

Preservica Support Guide

v7.1.0



Table of Contents

References	iv
1. Introduction	1
1.1. Context	1
1.2. Purpose of This Document	1
1.3. Scope of This Documentation	1
2. Enabling Preservica Features	2
2.1. anonymous.read.only.registry.access	2
2.2. export.to.server.path	2
3. ArchivesSpace Standard Deployment	3
3.1. ArchivesSpace Configuration	3
3.2. Information Required	3
3.3. Initial Checks	4
3.3.1. Check API Access	4
3.3.2. Check Login Details	5
3.4. System Configuration	5
3.4.1. XML Schemas	5
3.4.2. XML Transforms	5
3.4.3. XML Documents	5
3.4.4. Reports	6
3.4.5. Workflow Definitions	6
3.4.6. Workflow Contexts	6
4. Adlib Standard Deployment	13
4.1. Adlib and Adlib-API Configuration	13
4.2. Information Required	13
4.3. Initial Checks	14
4.3.1. Check API Access	14
4.4. System Configuration	15
4.4.1. XML Schemas	15
4.4.2. XML Transforms	15
4.4.3. XML Documents	15
4.4.4. Reports	15
4.4.5. Workflow Definitions	15
4.4.6. Workflow Contexts	16
5. CALM Standard Deployment	21
5.1. CALM and CALM-API Configuration	21
5.2. CALM Data Checks	21
5.3. Information Required	21
5.4. Initial Checks	22
5.4.1. Check API Access	22
5.5. System Configuration	23
5.5.1. XML Schemas	23
5.5.2. XML Transforms	23
5.5.3. XML Documents	23
5.5.4. Reports	24
5.5.5. Workflow Definitions	24
5.5.6. Workflow Contexts	24
6. Standalone Upgrade Tools	29
6.1. Migration of Thumbnails from Universal Access	29
6.1.1. Results	29
7. AWS Bulk Upload Service	30
7.1. Overview	30
7.2. Review Volumetrics	30
7.3. Check the Ingest Bucket	30

7.4. Check the Ingest Collection	31
7.5. Order the AWS Snowball (Mandatory)	31
7.6. Create the Necessary Workflows (Mandatory)	31
7.7. Provision Additional JobQueue Nodes (Recommended)	32
7.8. Run the Ingest Process (Mandatory)	32
7.9. Problems and Recovery (Optional)	33
7.10. Post Ingest Tidy Up (Mandatory)	33
8. Instance Migration	34
8.1. Overview	34
8.1.1. Limitations	34
8.1.2. Scope of the Migration Service	34
8.1.3. Migration Timeline	35
8.2. Migration of System Components	36
8.2.1. Metadata Store	36
8.2.2. Solr Search Index	37
8.2.3. User Accounts	37
8.2.4. Bulk Storage	38
8.2.5. Preview Image Cache	39
8.2.6. Generated DIPs	39
8.2.7. Test Migrations	39
8.3. Migration Process	40
8.3.1. Configuration	40
8.3.2. Source System	42
8.3.3. Target System	43
8.4. Deletion of the Migrated Tenant	45
9. System Backups	46
9.1. Solr	46
9.1.1. Backing up using Solr snapshots	46
9.1.2. Backing up using Solr index replication.	46
10. Manual Database Changes	49
10.1. Adapters	49

References

Document	Ref	Date	Details & Issue
Preservica Guide to System Documentation	[DOC]	Sept 2022	git/doc/UG/SUG V6.6.0
Preservica AWS Bulk Upload Service	[ABU]	See [DOC] for version information.	
Preservica System Administration Guide	[SAG]		
Preservica System Maintenance Guide	[SMG]		

Chapter 1. Introduction

1.1. Context

This document is written for Cloud Edition but, in most cases, will be applicable to all Preservica editions.

1.2. Purpose of This Document

This document describes the standard configuration for specific Preservica options. Customisation of the standard options is outside of the scope of this document.

1.3. Scope of This Documentation

This is an internal document for the Preservica Support Team.

Chapter 2. Enabling Preservica Features

Preservica supports different configuration options available in its `features.properties` file. The standard locations for this file is at `<preservica_install_directory>/shared/classes/com/tessella/sdb/core/features.properties`. Features can be enabled by setting properties names to `true`. If the setting is not present, the default is `false` unless specified. The following sections summarise these properties.

2.1. `anonymous.read.only.registry.access`

Set this to `true` to deploy Preservica with a Linked Data Registry that has no authentication and is read only. This setting should only be used for Preservica's publicly accessible registry. It should not be used for customer deployments.

2.2. `export.to.server.path`

Set this to `true` to make workflow context parameters visible that are designed to export to a file path accessible to the JobQueue server. For example, the *Sync Catalogue to Preservica* workflow is able to export an audit of catalogue ids to a file location. However, this option would not be appropriate for cloud-based Preservica deployments where the user cannot access the file system, so by not setting the `export.to.server.path` property, this workflow context parameter will not be shown.

Chapter 3. ArchivesSpace Standard Deployment

3.1. ArchivesSpace Configuration

A small number of changes need to be made to the default ArchivesSpace configuration to enable Preservica to link correctly to ArchivesSpace. This configuration must be done by the customer prior to linking Preservica to ArchivesSpace:

- The ArchivesSpace admin user account should have its password changed from the default.
- Preservica communicates with ArchivesSpace via its back-end REST web service. By default this runs unencrypted on port 8089. However, this should be changed to use encrypted HTTPS.
- The firewall in front of ArchivesSpace should be set up to only allow incoming connections from Preservica.
- New control list values should be added to ArchivesSpace to store the Preservica Explorer and Universal Access URLs with the following steps.

1. From ArchivesSpace home page select 'System > Manage Control Value Lists'
2. Choose the list name to be 'File Version Use Statement (file_version_use_statement)'
3. Click 'Create Value' to add 3 new values to the list with the following names:
 - a. Digital Preservation Staff System
 - b. Electronic Record Access System
 - c. Digital Preservation Link
4. The list should now include the following 3 new entries at the bottom.

video-streaming	Video-Streaming	23	↑ ↓		Set as Default Suppress Merge Delete
Electronic Record Access System	Electronic Record Access System	24	↑ ↓		Set as Default Suppress Merge Delete
Digital Preservation Staff System	Digital Preservation Staff System	25	↑ ↓		Set as Default Suppress Merge Delete
Digital Preservation Link	Digital Preservation Link	26	↑		Set as Default Suppress Merge Delete

3.2. Information Required

The following information is required to configure Preservica to link to ArchivesSpace.

- eMail address for a single user or eMail group that will receive the Illegal Catalogue Moves email and Missing Catalogue Records Report.
- URL of the ArchivesSpace user interface e.g. <https://archivesspace.preservica.com>. By default the ArchivesSpace user interface runs on port 8081 but connections are typically proxied via https using Apache httpd.
- URL of the ArchivesSpace API interface e.g. <https://api-archivesspace.preservica.com>. By default the ArchivesSpace API interface runs on port 8089 so a common URL is <https://archivesspace.preservica.com:8089>. Connections may be proxied via httpd so the port may not be required.
- Username for a valid ArchivesSpace user. Creating a dedicated user for the Preservica interface to use (e.g. preservica) is recommended.
- Password for user account provided.

Without the user account details we will be unable to fully test access from Preservica to ArchivesSpace and will not be able to complete the configuration of Preservica.

In addition, a top level collection to hold records awaiting synchronisation to ArchivesSpace is required. We should inform customers that we will create a Top Level collection called "Pending Synchronisation" unless they inform us otherwise. In which case they should either give us an alternative name to create or create the collection and tell us the following information:

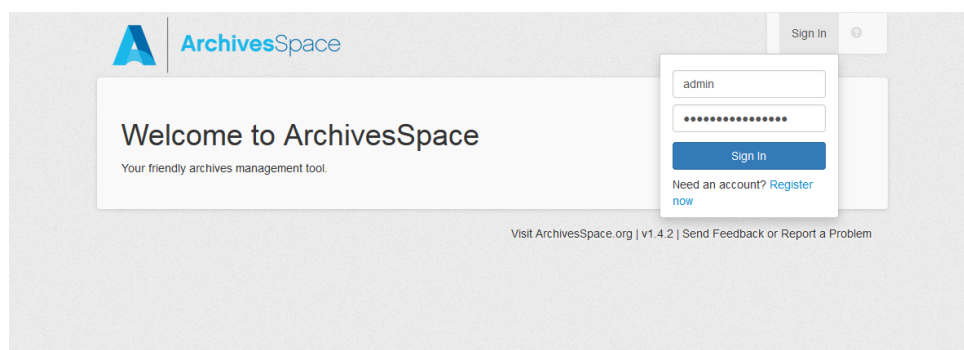
- Name
- Collection Code:
- References

The customer will need to complete the Preservica ArchivesSpace Synchronisation Configuration Document. This provides us with the necessary information to configure the synchronisation process. This document will take some time to complete. The higher level information requested above should be provided first to allow us to complete the initial checks.

3.3. Initial Checks

Once the ArchivesSpace information has been provided a number of checks should be made to ensure that the Preservica system can successfully connect to ArchivesSpace.

Initially the URLs and user details provided can be checked locally from a browser. The ArchivesSpace login page should be displayed but this may fail if the customer has configured firewall rules to restrict access to the Preservica system only.



Connectivity should be checked from the Preservica JobQueue server using the wget application.

3.3.1. Check API Access

It may be possible to access the ArchivesSpace API using the URL provided without any user details:

```
# wget https://archivesspace.preservica.com:8089
```

Typically this will return basic configuration information contained in the file `index.html`:

```
{
  "databaseProductName": "MySQL",
  "databaseProductVersion": "5.6.24-log",
  "ruby_version": "1.9",
  "host_os": "mswin32",
  "host_cpu": "x86_64",
  "build": "java1.6",
```



```
"archivesSpaceVersion": "v1.4.2"
}
```

3.3.2. Check Login Details

It should also be possible to check the URL and user details provided by accessing the ArchivesSpace API e.g.:

```
# wget --user=admin --password=passwd
https://archivespace.preservica.com:8089/repositories
```

If the user credentials are correct this should return a list of the ArchivesSpace repositories available in the file `repositories`.

3.4. System Configuration

Once connectivity to ArchivesSpace has been confirmed Preservica should be set-up with the standard ArchivesSpace interface configuration. This will require a number of files to be loaded into the appropriate tenancy and a number of workflow contexts to be created.



All the XML files are renamed replacing Default with a short tenant name/identifier. This name is persisted if the file is downloaded from Preservica and should prevent bulk updates to files during the release process updating any files accidentally.

3.4.1. XML Schemas

Rename the standard ArchivesSpace schema `ArchivesSpace_Default.xsd` to `ArchivesSpace_<TENANT>.xsd` e.g. `ArchivesSpace_AP.xsd` and upload using the standard Preservica administration functionality. The following settings should be used:

- **Schema Name:** ArchivesSpace
- **Description:** As required

3.4.2. XML Transforms

Rename the standard ArchivesSpace metadata viewer `ArchivesSpace-viewer_Default.xslt` to `ArchivesSpace-viewer_<TENANT>.xslt` e.g. `ArchivesSpace-viewer_AP.xslt` and upload using the standard Preservica administration functionality. The following settings should be used:

- **Name:** ArchivesSpace Viewer
- **XSD Input:** ArchivesSpace (ie the name of the schema added earlier)
- **XSD Output:** XHTML
- **Purpose:** View

3.4.3. XML Documents

Rename the standard ArchivesSpace mapping document `ArchivesSpace-mapping_Default.xml` to `ArchivesSpace-mapping_<TENANT>.xml` e.g. `ArchivesSpace-mapping_AP.xml` and upload using the standard Preservica administration functionality. The following settings should be used:

- **Name:** ArchivesSpace Mapping
- **Context:** Configuration File

Rename the standard ArchivesSpace custom index document `ArchivesSpace-search_Default.xml` to `ArchivesSpace-search_<TENANT>.xml` e.g. `ArchivesSpace-search_AP.xml` and upload using the standard Preservica administration functionality. The following settings should be used:

- **Name:** ArchivesSpace Indexer
- **Context:** Custom Index Definition

3.4.4. Reports

If not already loaded the standard report `MissingCatalogueRecords.jrxml` (Missing Catalogue Records) should be uploaded using the standard Preservica administration functionality.

