPLATE_BCC	Text	Email address(s) the emails are bcc'd to.
PRT_TEMPLATE_REPLY_TO		Email address used as the reply-to in permission emails.

TASK

TASK contains information describing a WCT task.

When a task is created, it is not assigned to a user, but will be displayed (and emailed) to all the users in the same agency who have sufficient rights to perform the task. When one of these users "claims" the task, it will no longer be displayed to the other users. When the user completes the task, it will be removed from their task list and deleted.

Name	Type	Description
TSK_OID	Primary key	
TSK_USR_OID	Foreign key	The key of the user who has claimed (or been assigned) the task, if any.
TSK_MESSAGE	Text	The message describing the task.
TSK_SENDER	Text	The email address of the sender of the task.
TSK_SENT_DATE	Timestamp	The date and time the task was created.
TSK_SUBJECT	Text	The subject line of the task, used in the InTray and in email notifications.
TSK_PRIVILEGE	Text	The privilege code that a user must have in order to complete the task. This field identifies which users will see an unassigned task.
TSK_AGC_OID	Foreign key	The key of the agency this task belongs to.
TSK_MSG_TYPE	Text	A type code for the message.
TSK_RESOURCE_OID	Foreign key	The key of the object this task will be performed on.
TSK_RESOURCE_TYPE	Text	The type of object the *TSK_RESOURCE_OID* identifies.

10.6.6 Other tables

ANNOTATIONS

The *ANNOTATIONS* table contains information about annotations. Annotations can be attached to many types of object, including Targets, target Instances, and Permissions.

Name	Type	Description
AN_OID	Primary key	
AN_DATE	Timestamp	The date the annotation was created.
AN_NOTE	Text	The text of the annotation.
AN_USER_OID	Foreign key	The foreign key of the user who created the annotation.
AN_OBJ_OID	Foreign key	The foreign key of the object that the annotation is attached to.
AN_OBJ_TYPE	Number	Specifies the type of object that the annotation is attached to.
AN_ALERTABLE	Boolean	Is this annotation to display a warning in the GUI.

BANDWIDTH_RESTRICTIONS

The *BANDWIDTH_RESTRICTIONS* table records the bandwidth restrictions in place at different intervals.

Name	Type	Description
BR_OID	Primary key	
BR_BANDWIDTH	Number	The bandwidth level for an interval.
BR_DAY	Text	The day the interval applies to.
BR_END_TIME	Timestamp	The end time of the interval.
BR_START_TIME	Timestamp	The start time of the interval.
BR_OPTIMIZATION_ALLOWED	Boolean	Whether harvest optimization is permitted during this restriction period.

DUBLIN_CORE

The *DUBLIN_CORE* table records the Dublin Core metadata for a Target.

Name	Type	Description
DC_OID	Primary key	
DC_CONTRIBUTOR	Text	Dublin Core metadata value.
DC_COVERAGE	Text	Dublin Core metadata value.
DC_CREATOR	Text	Dublin Core metadata value.
DC_DESCRIPTION	Text	Dublin Core metadata value.
DC_FORMAT	Text	Dublin Core metadata value.
DC_IDENTIFIER	Text	Dublin Core metadata value.
DC_IDENTIFIER_ISBN	Text	Dublin Core metadata value.
DC_IDENTIFIER_ISSN	Text	Dublin Core metadata value.
DC_LANGUAGE	Text	Dublin Core metadata value.
DC_PUBLISHER	Text	Dublin Core metadata value.
DC_RELATION	Text	Dublin Core metadata value.
DC_SOURCE	Text	Dublin Core metadata value.
DC_SUBJECT	Text	Dublin Core metadata value.
DC_TITLE	Text	Dublin Core metadata value.
DC_TYPE	Text	Dublin Core metadata value.

HARVEST_STATUS

The *HARVEST_STATUS* table records information about a specific Heritrix Harvest.

Name	Type	Description
HS_OID	Primary key	
HS_AVG_KB	Double	Average Kilobytes per second downloaded.
HS_AVG_URI	Double	Average number of URLs per second downloaded.
HS_DATA_AMOUNT	Number	Total data downloaded.
HS_ELAPSED_TIME	Number	Elapsed time of the harvest.
HS_JOB_NAME	Text	The identifier of the harvest job.
HS_STATUS	Text	The status of the harvest.
HS_URLS_DOWN	Number	The number of URLs downloaded.
HS_URLS_FAILED	Number	The number of URLs that filed to download.
HS_ALERTS	Number	The umber of alerts reported by the harvester during the crawl.
HS_HRTX_VERSION	Text	Version of Heretrix used during harvest.
HS_APP_VERSION	Text	Version of WCT used during harvest.

NOTIFICATION

The *NOTIFICATION* table records information about notifications sent to users.

