

AHG Reports Menu & Archival Export Module

Version 2.0

Technical Documentation

The Archives and Heritage Group (AHG)

December 2024

Author: Johan Pieterse

1. Overview

This module provides comprehensive reporting, dashboard access, and archival export capabilities for AtoM installations. It includes:

1. **Hierarchical Reports Menu** with sub-menu pop-outs for all reports, audits, dashboards, exports, and imports
2. **Full Archival Description Export** supporting CSV (ISAD-G) and EAD formats
3. **Transportable Package Creation** with digital objects for system migration
4. **Package Import** to restore exported content on another AtoM instance

2. Menu Structure

The reports menu provides organized access to all reporting and data management functions through a modern hierarchical interface with sub-menu pop-outs.

2.1 Reports

- Accession Report
- Archival Description Report
- Authority Record Report
- Donor Report
- Physical Storage Report
- Repository Report
- Taxonomy Report
- Recent Updates Report
- User Activity Report

2.2 Audit

- Archival Description Audit
- Authority Record Audit
- Repository Audit
- Donor Audit
- Physical Storage Audit
- Taxonomy Audit
- Permissions Audit

2.3 Dashboards

Collections Management

- Spectrum Dashboard
- Condition Monitoring
- Loans Dashboard
- Valuation Dashboard

Financial

- GRAP 103 Dashboard
- Asset Register
- Depreciation Overview

Security & Access

- Security Dashboard
- Access Requests
- Clearance Overview

Digital Assets

- IIIF Dashboard
- Metadata Dashboard

System

- System Overview
- Statistics Dashboard

3. Export Formats

3.1 CSV (ISAD-G)

The CSV export provides 56 columns following the standard AtoM CSV format, fully compatible with AtoM import functionality.

Identity Area:

legacyId, parentId, qubitParentSlug, accessionNumber, identifier, title, levelOfDescription

Context Area:

extentAndMedium, repository, archivalHistory, acquisition

Content & Structure Area:

scopeAndContent, appraisal, accruals, arrangement

Conditions of Access Area:

accessConditions, reproductionConditions, language, script, findingAids

Allied Materials Area:

locationOfOriginals, locationOfCopies, relatedUnitsOfDescription, publicationNote

Access Points:

subjectAccessPoints, placeAccessPoints, nameAccessPoints, genreAccessPoints

Control Area:

descriptionIdentifier, rules, descriptionStatus, publicationStatus, culture

Events:

eventDates, eventTypes, eventStartDates, eventEndDates, eventActors, eventActorHistories

3.2 EAD (Encoded Archival Description)

EAD is an XML standard for encoding archival finding aids. The module supports both EAD 2002 and EAD3 (2015) formats.

EAD Features:

- Complete <eadheader> with file description
- Full <archdesc> with all descriptive elements
- Nested <c> components for hierarchy
- Subject and name access points

3.3 Dublin Core

OAI-DC format for simple interoperability with other systems.

3.4 MODS

Library of Congress MODS schema for detailed bibliographic description.

4. Transportable Package

When "Create transportable package" is selected, the export creates a ZIP file containing all metadata and optionally digital objects, suitable for migration to another AtoM instance.

4.1 Package Structure

```
package_name.zip
├─ manifest.json          # Package metadata and checksums
├─ metadata/
│   ├─ descriptions.csv   # Full CSV export
│   ├─ ead.xml            # EAD finding aid
│   └─ dc.xml             # Dublin Core metadata
└─ objects/              # Digital objects (if included)
    ├─ 123_image.jpg
    └─ 124_document.pdf
```

4.2 Manifest Structure

The manifest.json file contains package metadata and checksums for verification:

```
{
  "created": "2024-12-06T10:30:00+02:00",
  "generator": "AHG AtoM Export Module",
  "version": "2.0",
  "includesDigitalObjects": true,
  "files": [ { "path": "...", "checksum": "..." } ]
}
```

5. File Structure

```
ahg_reports/
├── actions/
│   ├── archivalExportAction.class.php
│   ├── csvExportAction.class.php
│   ├── eadExportAction.class.php
│   ├── downloadAction.class.php
│   ├── archivalPackageAction.class.php
│   ├── apiExportPreviewAction.class.php
│   └── apiReportsPendingCountsAction.class.php
├── lib/
│   └── arArchivalExportService.class.php
├── templates/
│   ├── _reportsMenu.php
│   ├── archivalExportSuccess.php
│   ├── csvExportSuccess.php
│   ├── eadExportSuccess.php
│   └── archivalPackageSuccess.php
├── config/
│   └── routing.yml
└── web/
    ├── css/reports-menu.css
    └── js/reports-menu.js
```

6. Installation

6.1 Copy Files to Plugin

```
# Copy actions
cp -r ahg_reports/actions/* plugins/arAHGPlugin/modules/export/actions/
cp ahg_reports/actions/archivalPackageAction.class.php \
  plugins/arAHGPlugin/modules/import/actions/

# Copy library
cp ahg_reports/lib/* plugins/arAHGPlugin/lib/

# Copy templates
cp -r ahg_reports/templates/* plugins/arAHGPlugin/modules/export/templates/

# Copy assets
cp ahg_reports/web/css/* plugins/arAHGPlugin/css/
cp ahg_reports/web/js/* plugins/arAHGPlugin/js/
```

6.2 Add Routes

Append the contents of config/routing.yml to your apps/qubit/config/routing.yml.

6.3 Include Menu Component

Add to your layout file (e.g., layout.php):

```
<?php include_component('ahg', 'reportsMenu'); ?>
```

6.4 Include CSS and JS

Add to your view.yml or layout:

```
stylesheets:
  - /plugins/arAHGPlugin/css/reports-menu.css

javascripts:
  - /plugins/arAHGPlugin/js/reports-menu.js
```

6.5 Clear Cache

```
php symfony cc
```

7. Security

- All export actions require authentication
- Administrator or audit user role required for reports menu
- File downloads validated for type and age (24-hour expiry)
- Package checksums verified on import
- Directory traversal protection on download paths

8. Author & Version

Author: Johan Pieterse <johan.pieterse@sita.co.za>

Organization: The Archives and Heritage Group (AHG)

Version: 2.0

Date: December 2024