3.4.5. Workflow Definitions

The following workflow definitions should be uploaded using the standard Preservica administration functionality.

All Preservica Editions:

- `data.management.link.sdb.to.catalogue.rf`
- `data.management.sync.catalogue.to.sdb.rf`
- `data.management.report.deleted.catalogue.records.rf`
- `data.management.delete.aip.and.catalogue.rf`
- `data.management.recover.deleted.records.and.catalogue.rf`

Preservica CE Only:

- `ingest.ce.single.auto.standard.catalogue.rf`

Preservica EE/SE Only:

- `ingest.single.auto.standard.catalogue.rf`



Ingest with normalisation and immediate synchronisation to ArchivesSpace is not supported.

3.4.6. Workflow Contexts

The following workflow contexts should be created using the standard Preservica administration functionality.

The recommended order for the nightly scheduled workflows is:

1. Synchronise Preservica to Catalogue
2. Synchronise Catalogue to Preservica
3. Report Deleted Catalogue Records

This is reflected in the scheduled times suggested below.

3.4.6.1. Synchronise Preservica to Catalogue

This is an additional workflow workflow that finds records within Preservica that have not yet been synchronised to ArchivesSpace and updates ArchivesSpace. If records are being synchronised on ingest this

workflow will deal with outstanding records e.g. where the ArchivesSpace was not available during ingest. If records are not being synchronised on ingest this workflow will undertake a batch synchronisation.

The workflow context should be set-up using the `data.management.link.sdb.to.catalogue.rf` (Link Preservica to Catalogue) definition. The following settings should be used:

- **Name:** Synchronise Preservica to Catalogue
- **Description:** Synchronise Records to ArchivesSpace
- **Email Address:** Operator email as required (copy from existing workflow)
- **Notification on error:** Enabled
- **Notification on complete:** Disabled
- **Automatically Terminate:** Enabled
- **Allow Concurrent:** Disabled
- **Catalogue URL:** ArchivesSpace API URL e.g. `https://archivespace.preservica.com:8089`
- **Catalogue Database:** Not required for ArchivesSpace enter N/A as field is mandatory
- **Catalogue Username:** ArchivesSpace username as provided by customer
- **Catalogue Password:** ArchivesSpace password as provided by customer
- **Catalogue Mapping Document:** Select loaded mapping file by name i.e. `ArchivesSpace Mapping`
- **Number of Levels to Sync:** Leave blank (all records will be synchronised)
- **Top Level Collection Ref:** "Pending Synchronisation" collection or alternative provided by customer
- **Deliverable Unit Ref:** Place-holder only - leave blank.
- **Trigger:** Scheduled on a Daily basis out of office hours e.g. 01:00

3.4.6.2. Synchronise Catalogue to Preservica

This is an additional workflow that finds records within ArchivesSpace that are linked to Preservica but have since changed and updates Preservica to reflect the changes.

The workflow context should be set-up using the `data.management.sync.catalogue.to.sdb.rf` (Synchronise Catalogue to Preservica) definition. The following settings should be used:

- **Name:** Report Deleted Catalogue Records
- **Description:** Report Catalogue Records deleted within ArchivesSpace
- **Email Address:** Operator email as required (copy from existing workflow)
- **Notification on error:** Enabled
- **Notification on complete:** Disabled
- **Automatically Terminate:** Enabled
- **Allow Concurrent:** Disabled
- **Catalogue URL:** ArchivesSpace API URL e.g. `https://archivespace.preservica.com:8089`
- **Catalogue Database:** Not required for ArchivesSpace enter N/A as field is mandatory

- **Catalogue Username:** ArchivesSpace username as provided by customer
- **Catalogue Password:** ArchivesSpace password as provided by customer
- **Catalogue Mapping Document:** Select loaded mapping file by name i.e. `ArchivesSpace Mapping`
- **Re Sync All Changes:** Disabled (required for recovery only)
- **eMail Address:** User email as provided by the customer
- **Tolerance Seconds:** 600 (This prevents missing updates due to minor time differences between servers)
- **Catalogue IDs Filename:** Leave blank (only applicable to HES)
- **Trigger:** Scheduled on a Daily basis out of office hours e.g. 02:00

3.4.6.3. Report Deleted Catalogue Records

This is an additional workflow that checks for records that exist in Preservica but link to a non-existent record (i.e. a deleted record) within ArchivesSpace. This workflow is run on a daily basis and produces a report that is eMailed to a nominated eMail address.

The workflow context should be set-up using the `data.management.report.deleted.catalogue.records.rf` (Report Deleted Catalogue Records) definition. The following settings should be used:

- **Name:** Report Deleted Catalogue Records
- **Description:** Report Catalogue Records deleted within ArchivesSpace
- **Email Address:** Operator email as required (copy from existing workflow)
- **Notification on error:** Enabled
- **Notification on complete:** Disabled
- **Automatically Terminate:** Enabled
- **Allow Concurrent:** Disabled
- **Report Name:** Select loaded report by name i.e. `Missing Catalogue Records`
- **Report Format:** Select required report format, typically PDF
- **Email Address:** User email as provided by the customer
- **Catalogue URL:** ArchivesSpace API URL e.g. `https://archivespace.preservica.com:8089`
- **Catalogue Database:** Not required for ArchivesSpace enter N/A as field is mandatory
- **Catalogue Username:** ArchivesSpace username as provided by customer
- **Catalogue Password:** ArchivesSpace password as provided by customer
- **Catalogue Mapping Document:** Select loaded mapping file by name i.e. `ArchivesSpace Mapping`
- **Trigger:** Scheduled on a Daily basis out of office hours e.g. 03:00

3.4.6.4. Delete AIP and Synchronise to Catalogue

This workflow replaces the standard Delete AIP workflow and will synchronise deletions to ArchivesSpace. Any instances of the standard Delete AIP workflow should be duplicated and deleted.

When a record is deleted from Preservica the matching record, and any child records, in ArchivesSpace are deleted. A note is added to the parent record in ArchivesSpace indicating that the child record has been deleted.

The workflow context should be set-up using the `data.management.delete.aip.and.catalogue.rf` (Delete AIP Workflow with Catalogue Sync) definition. The following settings should be used:

- **Name:** Delete API and Synchronise to Catalogue
- **Description:** Delete API and Synchronise deletions to ArchivesSpace
- **Email Address:** Operator email as required (copy from existing workflow)
- **Notification on error:** Enabled
- **Notification on complete:** Disabled
- **Automatically Terminate:** Enabled
- **Allow Concurrent:** Enabled
- **Operator Role:** As required (copy from existing Delete AIP workflow)
- **Supervisor Role:** As required (copy from existing Delete AIP workflow)
- **Confirmation email:** As required (copy from existing Delete AIP workflow)
- **Delete Collection:** As required (copy from existing Delete AIP workflow)
- **Catalogue URL:** ArchivesSpace API URL e.g. `https://archivespace.preservica.com:8089`
- **Catalogue Database:** Not required for ArchivesSpace enter N/A as field is mandatory
- **Catalogue Username:** ArchivesSpace username as provided by customer
- **Catalogue Password:** ArchivesSpace password as provided by customer
- **Catalogue Mapping Document:** Select loaded mapping file by name i.e. `ArchivesSpace Mapping`
- **Trigger:** Manual only



No action is required for the existing content deletion workflows as metadata will still exist in both Preservica and ArchivesSpace.

3.4.6.5. Restore Deleted Records and Synchronise to Catalogue

This workflow replaces the standard Restore Deleted Records workflow and will synchronise recovery actions to ArchivesSpace. The existing Restore Deleted Records workflow should be deleted once the new workflow is configured.

When a record is restored in Preservica the matching record, and any child records that were previously synchronised, are restored within ArchivesSpace. The note added to the ArchivesSpace parent record as part of the deletion process is not changed and no additional note is added to the parent record.

The workflow context should be set-up using the `data.management.recover.deleted.records.and.catalogue.rf` (Recover Deleted Records with Catalogue Sync) definition. The following settings should be used:

- **Name:** Restore Deleted Records and Synchronise to Catalogue

- **Description:** Restore Deleted Records and Synchronise restoration to ArchivesSpace
- **Email Address:** Operator email as required (copy from existing workflow)
- **Notification on error:** Enabled
- **Notification on complete:** Disabled
- **Automatically Terminate:** Enabled
- **Allow Concurrent:** Enabled
- **Catalogue URL:** ArchivesSpace API URL e.g. <https://archivespace.preservica.com:8089>
- **Catalogue Database:** Not required for ArchivesSpace enter N/A as field is mandatory
- **Catalogue Username:** ArchivesSpace username as provided by customer
- **Catalogue Password:** ArchivesSpace password as provided by customer
- **Catalogue Mapping Document:** Select loaded mapping file by name i.e. `ArchivesSpace Mapping`
- **Trigger:** Manual only

3.4.6.6. Auto Ingest and Synchronise to Catalogue (EE/SE Only)

This workflow replaces the standard Auto Ingest workflow and will attempt to immediately synchronise the ingested records to ArchivesSpace. If the synchronisation fails, the nightly **Synchronise Preservica to Catalogue** workflow will synchronise these records overnight. The existing Auto Ingest workflow should be deleted once the new workflow is configured.

The workflow context should be set-up using the `ingest.single.auto.standard.catalogue.rf` (Standard Ingest Workflow (Auto Selection, Catalogue sync)) definition. The following settings should be used:

- **Name:** As required (copy from existing Auto Ingest workflow)
- **Description:** As required (copy from existing Auto Ingest workflow)
- **Email Address:** Operator email as required (copy from existing workflow)
- **Notification on error:** Enabled
- **Notification on complete:** Disabled
- **Automatically Terminate:** Enabled
- **Allow Concurrent:** Enabled
- **SIP Location:** As required (copy from existing Auto Ingest workflow)
- **Collection:** "Pending Synchronisation" collection or alternative provided by customer
- **Transfer Agreement:** As required (copy from existing Auto Ingest workflow)
- **Classification Document:** As required (copy from existing Auto Ingest workflow)
- **Classification Class:** As required (copy from existing Auto Ingest workflow)
- **Classification Tag:** As required (copy from existing Auto Ingest workflow)
- **Storage Policy:** As required (copy from existing Auto Ingest workflow)

- **Secondary Storage:** As required (copy from existing Auto Ingest workflow)
- **Delete Source:** As required (copy from existing Auto Ingest workflow)
- **Catalogue Mapping Document:** Select loaded mapping file by name i.e. `ArchivesSpace Mapping`
- **Catalogue URL:** ArchivesSpace API URL e.g. `https://archivespace.preservica.com:8089`
- **Catalogue Database:** Not required for ArchivesSpace enter N/A as field is mandatory
- **Catalogue Username:** ArchivesSpace username as provided by customer
- **Catalogue Password:** ArchivesSpace password as provided by customer
- **Levels to Sync:** Leave blank to synchronise all child DUs to ArchivesSpace
- **Trigger:** File system watcher

3.4.6.7. Manual Ingest and Synchronise to Catalogue (EE/SE Only)

This workflow replaces the standard Manual Ingest workflow and will attempt to immediately synchronise the ingested records to ArchivesSpace. If the synchronisation fails, the nightly **Synchronise Preservica to Catalogue** workflow will synchronise these records overnight. The existing Manual Ingest workflow should be deleted once the new workflow is configured.