Name	Type	Description
NOT_OID	Primary key	
NOT_MESSAGE	Text	The message text.
NOT_USR_OID	Foreign key	The foreign key of the user who will receive the notification.
NOT_SENDER	Text	The email address of the sender of the notification.
NOT_SENT_DATE	Timestamp	The date the notification was sent.
NOT_SUBJECT	Text	The subject line of the notification.

ID_GENERATOR

The *ID_GENERATOR* table is used to generate globally unique identifiers for objects in the database/. See *Generating primary keys* below for details.

Name	Type	Description
IG_TYPE	Text	The object type (or types) that this range of Identifier numbers applies to.
IG_VALUE	Number	The range of identifier numbers.

FLAG

The *FLAG* table defines arbitrary flag groups that are used to progress target instances through the WCT workflow. Each flag is allocated a description and colour.

Name	Type	Description
F_OID	Primary key	Unique identifier for the flag.
F_NAME	Text	Name for the flag group.
F_RGB	Text	Colour for the flag.
F_COMPLEMENT_RGB	Text	Complement colour for the flag (used to set a contrasting colour for the flag name).
F_AGC_OID	Foreign key	The foreign key of the agency that owns the flag.

INDICATOR_CRITERIA

The *INDICATOR_CRITERIA* table defines a template for the QA indicators. The template is used to initialise the indicators for a specific target instance (see the *INDICATOR* table).

Name	Type	Description
IC_OID	Primary key	Unique identifier for the indicator criteria.
IC_NAME	Text	Name for the indicator.
IC_DESCRIPTION	Text	Description for the indicator.
IC_UPPER_LIMIT_PERCENTAGE	Number	Upper limit used to define the upper watermark for the indicator as a percentage (eg: +10%).
IC_LOWER_LIMIT_PERCENTAGE	Number	Lower limit used to define the lower watermark for the indicator as a percentage (eg: -10%).
IC_UPPER_LIMIT	Number	Absolute value for the indicator's upper limit.
IC_LOWER_LIMIT	Number	Absolute value for the indicator's lower limit.
IC_AGC_OID	Foreign key	The foreign key of the agency that owns the indicator criteria.
IC_UNIT	Text	Unit of measurement for the indicator's value used to format the value for display (byte, millisecond or integer).
IC_SHOW_DELTA	Boolean	Displays the indictor delta compared with the reference crawl if true.
IC_ENABLE_REPORT	Boolean	Hyperlinks the indicator when set to true and generates a report based on the contents of the *INDICATOR_REPORT_LINE* table.

INDICATOR

The *INDICATOR* table defines the QA indicators for a specific target instance.

Name	Type	Description
I_OID	Primary key	Unique identifier for the indicator.
I_IC_OID	Foreign key	The foreign key of the indicator criteria on which this indicator is based.
I_TI_OID	Foreign key	The foreign key of the target instance that owns this indicator.
I_NAME	Text	Name for the indicator.
I_FLOAT_VALUE	Number	Value of the indicator.
I_UPPER_LIMIT_PERCENTAGE	Number	Upper limit used to define the upper watermark for the indicator as a percentage (eg: +10%).
I_LOWER_LIMIT_PERCENTAGE	Number	Lower limit used to define the lower watermark for the indicator as a percentage (eg: -10%).
I_UPPER_LIMIT	Number	Absolute value for the indicator's upper limit.
I_LOWER_LIMIT	Number	Absolute value for the indicator's lower limit.
I_ADVICE	Text	The advice issued for this indicator (eg: Reject).
I_JUSTIFICATION	Text	The justification for the advice reached for this indicator (eg: The content downloaded is 0KB).
I_AGC_OID	Foreign key	The foreign key of the agency that owns the indicator criteria.
I_UNIT	Text	Unit of measurement for the indicator's value used to format the value for display (byte, millisecond or integer).
I_SHOW_DELTA	Boolean	Displays the indicator delta compared with the reference crawl if true.
I_INDEX	Number	Display order for the indicator.
I_DATE	Date	Date on which the indicator was generated.

INDICATOR_REPORT_LINE

The *INDICATOR_REPORT_LINE* table is used to compile a report of the subject of the indicator (eg: for the Missing URLs indicator, each record in the *INDICATOR_REPORT_LINE* table represents a missing URL for that indicator).

Name	Type	Description
IRL_OID	Foreign key	The foreign key of the indicator that owns this report line.
IRL_LINE	Text	The indicator report line (eg: the missing URL for the Missing URLs indicator).
IRL_INDEX	Number	Display order for the indicator report line.

HEATMAP_CONFIG

HEATMAP_CONFIG contains the names and colors of the thresholds for the scheduling heat-map introduced in WCT version *1.6.1*.

Name	Type	Description
HM_OID	Primary key	
HM_NAME	Text	Name of the threshold, used as an identifier.
HM_DISPLAY_NAME	Text	Display name of the threshold.
HM_COLOR	Text	RGB color of the threshold.
HM_THRESHOLD_LOWEST	Number	The lowest number of scheduled harvests on a given day to allow this indicator to be used.

10.7 Generating primary keys

The WCT stores all primary keys as numbers.

10.7.1 Tables involved

The *ID_GENERATOR* table is used to track the reservation of ID values in a number of different key sets.

The *ABSTRACT_TARGET*, *TARGET*, *GROUP*, *SEED* and other important tables share a set of keys that are controlled by the *ID_GENERATOR.IG_TYPE* value of *General*, ensuring that their object IDs will never clash. Other objects have their own *ID_GENERATOR* to ensure that the ID numbers do not grow too quickly.

10.7.2 Reserving sequence numbers

If you want to insert new rows into WCT fields, you need to reserve a sequence number. To get a sequence number you need to:

1. Ensure that WCT is shutdown.
2. List the sequences available by running:

   ```
   SELECT
     *
   FROM
     id_generator;
   ```

3. Select the sequence for the objects you want to create. If there is not a specific sequence, choose the General sequence.
4. Run the following, substituting the sequence name as appropriate, and note the values returned:

   ```
   SELECT
     ig_value,
     ig_value * 32768 AS MIN_RES_VAL,
     ig_value * 32768 + 32767 AS MAX_RES_VAL
   FROM
     id_generator
   WHERE
     ig_type LIKE '%General%';
   ```

5. Now update the table to reserve your sequence numbers, using the same ID Generator Key as above, and the IG_VALUE returned by the above select statement:

   ```
   UPDATE
     id_generator
   SET
     ig_value = ig_value+1
   WHERE
     ig_type LIKE '%General%' AND ig_value = :IG_VALUE;
   ```

6. If the update statements reports one record updated, then you have successfully reserved the range between *MIN_RES_VAL* and *MAX_RES_VAL*. If the update reports no records updated, then you must repeat the process from step three as someone else may have reserved the numbers you were after.

Once you have all the numbers you need you can restart WCT.

10.7.3 Notes

Note that different object types may use different runs of numbers; for example *ANNOTATION* objects have *IG_TYPE* Annotation. Also note that the *IG_TYPE* field contents include some strange whitespace (hence the use of *like* in the SQL code above).

Every time a sequence is reserved, all 32,676 values are reserved, regardless of whether they get used or not.

CHAPTER 11

Frequently Asked Questions

11.1 Additional TODO

- Placeholder for needed changes to this document. In future it may be useful to organize the questions in sections.

11.2 Introduction

The Web Curator Tool has many interconnected components and depends on several sets of technologies. This document aims to unravel some of that complexity by answering some frequently asked questions.

11.2.1 Contents of this document

Following this introduction, the FAQ covers each issue in a question and answer format.

11.3 Index of Questions

11.4 Questions

Why can't I login with a new user I've created? How do I change where my harvests are being stored? Why isn't WCT using Heritrix 3.x? Why can't I find my harvests in Wayback?

11.4.1 Q: Why can't I login with a new user I've created?

A: Check that you have assigned a Role with the Login permission to your new User. By default a new User is not assigned to any Roles. Start by creating a new Role that at least has the Login permission checked. Then when viewing your list of Users, select the *Roles* icon under *Action* buttons.

11.4.2 Q: How do I change where my harvests are being stored?

A: Your harvest collection is stored by the WCT-Store module. The location of this store can by set via the *wct-das.properties* file, located in */<path to tomcat>/webapps/wct-store/WEB-INF/classes/*. Each harvest is stored in a folder with the Target Instance number. Please note warc/arc files are only transferred here after the harvest has been completed.

```
# the base directory for the arc store
arcDigitalAssetStoreService.baseDir=C:/wct/store
```

11.4.3 Q: Why can't I find my harvests in Wayback?

A: You have configured Wayback integration with WCT, but when trying to review your harvest in Wayback you get the following message: *Resource Not In Archive*. There is no exact answer to why this might have happened, but there are several steps you can check to make sure the indexing process has worked.

- Check the harvest warc/arc file has been copied into the common location that Wayback is watching.
- Check that there is a corresponding index file with the same name in */<wayback dir>/index-data/merged/*.
- If there is no index file, check the folders inside */<wayback dir>/index-data/* for any sign of your harvest.
- If the index had been completed successfully you should see an entry for your harvest warc/arc in the */<wayback dir>/file-db/db.log* file.
- If you have moved your Wayback common location, check that the required configuration files have been updated correctly. Listed in the **'Wayback configuration<https://github.com/DIA-NZ/webcurator/wiki/Wayback-Integration>'_** page.
- Try restarting your Tomcat server.

CHAPTER 12

Indices and tables

- genindex
- modindex
- search