The workflow context should be set-up using the `ingest.single.manual.standard.catalogue.rf` (Standard Ingest Workflow (Manual Selection, Catalogue Sync)) definition. The following settings should be used:

- **Name:** As required (copy from existing Manual Ingest workflow)
- **Description:** As required (copy from existing Manual Ingest workflow)
- **Email Address:** Operator email as required (copy from existing workflow)
- **Notification on error:** Enabled
- **Notification on complete:** Disabled
- **Automatically Terminate:** Enabled
- **Allow Concurrent:** Enabled
- **SIP Location:** As required (copy from existing Manual Ingest workflow)
- **Collection:** "Pending Synchronisation" collection or alternative provided by customer
- **Transfer Agreement:** As required (copy from existing Manual Ingest workflow)
- **Classification Document:** As required (copy from existing Manual Ingest workflow)
- **Classification Class:** As required (copy from existing Manual Ingest workflow)
- **Classification Tag:** As required (copy from existing Manual Ingest workflow)
- **Storage Policy:** As required (copy from existing Manual Ingest workflow)
- **Secondary Storage:** As required (copy from existing Manual Ingest workflow)
- **Delete Source:** As required (copy from existing Manual Ingest workflow)
- **Catalogue Mapping Document:** Select loaded mapping file by name i.e. `ArchivesSpace Mapping`

- **Catalogue URL:** ArchivesSpace API URL e.g. <https://archivesspace.preservica.com:8089>
- **Catalogue Database:** Not required for ArchivesSpace enter N/A as field is mandatory
- **Catalogue Username:** ArchivesSpace username as provided by customer
- **Catalogue Password:** ArchivesSpace password as provided by customer
- **Levels to Sync:** Leave blank to synchronise all child DUs to ArchivesSpace
- **Trigger:** Manual Only

3.4.6.8. Auto Ingest and Synchronise to Catalogue (CE Only)

This workflow replaces the standard CE Auto Ingest workflow and will attempt to immediately synchronise the ingested records to ArchivesSpace. If the synchronisation fails, the nightly **Synchronise Preservica to Catalogue** workflow will synchronise these records overnight. The existing Auto Ingest workflow should be deleted once the new workflow is configured.

The workflow context should be set-up using the `ingest.ce.single.auto.standard.catalogue.rf` (Standard Ingest Workflow (Auto Selection, Catalogue sync)) definition. The following settings should be used:

3.4.6.9. Manual Ingest and Synchronise to Catalogue (CE Only)

This workflow replaces the standard CE Manual Ingest workflow and will attempt to immediately synchronise the ingested records to ArchivesSpace. If the synchronisation fails, the nightly **Synchronise Preservica to Catalogue** workflow will synchronise these records overnight. The existing Manual Ingest workflow should be deleted once the new workflow is configured.

The workflow context should be set-up using the `ingest.ce.single.manual.standard.catalogue.rf` (Standard Ingest Workflow (Manual Selection, Catalogue Sync)) definition. The following settings should be used:

Chapter 4. Adlib Standard Deployment

4.1. Adlib and Adlib-API Configuration

Adlib supports multiple databases for different types of records. However, it is only the 'Archives (catalogue)' database which is synchronised to Preservica. This database is only available if the Adlib 'Archives + Library' module has been installed which the customer will need to do prior to synchronising with Preservica.

For Preservica to be able to synchronise with Adlib, Adlib-API must be installed by the customer. Adlib-API is a separate Axiell- produced module that typically runs on the same server as Adlib using IIS. The following adjustments should be made to Adlib-API by the customer if it is connecting to Preservica over the public internet.

- Adlib-API/IIS should be installed with a non-default user name and strong password.
- Adlib-API/IIS should be installed to run using HTTPS only (by default it is configured to use unencrypted HTTP).
- The firewall in front of Adlib-API should be set up to only allow incoming connections from Preservica's JobQueue.

Adlib-API is a RESTful web api allowing record retrieval via GET requests, and record updates via POST requests. Documentation on Adlib-API is available at <http://api.adlibsoft.com/site/api>. Thus far, Preservica has not installed Adlib or Adlib-API. Instead Axiell have installed a test instance for Preservica that has matched that installed by a given customer. However, some additional configuration is required to allow Adlib-API to provide enough information to be able to synchronise with Preservica. This configuration involves the `adlibweb.xml` file. Adlib-API needs to be restarted if changes are made to `adlibweb.xml`.

- The `writeAllowed` attribute must be set to true for the database with which Preservica is to synchronise. This database is "collect.inf".
- Any fields that are required to be synchronised to Preservica must be made available either in the `brieffields` or `detailfields` sections.
- It is recommended that the parent fixed id field `part_of_reference.lref` and the human readable reference code field `object_number` are both added to the `brieffields` section as it make requests to get recently modified records faster.

4.2. Information Required

The following information is required to configure Preservica to link to Adlib.

- eMail address for a single user or eMail group that will receive the Illegal Catalogue Moves email and Missing Catalogue Records Report.
- URL of the Adlib-API REST interface e.g. <https://adlib.preservica.com/api/wwwopac.ashx>. By default the Adlib-API REST interface runs on port 80 so if using HTTPS the port is likely to be another port such as 443.
- Username for a valid Adlib-API user. Creating a dedicated user for the Adlib-API REST interface to use (eg `preservica`) is recommended.
- Password for the above Adlib-API user account provided.
- The index of the default language to use when retrieving multilingual fields from Adlib-API. Many Adlib-API fields are multilingual and Preservica needs to be configured to read the correct language fields when connecting. This is specified as an integer in the `DefaultLanguage` element in the Preservica catalogue mapping file.

Without the user account details we will be unable to fully test access from Preservica to Adlib-API and will not be able to complete the configuration of Preservica.

In addition, a top level collection to hold records awaiting synchronisation to Adlib is required. We should inform customers that we will create a Top Level collection called "Pending Synchronisation" unless they inform us otherwise. In which case they should either give us an alternative name to create or create the collection and tell us the following information:

- Name
- Collection Code

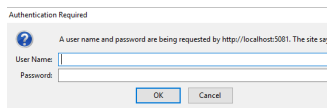
The customer will need to complete the Preservica Adlib Synchronisation Configuration Document. This provides us with the necessary information to configure the synchronisation process. This document will take some time to complete. The higher level information requested above should be provided first to allow us to complete the initial checks.

4.3. Initial Checks

4.3.1. Check API Access

Once the Adlib information has been provided a number of checks should be made to ensure that the Preservica system can successfully connect to Adlib.

Initially the Adlib-API URL and user details provided can be checked locally from a browser. A pop-up login dialogue should be displayed but this may fail if the customer has configured firewall rules to restrict access to the Preservica system only.



Connectivity should be checked from the Preservica JobQueue server using the wget application, for example.

```
# wget --user=admin --password=passwd  
https://adlib.preservica.com/api/wwwopac.ashx?database=collect.inf  
&command=getmetadata
```

If the user credentials are correct, this should return a list of the configured fields for this instance of Adlib, for example

```
<?xml version="1.0" encoding="UTF-8"?>  
<adlibXML>  
  <recordList>  
    <record>  
      <tag>IN</tag>  
      <fieldName>  
        <value lang="0">object_number</value>  
        <value lang="1">objectnummer</value>  
        <value lang="2">numéro_d_objet</value>  
        ... etc ...  
      </fieldName>  
    </record>  
  </recordList>  
</adlibXML>
```

4.4. System Configuration

Once connectivity to Adlib-API has been confirmed Preservica should be set-up with the standard Adlib interface configuration. This will require a number of files to be loaded into the appropriate tenancy and a number of workflow contexts to be created.



All the XML files are renamed replacing Default with a short tenant name/identifier. This name is persisted if the file is downloaded from Preservica and should prevent bulk updates to files during the release process updating any files accidental.

4.4.1. XML Schemas

Rename the standard Adlib schema `Adlib_Default.xsd` to `Adlib_<TENANT>.xsd` eg `Adlib_BG.xsd` and upload using the standard Preservica administration functionality. The following settings should be used:

- **Schema Name:** Adlib
- **Description:** As required

4.4.2. XML Transforms

Rename the standard Adlib metadata viewer `Adlib-viewer_Default.xslt` to `Adlib-viewer_<TENANT>.xslt` e.g. `Adlib-viewer_BG.xslt` and upload using the standard Preservica administration functionality. The following settings should be used:

- **Name:** Adlib Viewer
- **XSD Input:** Adlib (ie the name of the schema added earlier)
- **XSD Output:** XHTML
- **Purpose:** View

4.4.3. XML Documents

Rename the standard Adlib mapping document `Adlib-mapping_Default.xml` to `Adlib-mapping_<TENANT>.xml` e.g. `Adlib-mapping_BG.xml` and upload using the standard Preservica administration functionality. The following settings should be used:

- **Name:** Adlib Mapping
- **Context:** Configuration File

Rename the standard Adlib custom index document `Adlib-search_Default.xml` to `Adlib-search_<TENANT>.xml` e.g. `Adlib-search_BG.xml` and upload using the standard Preservica administration functionality. The following settings should be used:

- **Name:** Adlib Indexer
- **Context:** Custom Index Definition

4.4.4. Reports

If not already loaded the standard report `MissingCatalogueRecords.jrxml` (Missing Catalogue Records) should be uploaded using the standard Preservica administration functionality.

4.4.5. Workflow Definitions

The following workflow definitions should be uploaded using the standard Preservica administration functionality.

All Preservica Editions:

- `data.management.link.sdb.to.catalogue.rf`
- `data.management.sync.catalogue.to.sdb.rf`
- `data.management.report.deleted.catalogue.records.rf`



Deletion from Preservica with immediate synchronisation to Adlib is not supported.

Preservica CE Only:

- `ingest.ce.single.auto.standard.catalogue.rf`

Preservica EE/SE Only:

- `ingest.single.auto.standard.catalogue.rf`



Ingest with normalisation and immediate synchronisation to Adlib is not supported.

4.4.6. Workflow Contexts

The following workflow contexts should be created using the standard Preservica administration functionality.

The recommended order for the nightly scheduled workflows is:

1. Synchronise Preservica to Catalogue
2. Synchronise Catalogue to Preservica
3. Report Deleted Catalogue Records

This is reflected in the scheduled times suggested below.

4.4.6.1. Synchronise Preservica to Catalogue

This is an additional workflow that finds records within Preservica that have not yet been synchronised to Adlib and updates Adlib with Preservica metadata and vice-versa. If records are being synchronised on ingest this workflow will deal with outstanding records e.g. where the Adlib was not available during ingest. If records are not being synchronised on ingest this workflow will undertake a batch synchronisation.

The workflow context should be set-up using the `data.management.link.sdb.to.catalogue.rf` (Link Preservica to Catalogue) definition. The following settings should be used:

- **Name:** Synchronise Preservica to Catalogue
- **Description:** Synchronise Records to Adlib
- **Email Address:** Operator email as required (copy from existing workflow)
- **Notification on error:** Enabled
- **Notification on complete:** Disabled
- **Automatically Terminate:** Enabled

- **Allow Concurrent:** Disabled
- **Catalogue URL:** Adlib-API URL e.g. <https://adlib.preservica.com/api/wwwopac.ashx>
- **Catalogue Database:** collect.inf
- **Catalogue Username:** Adlib-API username as provided by customer
- **Catalogue Password:** Adlib-API password as provided by customer
- **Catalogue Mapping Document:** Select loaded mapping file by name i.e. *Adlib Mapping*
- **Number of Levels to Sync:** Leave blank (all records will be synchronised)
- **Pending Collection:** "Pending Synchronisation" collection or alternative provided by customer
- **Deliverable Unit Ref:** Place-holder only - leave blank.
- **Deliverable Unit Refs:** Place-holder only - leave blank.
- **Trigger:** Scheduled on a Daily basis out of office hours e.g. 01:00

4.4.6.2. Synchronise Catalogue to Preservica

This is an additional workflow that finds records within Adlib that are linked to Preservica but have since changed and updates Preservica to reflect the changes.

The workflow context should be set-up using the `data.management.sync.catalogue.to.sdb.rf` (Synchronise Catalogue to Preservica) definition. The following settings should be used:

- **Name:** Synchronise Catalogue to Preservica
- **Description:** Synchronises Changed Adlib Records to Preservica
- **Email Address:** Operator email as required (copy from existing workflow)
- **Notification on error:** Enabled
- **Notification on complete:** Disabled
- **Automatically Terminate:** Enabled
- **Allow Concurrent:** Disabled
- **Catalogue URL:** Adlib-API URL e.g. <https://adlib.preservica.com/api/wwwopac.ashx>
- **Catalogue Database:** collect.inf
- **Catalogue Username:** Adlib-API username as provided by customer
- **Catalogue Password:** Adlib-API password as provided by customer
- **Catalogue Mapping Document:** Select loaded mapping file by name i.e. *Adlib Mapping*
- **Re Sync All Changes:** Disabled (required if change to mapping file only)
- **eMail Address:** User email as provided by the customer
- **Tolerance Seconds:** 600 (This prevents missing updates due to minor time differences between servers)
- **Catalogue IDs Filename:** Leave blank (only applicable to HES)
- **Trigger:** Scheduled on a Daily basis out of office hours e.g. 02:00

4.4.6.3. Report Deleted Catalogue Records

This is an additional workflow that checks for records that exist in Preservica but link to a non-existent record (ie a deleted record) within Adlib. This workflow is run on a daily basis and produces a report that is eMailed to a nominated eMail address.

The workflow context should be set-up using the `data.management.report.deleted.catalogue.records.rf` (Report Deleted Catalogue Records) definition. The following settings should be used:

- **Name:** Report Deleted Catalogue Records
- **Description:** Report Catalogue Records deleted within Adlib
- **Email Address:** Operator email as required (copy from existing workflow)
- **Notification on error:** Enabled
- **Notification on complete:** Disabled
- **Automatically Terminate:** Enabled
- **Allow Concurrent:** Disabled
- **Report Name:** Select loaded report by name i.e. Missing Catalogue Records
- **Report Format:** Select required report format, typically PDF
- **Email Address:** User email as provided by the customer
- **Catalogue URL:** Adlib-API URL e.g. `https://adlib.preservica.com/api/wwwopac.ashx`
- **Catalogue Database:** `collect.inf`
- **Catalogue Username:** Adlib-API username as provided by customer
- **Catalogue Password:** Adlib-API password as provided by customer
- **Catalogue Mapping Document:** Select loaded mapping file by name i.e. Adlib Mapping
- **Trigger:** Scheduled on a Daily basis out of office hours e.g. 03:00

4.4.6.4. Auto Ingest and Synchronise to Catalogue (EE/SE Only)

This workflow replaces the standard Auto Ingest workflow and will attempt to immediately synchronise the ingested records to Adlib. If the synchronisation fails, the nightly **Synchronise Preservica to Catalogue** workflow will synchronise these records overnight. The existing Auto Ingest workflow should be deleted once the new workflow is configured.

The workflow context should be set-up using the `ingest.single.auto.standard.catalogue.rf` (Standard Ingest Workflow (Auto Selection, Catalogue sync)) definition. The following settings should be used:

- **Name:** As required (copy from existing Auto Ingest workflow)
- **Description:** As required (copy from existing Auto Ingest workflow)
- **Email Address:** Operator email as required (copy from existing workflow)
- **Notification on error:** Enabled
- **Notification on complete:** Disabled

- **Automatically Terminate:** Enabled
- **Allow Concurrent:** Enabled
- **SIP Location:** As required (copy from existing Auto Ingest workflow)
- **Collection:** "Pending Synchronisation" collection or alternative provided by customer
- **Transfer Agreement:** As required (copy from existing Auto Ingest workflow)
- **Classification Document:** As required (copy from existing Auto Ingest workflow)
- **Classification Class:** As required (copy from existing Auto Ingest workflow)
- **Classification Tag:** As required (copy from existing Auto Ingest workflow)
- **Storage Policy:** As required (copy from existing Auto Ingest workflow)
- **Secondary Storage:** As required (copy from existing Auto Ingest workflow)
- **Delete Source:** As required (copy from existing Auto Ingest workflow)
- **Link To Catalogue Workflow:** "Synchronise Preservica to Catalogue" e.g. set to the name of the workflow context created in Section 4.4.6.1.
- **Trigger:** File system watcher

4.4.6.5. Auto Ingest and Synchronise to Catalogue (CE Only)

This workflow replaces the standard CE Auto Ingest workflow and will attempt to immediately synchronise the ingested records to Adlib. If the synchronisation fails, the nightly **Synchronise Preservica to Catalogue** workflow will synchronise these records overnight. The existing Auto Ingest workflow should be deleted once the new workflow is configured.

The workflow context should be set-up using the `ingest.ce.single.auto.standard.catalogue.rf` (Standard Ingest Workflow (Auto Selection, Catalogue sync)) definition. The following settings should be used:

- **Name:** As required (copy from existing Auto Ingest workflow)
- **Description:** As required (copy from existing Auto Ingest workflow)
- **Email Address:** Operator email as required (copy from existing workflow)
- **Notification on error:** Enabled
- **Notification on complete:** Disabled
- **Automatically Terminate:** Enabled
- **Allow Concurrent:** Enabled
- **S3 Bucket Name:** As required (copy from existing Auto Ingest workflow)
- **Collection:** "Pending Synchronisation" collection or alternative provided by customer
- **Transfer Agreement:** As required (copy from existing Auto Ingest workflow)
- **Classification Document:** As required (copy from existing Auto Ingest workflow)
- **Classification Class:** As required (copy from existing Auto Ingest workflow)

-
- **Classification Tag:** As required (copy from existing Auto Ingest workflow)
 - **Storage Policy:** As required (copy from existing Auto Ingest workflow)
 - **Secondary Storage:** As required (copy from existing Auto Ingest workflow)
 - **Link To Catalogue Workflow:** "Synchronise Preservica to Catalogue" e.g. set to the name of the workflow context created in Section 4.4.6.1.
 - **Trigger:** Start Ingest Automatically

Chapter 5. CALM Standard Deployment

5.1. CALM and CALM-API Configuration

Preservica synchronisation with CALM is currently supported for CALM versions 9.2 through to 10. For Preservica to be able to synchronise with CALM, CALM-API must be installed by the customer. CALM-API is a separate Axiell-produced module that typically runs on the same server as CALM using IIS. Many clients have an Axiell-hosted CALM server that includes CALM-API. CALM can be installed with the standard instructions. However, the customer should make the following changes to CALM-API:

- CALM-API/IIS should be configured to only allow connections with a user name and password (by default anonymous authentication is enabled so this should be disabled).
- CALM-API/IIS should be configured to run using HTTPS only (by default it is configured to use unencrypted HTTP).
- The firewall in front of CALM-API should be set up to only allow incoming connections from Preservica's JobQueue.
- By default only 'open' records can be accessed via CALM-API. To be able to access CALM records with other access statuses the CALM-API DatabaseList.xml file needs to be modified. Assuming CALM-API is installed in C:\inetpub\wwwroot\CalmAPI, then this file is at C:\inetpub\wwwroot\CalmAPI\DatabaseList.xml. To enable access, replace the line:

```
<SearchFilter>(AccessStatus=open) & (CatalogueStatus=catalogued)</SearchFilter>
```

With the following line which has the filter removed.

```
<SearchFilter></SearchFilter>
```

CALM-API needs to be restarted following this change.

5.2. CALM Data Checks

In theory, the CALM *RefNo* and *RecordID* fields should be unique across the entire CALM catalogue. However, in practice, issues in earlier versions of CALM have led to multiple records having the same id values. Prior to synchronisation with Preservica we recommend that all duplicates are resolved since Preservica needs to be able uniquely retrieve records by their ids. Axiell are able to help with this removal of duplicates.

5.3. Information Required

The following information is required to configure Preservica to link to CALM.

- eMail address for a single user or eMail group that will receive the Illegal Catalogue Moves email and Missing Catalogue Records Report.
- URL of the CALM-API REST interface e.g. <https://calm.preservica.com/CalmAPI/Archive/Catalogue.asmx>. By default the CALM-API REST interface runs on port 80 so if using HTTPS the port is likely to be another port such as 443.
- Username for a valid CALM-API user. Note, this is likely to be a different user to the ones that can be used to log on to the CALM client. Creating a dedicated user for the CALM-API REST interface to use (eg preservica) is recommended.

- Password for the above CALM-API user account provided.

Without the user account details we will be unable to fully test access from Preservica to CALM-API and will not be able to complete the configuration of Preservica.

In addition, a top level collection to hold records awaiting synchronisation to CALM is required. We should inform customers that we will create a Top Level collection called "Pending Synchronisation" unless they inform us otherwise. In which case they should either give us an alternative name to create or create the collection and tell us the following information:

- Name
- Collection Code

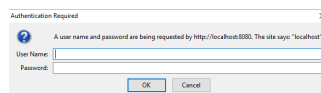
The customer will need to complete the Preservica CALM Synchronisation Configuration Document. This provides us with the necessary information to configure the synchronisation process. This document will take some time to complete. The higher level information requested above should be provided first to allow us to complete the initial checks.

5.4. Initial Checks

5.4.1. Check API Access

Once the CALM-API information has been provided a number of checks should be made to ensure that the Preservica system can successfully connect to Calm-API.

Initially the CALM-API URL and user details provided can be checked locally from a browser. A pop-up login dialogue should be displayed but this may fail if the customer has configured firewall rules to restrict access to the Preservica system only.



Connectivity should be checked from the Preservica JobQueue server using the wget application, for example.

```
# wget --user=admin --password=passwd  
https://calm.preservica.com/CalmAPI/Archive/Catalogue.asmx?WSDL
```

If the user credentials are correct, this should return the WSDL for the CALM-API SOAP web service, for example

```
<?xml version="1.0" encoding="utf-8"?>  
<wSDL:definitions xmlns:s="http://www.w3.org/2001/XMLSchema"  
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"  
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"  
  xmlns:tns="http://www.axiell.co.uk/WebServices/CalmApi/Archive/  
Catalogue/"  
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"  
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"  
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"  
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"  
  targetNamespace="http://www.axiell.co.uk/WebServices/CalmApi/Archive/  
Catalogue/"
```

```
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
<wsdl:types>
  <s:schema elementFormDefault="qualified"
    targetNamespace="http://www.axiell.co.uk/WebServices/CalmApi/
Archive/Catalogue/">
    <s:element name="IsItemOrderable">
      ... etc ...
```

5.5. System Configuration

Once connectivity to CALM-API has been confirmed Preservica should be set-up with the standard CALM interface configuration. This will require a number of files to be loaded into the appropriate tenancy and a number of workflow contexts to be created.



All the XML files are renamed replacing Default with a short tenant name/identifier. This name is persisted if the file is downloaded from Preservica and should prevent bulk updates to files during the release process updating any files accidental.

5.5.1. XML Schemas

Rename the standard CALM schema `CALM_Default.xsd` to `CALM_<TENANT>.xsd` eg `CALM_DOR.xsd` and upload using the standard Preservica administration functionality. The following settings should be used:

- **Schema Name:** CALM
- **Description:** As required

5.5.2. XML Transforms

Rename the standard CALM metadata viewer `CALM-viewer_Default.xslt` to `CALM-viewer_<TENANT>.xslt` eg `CALM-viewer_DOR.xslt` and upload using the standard Preservica administration functionality. The following settings should be used:

- **Name:** CALM Viewer
- **XSD Input:** CALM (i.e. the name of the schema added earlier)
- **XSD Output:** XHTML
- **Purpose:** View

5.5.3. XML Documents

Rename the standard CALM mapping document `CALM-mapping_Default.xml` to `CALM-mapping_<TENANT>.xml` eg `CALM-mapping_DOR.xml` and upload using the standard Preservica administration functionality. The following settings should be used:

- **Name:** CALM Mapping
- **Context:** Configuration File

Rename the standard CALM custom index document `CALM-search_Default.xml` to `CALM-search_<TENANT>.xml` eg `CALM-search_DOR.xml` and upload using the standard Preservica administration functionality. The following settings should be used:

- **Name:** CALM Indexer
- **Context:** Custom Index Definition

5.5.4. Reports

If not already loaded the standard report `MissingCatalogueRecords.jrxml` (Missing Catalogue Records) should be uploaded using the standard Preservica administration functionality.

5.5.5. Workflow Definitions

The following workflow definitions should be uploaded using the standard Preservica administration functionality.

All Preservica Editions:

- `data.management.link.sdb.to.catalogue.rf`
- `data.management.sync.catalogue.to.sdb.rf`
- `data.management.report.deleted.catalogue.records.rf`



Deletion from Preservica with immediate synchronisation to CALM is not supported.

Preservica CE Only:

- `ingest.ce.single.auto.standard.catalogue.rf`

Preservica EE/SE Only:

- `ingest.single.auto.standard.catalogue.rf`



Ingest with normalisation and immediate synchronisation to CALM is not supported.

5.5.6. Workflow Contexts

The following workflow contexts should be created using the standard Preservica administration functionality.

The recommended order for the nightly scheduled workflows is:

1. Synchronise Preservica to Catalogue
2. Synchronise Catalogue to Preservica
3. Report Deleted Catalogue Records

This is reflected in the scheduled times suggested below.

5.5.6.1. Synchronise Preservica to Catalogue

This is an additional workflow that finds records within Preservica that have not yet been synchronised to CALM and updates CALM with Preservica metadata and vice-versa. If records are being synchronised on ingest this workflow will deal with outstanding records e.g. where the CALM was not available during ingest. If records are not being synchronised on ingest this workflow will undertake a batch synchronisation.

The workflow context should be set-up using the `data.management.link.sdb.to.catalogue.rf` (Link Preservica to Catalogue) definition. The following settings should be used:

- **Name:** Synchronise Preservica to Catalogue

- **Description:** Synchronise Records to CALM
- **Email Address:** Operator email as required (copy from existing workflow)
- **Notification on error:** Enabled
- **Notification on complete:** Disabled
- **Automatically Terminate:** Enabled
- **Allow Concurrent:** Disabled
- **Catalogue URL:** CALM-API URL e.g. <https://calm.preservica.com/CalmAPI/Archive/Catalogue.asmx?WSDL>
- **Catalogue Database:** collect.inf
- **Catalogue Username:** CALM-API username as provided by customer
- **Catalogue Password:** CALM-API password as provided by customer
- **Catalogue Mapping Document:** Select loaded mapping file by name i.e. CALM Mapping
- **Number of Levels to Sync:** Leave blank (all records will be synchronised)
- **Pending Collection:** "Pending Synchronisation" collection or alternative provided by customer
- **Deliverable Unit Ref:** Place-holder only - leave blank.
- **Deliverable Unit Refs:** Place-holder only - leave blank.
- **Trigger:** Scheduled on a Daily basis out of office hours e.g. 01:00

5.5.6.2. Synchronise Catalogue to Preservica

This is an additional workflow that finds records within CALM that are linked to Preservica but have since changed and updates Preservica to reflect the changes.

The workflow context should be set-up using the `data.management.sync.catalogue.to.sdb.rf` (Synchronise Catalogue to Preservica) definition. The following settings should be used:

- **Name:** Synchronise Catalogue to Preservica
- **Description:** Synchronises Changed CALM Records to Preservica
- **Email Address:** Operator email as required (copy from existing workflow)
- **Notification on error:** Enabled
- **Notification on complete:** Disabled
- **Automatically Terminate:** Enabled
- **Allow Concurrent:** Disabled
- **Catalogue URL:** CALM-API URL e.g. <https://calm.preservica.com/CalmAPI/Archive/Catalogue.asmx?WSDL>
- **Catalogue Database:** collect.inf
- **Catalogue Username:** CALM-API username as provided by customer
- **Catalogue Password:** CALM-API password as provided by customer

- **Catalogue Mapping Document:** Select loaded mapping file by name i.e. `CALM Mapping`
- **Re Sync All Changes:** Disabled (required if change to mapping file only)
- **eMail Address:** User email as provided by the customer
- **Tolerance Seconds:** 600 (This prevents missing updates due to minor time differences between servers)
- **Catalogue IDs Filename:** Leave blank (only applicable to HES)
- **Trigger:** Scheduled on a Daily basis out of office hours e.g. 02:00

5.5.6.3. Report Deleted Catalogue Records

This is an additional workflow that checks for records that exist in Preservica but link to a non-existent record (i.e. a deleted record) within CALM. This workflow is run on a daily basis and produces a report that is eMailed to a nominated eMail address.

The workflow context should be set-up using the `data.management.report.deleted.catalogue.records.rf` (Report Deleted Catalogue Records) definition. The following settings should be used:

- **Name:** Report Deleted Catalogue Records
- **Description:** Report Catalogue Records deleted within CALM
- **Email Address:** Operator email as required (copy from existing workflow)
- **Notification on error:** Enabled
- **Notification on complete:** Disabled
- **Automatically Terminate:** Enabled
- **Allow Concurrent:** Disabled
- **Report Name:** Select loaded report by name i.e. `Missing Catalogue Records`
- **Report Format:** Select required report format, typically PDF
- **Email Address:** User email as provided by the customer
- **Catalogue URL:** CALM-API URL e.g. `https://calm.preservica.com/CalmAPI/Archive/Catalogue.aspx?WSDL`
- **Catalogue Database:** `collect.inf`
- **Catalogue Username:** CALM-API username as provided by customer
- **Catalogue Password:** CALM-API password as provided by customer
- **Catalogue Mapping Document:** Select loaded mapping file by name i.e. `CALM Mapping`
- **Trigger:** Scheduled on a Daily basis out of office hours e.g. 03:00

5.5.6.4. Auto Ingest and Synchronise to Catalogue (EE/SE Only)

This workflow replaces the standard Auto Ingest workflow and will attempt to immediately synchronise the ingested records to CALM. If the synchronisation fails, the nightly **Synchronise Preservica to Catalogue** workflow will synchronise these records overnight. The existing Auto Ingest workflow should be deleted once the new workflow is configured.

The workflow context should be set-up using the `ingest.single.auto.standard.catalogue.rf` (Standard Ingest Workflow (Auto Selection, Catalogue sync)) definition. The following settings should be used:

- **Name:** As required (copy from existing Auto Ingest workflow)
- **Description:** As required (copy from existing Auto Ingest workflow)
- **Email Address:** Operator email as required (copy from existing workflow)
- **Notification on error:** Enabled
- **Notification on complete:** Disabled
- **Automatically Terminate:** Enabled
- **Allow Concurrent:** Enabled
- **SIP Location:** As required (copy from existing Auto Ingest workflow)
- **Collection:** "Pending Synchronisation" collection or alternative provided by customer
- **Transfer Agreement:** As required (copy from existing Auto Ingest workflow)
- **Classification Document:** As required (copy from existing Auto Ingest workflow)
- **Classification Class:** As required (copy from existing Auto Ingest workflow)
- **Classification Tag:** As required (copy from existing Auto Ingest workflow)
- **Storage Policy:** As required (copy from existing Auto Ingest workflow)
- **Secondary Storage:** As required (copy from existing Auto Ingest workflow)
- **Delete Source:** As required (copy from existing Auto Ingest workflow)
- **Link To Catalogue Workflow:** "Synchronise Preservica to Catalogue" e.g. set to the name of the workflow context created in Section 5.5.6.1.
- **Trigger:** File system watcher

5.5.6.5. Auto Ingest and Synchronise to Catalogue (CE Only)

This workflow replaces the standard CE Auto Ingest workflow and will attempt to immediately synchronise the ingested records to CALM. If the synchronisation fails, the nightly **Synchronise Preservica to Catalogue** workflow will synchronise these records overnight. The existing Auto Ingest workflow should be deleted once the new workflow is configured.

The workflow context should be set-up using the `ingest.ce.single.auto.standard.catalogue.rf` (Standard Ingest Workflow (Auto Selection, Catalogue sync)) definition. The following settings should be used:

- **Name:** As required (copy from existing Auto Ingest workflow)
- **Description:** As required (copy from existing Auto Ingest workflow)
- **Email Address:** Operator email as required (copy from existing workflow)
- **Notification on error:** Enabled
- **Notification on complete:** Disabled

-
- **Automatically Terminate:** Enabled
 - **Allow Concurrent:** Enabled
 - **S3 Bucket Name:** As required (copy from existing Auto Ingest workflow)
 - **Collection:** "Pending Synchronisation" collection or alternative provided by customer
 - **Transfer Agreement:** As required (copy from existing Auto Ingest workflow)
 - **Classification Document:** As required (copy from existing Auto Ingest workflow)
 - **Classification Class:** As required (copy from existing Auto Ingest workflow)
 - **Classification Tag:** As required (copy from existing Auto Ingest workflow)
 - **Storage Policy:** As required (copy from existing Auto Ingest workflow)
 - **Secondary Storage:** As required (copy from existing Auto Ingest workflow)
 - **Link To Catalogue Workflow:** "Synchronise Preservica to Catalogue" e.g. set to the name of the workflow context created in Section 5.5.6.1.
 - **Trigger:** Start Ingest Automatically

Chapter 6. Standalone Upgrade Tools

6.1. Migration of Thumbnails from Universal Access

An optional tool for customers upgrading from versions before 5.8. This tool will migrate thumbnails which had been manually or automatically assigned for deliverable units and collections in older versions of Universal Access, and copy the information and the thumbnail into the Preservica database, so those thumbnails will once again be visible in UA, but also in Explorer. Thumbnails will be saved into the usual Preservica location, so ensure that the thumbnailing settings for the tenancy in Preservica (through the Collection Properties page) are set up correctly. In particular, if you're using disk storage for thumbnails, make sure the target network share is accessible from where you will be running the tool.

The tool is provided as a ZIP file, containing start scripts, the application JAR and the required dependencies. This ZIP can be provided to customers with a briefing note explaining the operation of the tool, or used by Ops to do the migration for them. The tool should be run with `copy_thumbnails.bat` (Windows) or `copy_thumbnails.sh` (Linux).

It should be run on the machine which is running Universal Access (on a default deployment, this will be the access server), and that machine needs access to the Preservica database and to the designated thumbnail storage (likely to be a disk location, for EE). On a default deployment this will already be the case because the API endpoints that UA requests data from have the same requirements.

The simplest invocation of the tool is:

```
copy_thumbnails -t __TENANT_NAME__
```

The tenant name is compulsory to make sure the data is being compared to the right system. The tool will look for both UA and Preservica settings (local/features.properties) in their default locations. If either is installed in a non-default location, you can use `-c` and `-p` options respectively to locate them.

In a multi-site Wordpress install, you also need to provide the `-w` option to specify the site prefix for the Wordpress database tables. Wordpress site prefixes are typically a number, which is the Blog ID (accessible from the admin panel).

By default, only thumbnails which were uploaded into UA as files will be copied into the Preservica database. To include thumbnail links as well (which includes any that were assigned by UA's auto-thumbnailing mechanism, as well as links manually created by a UA admin), use the `-l` option.

6.1.1. Results

Assuming the tool can connect to all its data environments, it will process each uploaded thumbnail in the UA database and, if `-l` is specified, the linked thumbnails. A log entry will be written to the log file (`logs/thumbnail-copier.log`) for each thumbnail. Warnings will be logged, both in the file and on the console, for any thumbnails that can't be copied.

If an item already has a thumbnail manually assigned in the Preservica database (including an explicit choice of None), then it will not be overwritten by an assignment from UA. Automatically assigned thumbnails for deliverable units will be overwritten, if there is a thumbnail for that object in UA.

Chapter 7. AWS Bulk Upload Service

7.1. Overview

The AWS Snowball is an encrypted, aggrandized, high capacity network attached storage device which AWS provide as a means of bulk upload into their cloud based storage.

Preservica provides a chargeable bulk ingest service based on the AWS Snowball. This has replaced the previous disk based solution.

The AWS Bulk Upload Service is described from a customer perspective in the Preservica AWS Bulk Upload Service [ABU].

This document details the activities that need to be undertaken by Preservica to enable a Customer to use the Bulk Upload Service. At a high level these tasks are:

- Review the Volumetrics and DROID profile(s) as provided by the customer to estimate how long the content will take to ingest. and determine the number job queues required. Using same calculator used by the Sales team.
- Ensure that the appropriate contract is in place and a confirmed order has been placed by the customer
- Confirm that the Customer has provided the name of the **Source** Upload Area into which the content of the Snowball will initially be copied. Check that the corresponding S3 Bucket exists and is empty.
- Confirm that the Customer has provided details of the **Collection** into which the uploaded content should be ingested and check that the Collection exists and is empty.
- Confirm the customer contact details and dates for delivery of the AWS Snowball
- Order the AWS Snowball within the Customer AWS account using the AWS Console
- Configure additional dedicated JobQueue nodes for the Bulk ingest
- Check that underlying AWS database has sufficient free space to allow the bulk upload to complete. As a rule of thumb 1GB of database space should be allowed per TB of content.
- Create the two workflows necessary to ingest the uploaded content
- Monitor the ingest process, where possible resolve any issues and notify the customer once complete
- Disable, Shut-down and (in appropriate) Delete any additional JobQueue nodes created for the Bulk Upload.
- Create and file a summary report. Profile of content, resources used, duration, issues. Send copy of this to the Product Manager
- Confirm successful upload with the customer. Seek approval to clear the holding area from the Operations director and then delete the uploaded files from the holding area as this is not done as part of the automated process

In addition, the Customer is likely to have a number of questions and/or problems that will need to be resolved. This process deals with the ingest of large volume of customer data. Extreme care and caution should be excised at every step. As a basic principle an issues or any doubt then ask.

7.2. Review Volumetrics

<TODO>

7.3. Check the Ingest Bucket

The customer should have provided details of (empty) ingest bucket to use for the Bulk Upload

Check that this bucket exists in the customer tenancy (Administration > Sources) and that it is an upload area. Also login to the AWS console and check that the bucket exists and is empty

7.4. Check the Ingest Collection

The customer should have provided details of (empty) Collection to use for the Bulk Upload

Check that this collection exists and is empty. The collection GUID will later be required to configure the Bulk Upload.

7.5. Order the AWS Snowball (Mandatory)

<TODO>

7.6. Create the Necessary Workflows (Mandatory)

The following Workflow Definitions must be loaded into the Customer tenancy within Preservica unless they have already been uploaded as a result of a previous Bulk Upload.

This can be done using standard Preservica functionality in Administration > Workflow Definitions.

- **ingest.ce.bulk.load.rf** - Bulk Ingest from Cloud Storage
(com.tessella.sdb.core.workflow.ingest.ce.bulk.load)
- **ingest.ce.bulk.segment.rf** - Bulk Ingest from Cloud Storage: Single Segment
(com.tessella.sdb.core.workflow.ingest.ce.bulk.segment)

The following Workflow Contexts must be created in the Customer tenancy within Preservica unless they have already been created as a result of a previous Bulk Upload.

This can be done using standard Preservica functionality in Ingest > Manage.

A Child Ingest workflow context should be set-up using the `ingest.ce.bulk.segment.rf` (Bulk Ingest from Cloud Storage: Single Segment) definition. The following settings should be used:

- **Name:** As required e.g. Bulk Upload Single Segment
- **Description:** As required e.g. Accrual of files from single folder
- **Email Address:** Operator email as required (ideally operations@preservica.com [mailto:operations@preservica.com])
- **Notification on error:** Enabled
- **Notification on complete:** Disabled
- **Automatically Terminate:** Enabled
- **Allow Concurrent:** Enabled
- **Max Files:** Maximum number of files to accrue in a single workflow instance (Default 100000)
- **Max Size:** Maximum total size of files (in GB) to accrue in a single workflow instance (Default 200)
- **Storage Policy Document:** As required
- **Trigger:** Manual only



Only a single Ingest Workflow Context using the `ingest.ce.bulk.segment.rf` definition should be active within the Customer tenancy.

A Parent Ingest workflow context should be set-up using the `ingest.ce.bulk.load.rf` (Bulk Ingest from Cloud Storage: Single Segment) definition. The following settings should be used:

- **Name:** As required e.g. Bulk Upload Control
- **Description:** As required e.g. Control workflow for Bulk Upload
- **Email Address:** Operator email as required (ideally `operations@preservica.com` [`mailto:operations@preservica.com`])
- **Notification on error:** Enabled
- **Notification on complete:** Enabled
- **Automatically Terminate:** Enabled
- **Allow Concurrent:** Disabled
- **S3 Bucket Name:** Select name of the Bulk Upload S3 Bucket as created in section Section 7.5
- **Collection:** Enter GUID of parent collection for the Bulk Upload (Name will auto populate)
- **Security Tag:** Select required security tag for ingested content
- **Digital Surrogate:** As required
- **Recovery Mode:** Normally Disabled; see section Section 7.9 for use of this setting
- **Trigger:** Manual only

It is not necessary to specify the Child Ingest Workflow Context as this is automatically selected.

7.7. Provision Additional JobQueue Nodes (Recommended)

Typically, additional JobQueue nodes will be required. This is to ensure that other users are not adversely impacted and complete the ingest of the uploaded content relatively quickly thus minimising storage costs. Clearly a balance exists between the storage costs and the cost of additional JobQueue nodes so the number of additional JobQueue nodes required will need to be assessed on a case by case basis.

Additional JobQueue nodes (AWS instances) should be created using standard AWS functionality. In addition, it may be necessary to amend the Apache httpd configuration on the main application server to route connections to the new nodes this can be done by editing the file:

```
/opt/preservica/conf/httpd/httpd-https.conf
```

It is recommended that the Bulk Upload Ingest Workflow Instances are restricted to these additional JobQueue nodes. To do this the Preservica workflow engine must be configured to use a specific load balancer. This can be done using standard Preservica functionality in Administration > System Properties. The **System Property** `job.queue.load.balancer.bean` should be set to a value of `priorityValueJobQueueLoadBalancer`

Once the additional nodes are available they should be added to Preservica using standard Preservica functionality in Administration > JobQueue. In addition to setting the **Client URL** in the normal manner the **tag** must be set to `BulkIngest`

7.8. Run the Ingest Process (Mandatory)

Once confirmation has been received from Amazon that the copy of the Customer files from the Snowball to the ingest bucket has completed a basic check on the ingest bucket should be made using either the AWS

console, a third party tool like CloudBerry or the AWS S3 Command Line. The number of files and total size the files should be confirmed.

Once this confirmation step has been completed any additional JobQueue nodes should be activated and the ingest process started. The ingest is started by running a single instance of the Parent Ingest workflow context created in section Section 7.6 using standard Preservica functionality in Ingest > Start.

The Parent Ingest Workflow will:

- Validate the specified collection is present, accessible and empty
- Create a Deliverable Unit hierarchy within the specified collection to mirror the original folder structure
- Start a Child Ingest Workflow for each Deliverable Unit (folder) that contains files
- Validate that the number of files within each Deliverable Unit (folder) matches the number expected

Each Child Ingest Workflow will correspond to a single folder in the uploaded data and will accrue the files into the corresponding Deliverable Unit in Preservica. For performance reasons this accrual will be chunked for very large folder with each individual accrual being limited to either the number of files or total file size (whichever results in the smaller accrual) as specified in the Child Ingest Workflow Context. If a single file exceeds the specified size limit it will be ingested as a single file accrual.

7.9. Problems and Recovery (Optional)

<TODO>

7.10. Post Ingest Tidy Up (Mandatory)

The Ingest Workflow Contexts created in section Section 7.6 must be deactivated and deleted. This can be done using standard Preservica functionality in Ingest > Manage.

The Ingest Workflow Definitions uploaded in section Section 7.6 must be deactivated. This can be done using standard Preservica functionality in Administration > Workflow Definitions.

Any additional JobQueue nodes created for the Bulk Upload process (see section Section 7.7) must be disabled within Preservica, shutdown and possibly deleted from AWS.

Once the Customer has validated the Bulk Upload and approval has been obtained from the Operations Director the contents of the S3 Bulk Upload Bucket must be manually deleted using either the AWS console, a third party tool like CloudBerry or the AWS S3 Command Line.

Chapter 8. Instance Migration

8.1. Overview

Preservica provides a migration service that will allow customers to move the data in their Preservica system between separate instances of Preservica.

The following migration paths are supported.

- Preservica CE to Preservica CE in a different geographical region **NOT CURRENTLY SUPPORTED IN V6**
- Preservica CE to Preservica CE Gov Cloud **NOT CURRENTLY SUPPORTED IN V6**
- Preservica CE to Preservica EPC(P)
- Preservica EE to Preservica EPC(P)

8.1.1. Limitations

The Preservica version (e.g. 5.4.0) cannot be changed as part of the migration process. If you need to do this Preservica must be separately upgraded before or after the migration using the standard upgrade approach. Please Note: downgrading of the Preservica version is not supported.

The migration service only supports migration of system based on the MySQL database. This applies to both the source and target instances. All Preservica CE and EPC(P) instances use the MySQL database.

8.1.2. Scope of the Migration Service

The following information will be migrated:

- The Preservica database (metadata store). This includes the hierarchy of the archive, the descriptive metadata of the objects, the full audit trail plus all workflow and administrative settings related to your archive.
- The search index used by Preservica to prevent the need to re-create this index which would take considerable time depending on the size of the archive.
- Any file preview files and thumbnails used within Explorer again this is to prevent the need to rebuild this information.
- Prepared DIP download packages and the results of trial Preservation migrations again this is to prevent the need to rebuild this information.
- The digital files stored within the archive (Bulk Storage).
- User details stored within User Management where the Preservica User Management module is used in both the source and target instances.

Please note the following components will NOT be migrated:

- Settings and configuration within the Universal Access module. We are unable to migrate the limited amount of configuration information within the Universal Access module due to the nature of the underlying WordPress database structure. Where Universal Access is in use a new, empty module will be provided within the target instance.
- System wide configuration settings as these relate to the Preservica instance.
- Unprocessed files in Ingest areas
- System log files

- Any data held relating to storage media, specifically data held in the database tables `storage_media` and `storage_media_locations`. These tables were made available to specific Preservica EE customers to enable them to develop custom extensions to Preservica. Preservica CE and ECP(P) do not make any use of these tables.
- Any data relating to state of running workflows or active users.

8.1.3. Migration Timeline

Day numbers are indicative only. Export/Import of the Bulk Storage and Storage Synchronisation are dependent on the volume of content and may take multiple days to complete.

Day 0:

- Customer ensures any administrator accounts also have the normal "user" roles assigned

Day 1:

- User account details are exported
- Administrator permissions are removed from all users
- All workflow contexts are disabled to prevent new workflows being started
- Any running workflows are allowed to complete or terminated

Day 2:

- Preservica database is copied
- Update and Delete permissions are removed from all user accounts (system is now read only)
- Database is exported
- Search index is exported
- Contents of the Bulk Storage is exported (not applicable for migration from CE)
- Contents of the preview cache is exported
- Contents of the download area (DIP and trial migrations) is exported

Day 3:

- Preservica database is imported
- Search index is imported
- Contents of the preview cache is imported
- Contents of the download area (DIP and trial migrations) is imported
- System is configured to use existing bulk storage (migration from CE only)
- Contents of bulk storage are imported (migration from SE only)
- User account details are imported or new account created as appropriate

Day 4 on:

- Storage synchronisation used to move content to new storage locations (migration from CE and potentially SE).

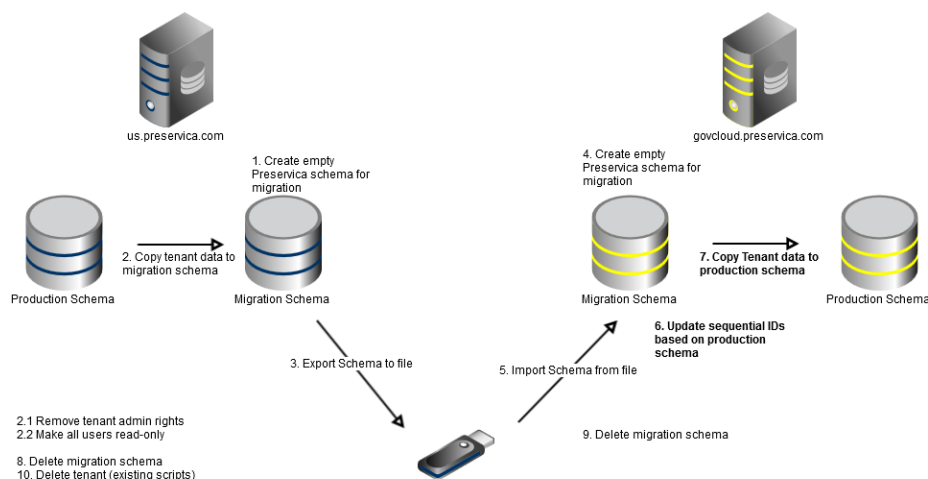
Day x:

- Customer data deleted from source instance

8.2. Migration of System Components

8.2.1. Metadata Store

Although migration of bulk storage will take longer to complete the database migration is the most complex part of the migration process. The overview of the process is shown in Figure 1.



The following steps are required.

On the source instance:

1. Create a cut down "migration" schema within the source database instance
2. Copy the specific customers (tenant) data to the migration schema
3. Export the migration schema to file

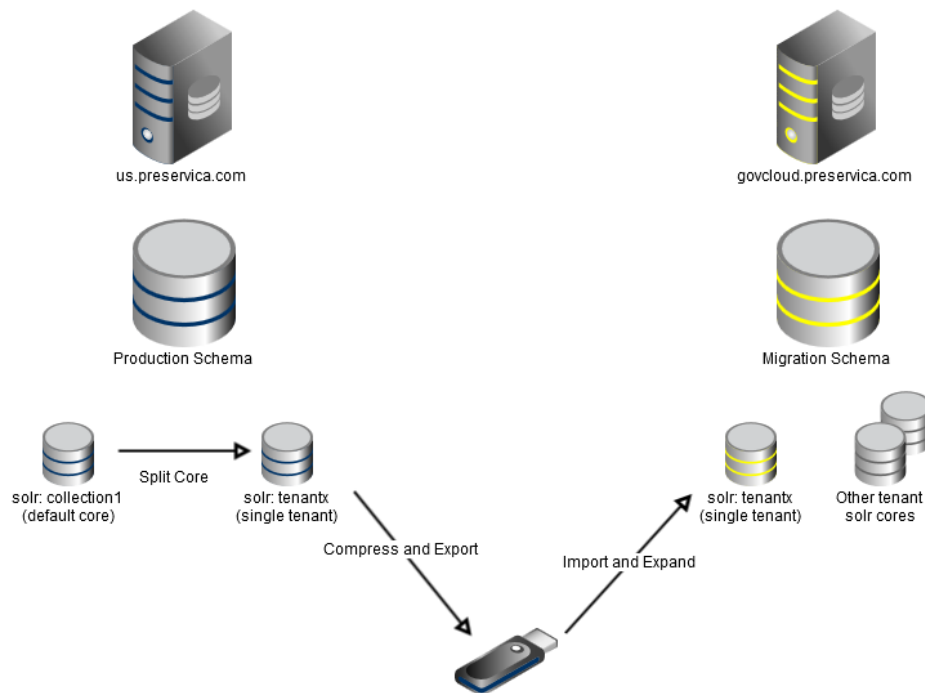
On the target instance:

4. Import the migration schema creating a new migration schema
5. Prepare the migration schema for the migration process
6. Shut down the target Preservica instance
7. Amend the record identifiers within the migration schema to avoid conflict with data already in the target database instance
8. Merge the migrated data into the target database instance
9. Restart the target Preservica instance. NB no access to the migrated data is available at this stage as user accounts have not been created.
10. Delete the migration schema

On the source instance:

11. Delete the migrated customer (tenant) data

8.2.2. Solr Search Index



The following steps are required.

On the source instance:

1. Compress and export the files comprising the solr core

On the target instance:

3. Import the solr core
4. Start or re-start the Preservica system.

On the source instance:

5. Delete the customer (tenant) specific solr core.

8.2.3. User Accounts

The following steps are required.

Preparation on the source instance:

1. Export the User Management accounts for the specific customer (tenant)
2. Update the User Management database to remove the following roles from all users for the specific customer:
 - a. SDB_MANGER_USER
 - b. SDB_ADMIN_USER
 - c. SDB_REGISTRY_ADMIN_USER

On the source instance preparatory:

3. Update the metadata store (database) to remove create, update and delete permissions from all users for the specific customer

On the target instance:

4. Import the customer User Management accounts

8.2.4. Bulk Storage

Copying of the bulk, digital file storage is likely to be very time consuming and depends directly on the size of the archive. A number of options for this process exist and advice should be sought from Preservica support on a case by case basis. At a high level the available options, in order of preference are:

1. Leave the existing storage in the same physical location and access it from the new instance
2. Synchronise content files between storage adapters using standard Preservica functionality. This will require one or more of the adapters on the source system to be connected to the target system (as per 1) on a temporary basis, while content is synchronised.
3. Physically copy the contents of the storage adapter(s) to the required locations for the target system. The file structure within the adapter **MUST** be preserved.

A new adapter, native to the target instance, could also be connected to the source instance on a temporary basis and synchronised before migration but this process is likely to be harder to control.

Where multiple adapters are in use these can be used to provide redundancy in the bulk storage or can be used with storage policies to manage the location in which specific files are stored. In these cases great care must be taken to ensure that at least one copy of each digital file is migrated. The recommended approach is to exactly replicate the source storage adapters and undertake any rationalisation of the storage as a separate activity.

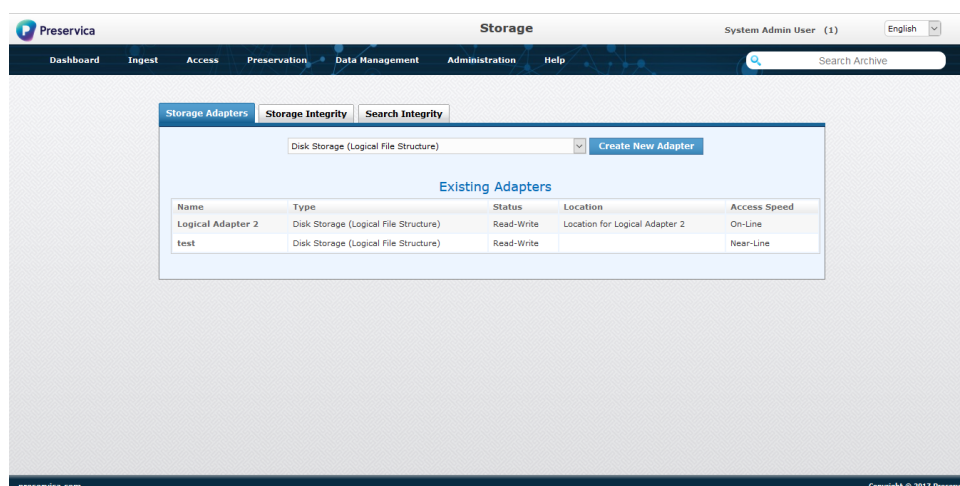
8.2.4.1. Amazon S3 Storage

Details of the existing adapters will be migrated to the target system. Due to the cloud based nature of these adapters the content of these adapters should be fully accessible from the target system. It may be necessary to migrate the files to new S3 storage adapters owned by a different Amazon account. This should be done post migration using standard Preservica functionality.

8.2.4.2. On-Site (file based) Storage

Details of the existing adapters will be migrated to the target system. If it is necessary to change the physical location of the on-site storage the contents of the storage adapters should be physically copied to the new location once the source instances are in read-only mode.

The location of the file based storage is detailed in the storage adapter settings:



The file structure of the adapter under the Volume Path must not be changed. If necessary the volume path can be changed to reflect the new location of the adapter.

8.2.5. Preview Image Cache

The location of the cache of preview images is defined in Administration > Collection Properties. This setting will be copied over on migration and should be updated as necessary. In principle the existing preview images can be copied to the target instance observing the same procedure as outlined above for the storage adapters. Preservica will however automatically re-create preview files as necessary so copying the existing files is not essential.

8.2.6. Generated DIPs

DIPs created by access workflow can be downloaded via the user interface. If the generated files are not migrated it will be necessary to re-run the export workflows to re-create the DIPs if they are required.

8.2.6.1. Amazon S3 Storage

Details of the location (S3 bucket) for generated DIPs is defined in Administration > Sources. The location is then referenced in the individual Access workflow contexts. These settings will be migrated. The following options exist:

1. If the S3 bucket is accessible by the source system the DIP files can be downloaded.
2. If the S3 bucket is not accessible by the source system a new bucket should be created and the workflow contexts updated. The DIP files can be manually copied to the new bucket or the DIPs marked as deleted.

8.2.6.2. On-Site (file based) Storage

Details of the location for generated DIPs is defined in Administration > Configuration. This settings will be migrated and should be amended as necessary. The generated DIP files should be manually copied to the new location.

8.2.7. Test Migrations

Test Migrations created by preservation workflow can be downloaded via the user interface. If the generated files are not migrated it will be necessary to re-run the preservation workflows to re-create the migrated packages if they are required.

8.2.7.1. Amazon S3 Storage

Details of the location (S3 bucket) for Test downloads is defined in Administration > Sources. The location is then referenced in the individual Preservation workflow contexts. These settings will be migrated. The following options exist:

1. If the S3 bucket is accessible by the source system the Test Migration files can be downloaded.
2. If the S3 bucket is not accessible by the source system a new bucket should be created and the workflow contexts updated. The Test Migration files can be manually copied to the new bucket or new test migrations run as required.

8.2.7.2. On-Site (file based) Storage

Details of the location for generated Test Migrations is based on the DIP Download location defined in Administration > Configuration. This settings will be migrated and should be amended as necessary. The generated Test Migration files should be manually copied to the new location.

8.3. Migration Process

8.3.1. Configuration

8.3.1.1. Database Creation Scripts

The standard database creation scripts are used to create the migration schema in the source instance.

Within the `..\Database\mysql` folder:

Configure the `database.properties` file as per the Preservica System Installation Guide [SIG]. The `mysql.database.sdb.username` should be set to the name of the temporary schema to be used for the migration eg `migration`. The associated password should be set in `mysql.database.sdb.password`. No reporting users are created so these values (`mysql.database.reporting...`) do not need to be set.

8.3.1.2. Migration Scripts

The migration scripts are in the `..\TenancyManagement\TenancyMigrate` project.

Configure the `database.properties` file as detailed below:

```
# Tenant to migrate. Must be present in the source instance and NOT in the
target instance
# tenant id must be exactly as it appears in LDAP, i.e. use capitals if
needed
tenant.name=TENANT_NAME

# Source Instance (Instance/database the Tenant is currently in)
mysql.database.source.host.name=sdb3.preservica.co.uk
mysql.database.source.host.port=3306
mysql.database.source.root.username=root
mysql.database.source.root.password=root
mysql.database.source.schema=source_schema
solr.url=http://localhost/solr
source.solr.home=C:/Preservica/EE/solr_home
source.solr.default.core.name=collection1
source.ldap.password=myPassword

# Target Instance (Instance/database the Tenant will end up in)
mysql.database.target.host.name=sdb3.preservica.co.uk
mysql.database.target.host.port=3306
mysql.database.target.root.username=root
mysql.database.target.root.password=root
mysql.database.target.schema=target_schema
target.solr.home=C:/Preservica/EE/solr_home
target.ldap.password=myPassword

#Liquibase Properties (needed for target schema only)
database.url=jdbc:mysql://${mysql.database.target.host.name}:
${mysql.database.target.host.port}/${mysql.database.migration.schema}?
serverTimezone=UTC
database.driver=com.mysql.jdbc.Driver
default.schema=${mysql.database.migration.schema}

# Migration schema - common between source and target
```

```
mysql.database.migration.schema=tenant_migration
mysql.database.migration.schema.user=tenant_migration
mysql.database.migration.schema.password=tenant_migration
```

Tenant to migrate:

`tenant.name` Name of the tenant to migrate
(case sensitive)

Source Instance details:

`mysql.database.source.host.name` Name of the MySQL database server

`mysql.database.source.host.port` Port the MySQL database is using default is 3306

`mysql.database.source.root.username` Administrator (root) username

`mysql.database.source.root.password` Password for the Administrative users

`mysql.database.source.schema` Name of the existing schema used by Preservica

`solr.url` URL to access solr, typically <http://localhost/solr>

`source.solr.home` Location of the solr_home directory

`source.solr.default.core.name` Name of the default solr core, typically collection1

`source.ldap.password` Password of the ldap-browser user

Target Instance details:

`mysql.database.target.host.name` Name of the MySQL database server

`mysql.database.target.host.port` Port the MySQL database is using default is 3306

`mysql.database.target.root.username` Administrator (root) username

`mysql.database.target.root.password` Password for the Administrative users

`mysql.database.target.schema` Name of the existing schema used by Preservica

`target.solr.home` Location of the solr_home directory

`target.ldap.password` Password of the ldap-browser user

Migration details:

`mysql.database.migration.schema` Name of the schema to be created for migration

`mysql.database.migration.schema.user` Name of the owner of the migration schema

mysql.database.schema.password Password of the schema owner

8.3.2. Source System

8.3.2.1. Create the Migration Schema

Within the `..\Database\mysql` folder:

Configure the `database.properties` file as per the Preservica System Installation Guide [SIG]. The `mysql.database.sdb.username` should be set to the name of the temporary schema to be used for the migration eg `migration`. The associated password should be set in `mysql.database.sdb.password`. No reporting users are created so these values (`mysql.database.reporting...`) do not need to be set.

Create the migration database using the following command:

```
ant create-4.0 create-v6-schemas apply-liquibase-diffs
```

8.3.2.2. Remove Manager Permissions

Within the `..\TenancyManagement\TenancyMigrate` folder:

Export the User Account data to file the following command:

```
ant remove-ldap-manger
```

This will remove the role `SDB_MANAGER_USER` from all accounts for the specified tenant.

Once this is done all workflow contexts should be manually disabled and running workflows allowed to complete.

8.3.2.3. Export User Management Data

Within the `..\TenancyManagement\TenancyMigrate` folder:

Export the User Account data to file the following command:

```
ant export-ldap
```

This will create the file:

```
..\TenancyManagement\TenancyMigrate\export\<tenant>_ldap_export.ldif
```

This file should be manually copied to the same location on the target instance

8.3.2.4. Copy Metadata

Within the `..\TenancyManagement\TenancyMigrate` folder:

Copy the customer (tenant) data to the migration schemas using the following command:

```
ant copy-tenant-data
```

8.3.2.5. Remove NULLTENANT Data

This step **MUST** be undertaken if the migration target is an existing, populated CE system. It **MUST NOT** be undertaken if the migration target is a new, empty EPC(P) system.

Within the `..\TenancyManagement\TenancyMigrate` folder:

Delete the NULLTENANT template security data the command:

```
ant delete-nulltenant-data
```

8.3.2.6. Export the Metadata (database)

Within the folder: `..\TenancyManagement\TenancyMigrate` folder:

Export the customer (tenant) data to files using the following command:

```
ant export-tenant-data
```

This will create several compressed SQL dump files containing the tenant:

```
..\TenancyManagement\TenancyMigrate\export\<tenant>_<schema>_export.zip
```

These files should be manually copied to the same location on the target instance

8.3.2.7. Export the Solr (index) data

Within the folder: `..\TenancyManagement\TenancyMigrate` folder:

Export the index data for the customer to a file using the following command:

```
ant export-solr-data
```

This will create a compressed file containing the tenant solr index:

```
..\TenancyManagement\TenancyMigrate\export\<tenant>_solr_export.zip
```

This file should be manually copied to the same location on the target instance

8.3.3. Target System

8.3.3.1. Create Migration Schema

Within the `..\Database\mysql` folder:

Configure the `database.properties` file as per the Preservica System Installation Guide [SIG]. The `mysql.database.sdb.username` should be set to the name of the temporary schema to be used for the migration eg migration. The associated password should be set in `mysql.database.sdb.password`. No reporting users are created so these values (`mysql.database.reporting...`) do not need to be set.

Create the migration database using the following command:

```
ant create-4.0 create-v6-schemas apply-liquibase-diffs
```

8.3.3.2. Import the Metadata (database) export

Within the `..\TenancyManagement\TenancyMigrate` folder:

Load the database dump using the command:

```
ant import-tenant-data
```

8.3.3.3. Prepare the migration schema

NOT CURRENTLY SUPPORTED IN V6

This step is **NOT** required if the migration target is a new, empty EPC(P) system.

Within the `..\TenancyManagement\TenancyMigrate` folder:

Add additional fields to the imported data:

```
ant prepare-schema
```

8.3.3.4. Import the Solr (index) data

Within the `..\TenancyManagement\TenancyMigrate` folder:

Import the Solr index data:

```
ant import-solr-data
```

8.3.3.5. Stop the Target Preservica Instance

At this point the target Preservica instance **MUST** be shutdown to ensure database consistency.

8.3.3.6. Add reporting users, schemas and views

This step is **ONLY** required if the migrated target is a new, empty EPC(P) system. The imported, migration, schema will be used as the Preservica database

Within the `..\Database\mysql` folder:

Configure the `database.properties` file as per the Preservica System Installation Guide [SIG]. The `mysql.database.sdb.username` should be set to the name of the temporary schema to be used for the migration eg `migration`. The associated password should be set in `mysql.database.sdb.password`. Reporting users are created so these values (`mysql.database.reporting...`) need to be set.

Create the reporting users, schemas and views using the following command:

```
ant upgrade-database-no-user-views, create-reporting-users
```

8.3.3.7. Update database identifiers

NOT CURRENTLY SUPPORTED IN V6

This step is **NOT** required if the migration target is a new, empty EPC(P) system.

Within the `..\TenancyManagement\TenancyMigrate` folder:

Update database identifiers in the migrated data to avoid conflict with the existing data:

```
ant update-ids
```

8.3.3.8. Merge Tenant data into Target Instance

NOT CURRENTLY SUPPORTED IN V6

This step is **NOT** required if the migration target is a new, empty EPC(P) system. The imported, migration, schema will be used as the Preservica database

Within the `..\TenancyManagement\TenancyMigrate` folder:

Merge the tenant data into the existing Preservica database:

```
ant merge-migrated-data
```

8.3.3.9. Restart the Target Preservica Instance

The target Preservica instance can now be restarted.

8.3.3.10. File Migration

At this point any changes relating to:

- Storage Adapters
- Preview Cache
- DIP Downloads
- Trial Migrations

should be made.

This may include physically copying the associated files to the target system and/or updating the associated settings within the Preservica database.

8.3.3.11. Import the User Account data

Within the `..\TenancyManagement\TenancyMigrate` folder:

Import the User Account (LDAP) data:

```
ant import-ldap
```

8.4. Deletion of the Migrated Tenant

Deletion of the migrated tenant is undertaken using the existing tenant deletion scripts found within the `..\TenancyManagement\TenancyDelete` folder.

Deletion should be undertaken once the migration has been validated. Please refer to the associated documentation for details of how to delete a tenant.

Chapter 9. System Backups

9.1. Solr

Although it is possible to rebuild the solr search index from Preservica this process would be very time consuming. As a result a regular backup of the solr index should be taken. The solr index consists of a set of binary files within the folder:

```
/opt/preservica/solr/<corename>/data/index (X:\Preservica\EE\solr_home\<corename>\data\index on Windows).
```

In multi-tenant systems it is possible (but not mandatory) to split each tenant into a separate index (core). The default core is called `collection1` and will contain the index for all tenants without a specific core. If a tenant specific core is required this **must** be named as per the tenant name **in lower case**. In the paths above `<corename>` corresponds to the name of the specific core. There may be multiple core files that need to be backed up.

The solr index is split across multiple files. A consistent backup must be made of all these files which may be changing due to ongoing changes to the solr index.

As a result following backup options are possible

1. Stop Preservica and backup the contents of each of the solr core folders (or the whole solr structure). In practice this approach is unlikely to be popular due to the necessary downtime
2. Mirror the solr directory at the Operating System level. Backup from a frozen mirror before restarting the mirroring. Due to the dependency on external hardware and software this approach is not supported by Preservica.
3. Create a consistent backup of the solr index (for each core in turn) using the inbuilt solr backup (snapshot) functionality.
4. Backup from a real time replica of the solr index (for each core in turn) having paused replication first.

9.1.1. Backing up using Solr snapshots

On Linux only a backup script `backup-solr.sh` is provided. This script should be scheduled on the required frequency (ideally daily) using a cron entry (or similar scheduling software). The script should be run as the `sdb` user.

The script will loop round each solr core found in the solr directory and use the solr snapshot functionality to create a consistent snapshot of the core data files to `/opt/preservica/backup/solr`. Once created the snapshot will be compressed into a single tar file the name of which will indicate the date and time of the backup and the solr core backed up e.g. `solr-backup-collection1-2017-01-28.tar.gz`.

By default the backup script will delete any backups found that are older than 10 days. It is recommended that either the backup files created are copied to another secure location or that the backup folder is mapped to an external drive (e.g. a NFS or CIFS share) and included in the backup of that external drive.

As the size of the index increases the time taken to create the consistent snapshot will increase and it may be necessary to use another backup approach.

9.1.2. Backing up using Solr index replication.

If you wish to use a replication based backup the same backup script can be used and will detect that the backups are being created from a replica of the index. However to use this approach it is first necessary to create the initial index replica.

To avoid conflicts the replica must be created on a different server than the one used to host the main solr application and index. This could be a dedicated additional sever; if a separate server is being used to host

the access component it is recommended that this server is used. The following steps should be taken to set up the replica:

- Stop the Preservica service:

```
# service preservica stop
```

- Install the solr application:

```
# yum -y install preservica-app-solr
```

- If not already installed, also install the utility scripts

```
# yum -y install preservica-scripts
```

- Update the solr index configuration for the default core by editing the following file:

/opt/preservica/solr/collection1/conf/solrconfig.xml

Change the section:

```
<requestHandler name="/replication" class="solr.ReplicationHandler" >
  <lst name="master">
    <str name="confFiles">schema.xml</str>
    <str name="commitReserveDuration">00:00:10</str>
  </lst>
</requestHandler>
```

to:

```
<requestHandler name="/replication" class="solr.ReplicationHandler" >
  <lst name="slave">
    <str name="confFiles">schema.xml</str>
    <str name="commitReserveDuration">00:00:10</str>
    <str name="masterUrl">http://devtest.preservica.com:200/
solr/collection1</str>
    <str name="pollInterval">00:00:20</str>
  </lst>
</requestHandler>
```

where:

- The `lst name` is changed from **master** to **slave**
- `str name="masterUrl"` specifies the URL of the main solr server (this can be taken from the `local.properties` file)
- `str name="pollInterval"` specifies the how regularly the index is replication. An initial value of 20 seconds is recommended.
- Update the httpd (Apache) configuration by editing the appropriate httpd configuration file.

If using https (recommended) the configuration file is: `/opt/preservica/conf/httpd/httpd-https.conf`

Insert the following lines:

```
ProxyPass /solr http://localhost:3535/solr
ProxyPassReverse /solr http://localhost:3535/solr
```

In to the VirtualHost on port 443 `VirtualHost *:443` **after** the directives enabling `ProxyPreserveHost`, `SSLProxyEngine` and `RewriteEngine`

If using http the configuration file is: `/opt/preservica/conf/httpd/httpd-http.conf`

Insert the following lines:

```
ProxyPass /solr http://localhost:3535/solr
ProxyPassReverse /solr http://localhost:3535/solr
```

In to the VirtualHost on port 80 `VirtualHost default:80` **after** the directives enabling `ProxyPreserveHost`, and `RewriteEngine`

- Reload the httpd (Apache) configuration:

```
# service httpd reload
```

- Restart the Preservica service:

```
# service preservica start
```

The script will loop round each solr core found in the solr directory and use the solr replication functionality to disable replication allowing a consistent backup of the core data files to `/opt/preservica/backup/solr`. Once created the backup will be compressed into a single tar file the name of which will indicate the date and time of the backup and the solr core backed up e.g. `solr-backup-collection1-2017-01-28.tar.gz` and replication for the core re-enabled.

By default the backup script will delete any backups found that are older than 10 days. It is recommended that either the backup files created are copied to another secure location or that the backup folder is mapped to an external drive (e.g. a NFS or CIFS share) and included in the backup of that external drive.

A warning will be reported if a corresponding master can not be found for any of the cores on the local (slave) solr instance. In addition a full list of cores on the master solr instance will be retrieved and checked against the local (slave) cores. If a core on the master instance is not replicated on the slave the script will automatically create a new local (slave) core and register this core with the local (slave) solr instance.

Chapter 10. Manual Database Changes

10.1. Adapters

An Adapter is invalidated by changing the modified field. This is done automatically when editing through the UI but must be done manually if changes are made directly to the database.



www.preservica.com

Reference: git/doc/ID/SG | Issue: 7.1.0

Copyright Preservica 2022 | All rights reserved | Preservica is a registered trademark

Preservica, 32 The Quadrant, Abingdon Science Park, Abingdon, Oxfordshire, OX14 3YS, UK info@preservica.com - [preservica.com](https://www.preservica.com)