Function 8: Export Identifier CSV

Overview

Function 8 creates a specialized 4-column CSV export that focuses exclusively on dc:identifier fields from bibliographic records in Alma. This function extracts and categorizes three specific types of identifiers used in Grinnell College's digital collections, making it easy to analyze identifier distribution, track migration progress, and identify records that need identifier updates.

What It Does

This function processes all records in a loaded set and exports a streamlined CSV file containing:

- **MMS ID**: Alma's unique record identifier
- *dg_ identifier**: Legacy Digital Grinnell identifiers (format: `dg_12345`)
- *Grinnell: identifier**: Standardized Grinnell identifiers (format: `Grinnell:12345`)
- **Handle identifier**: Persistent Handle System identifiers (format: `http://hdl.handle.net/11084/...`)

Key Features

- **Focused export**: Only identifier fields, no other metadata
- **Categorized identifiers**: Three specific identifier types in separate columns
- **Batch processing**: Efficient API calls (100 records per batch)
- **Progress tracking**: Real-time progress bar during export
- **Kill switch**: Stop processing if needed
- **Automatic naming**: Timestamped filename with date/time
- **UTF-8 support**: Preserves all character encodings
- **Empty cells**: Shows which identifier types are missing per record

The Need for This Function

Identifier Migration Tracking

During the migration from Digital Grinnell (Islandora) to Alma Digital, records went through an identifier transformation process:

- **Legacy**: `dg_12345` identifiers from old system
- **Standardized**: `Grinnell:12345` identifiers for new system
- **Persistent**: `http://hdl.handle.net/11084/...` for permanent URLs

Function 8 helps answer:

- Which records still have only legacy identifiers?
- Which records have been migrated to standardized format?
- Which records have Handle identifiers?
- Are there any records with multiple identifier types?
- What's the overall migration progress?

Simplified Analysis

Function 3 exports all Dublin Core fields (65 columns), which is comprehensive but overwhelming when you only need identifier information. Function 8 provides:

- **4 columns vs. 65**: Much easier to review
- **Instant overview**: See identifier status at a glance
- **Fast processing**: Smaller file, quicker to analyze
- **Spreadsheet-friendly**: Easy to filter, sort, and pivot

How It Works

Step-by-Step Process

1. **Load Set**:
    - User loads a set of records to process
    - Set members stored in application memory
2. **Execute Function**:
    - Click Function 8 button
    - Generate timestamped filename
    - Display progress bar
3. **Batch Fetching** (per 100 records):
    - Send batch GET request to Alma Bibs API
    - Receive XML for up to 100 records at once
    - Parse each record's Dublin Core section
4. **Identifier Extraction** (per record):
    - Find all `dc:identifier` elements
    - Check each identifier's prefix
    - Categorize into three types:
        - Starts with `dg_` → dg_* identifier column
        - Starts with `Grinnell:` → Grinnell:* identifier column
        - Starts with `http://hdl.handle.net/` → Handle identifier column
5. **CSV Writing**:
    - Write header row with 4 column names
    - Write one row per record with categorized identifiers
    - Empty string if identifier type not found
6. **Progress Updates**:
    - Update progress bar after each record
    - Log batch completion
    - Show final statistics

Identifier Categorization Logic

Python Implementation:

```
# Extract all dc:identifier values
identifiers = self._extract_dc_field("identifier", "dc")

# Initialize empty values
dg_identifier = ""
grinnell_identifier = ""
handle_identifier = ""

# Categorize each identifier
for identifier in identifiers:
    if identifier.startswith("dg_"):
        dg_identifier = identifier
    elif identifier.startswith("Grinnell:"):
        grinnell_identifier = identifier
    elif identifier.startswith("http://hdl.handle.net/"):
        handle_identifier = identifier

# Write to CSV row
row = {
    "MMS ID": mms_id,
    "dg_* identifier": dg_identifier,
    "Grinnell:* identifier": grinnell_identifier,
    "Handle identifier": handle_identifier
}
```

Important Notes:

- If multiple identifiers of same type exist, only the first is captured
- Identifiers not matching any pattern are ignored
- Empty strings used for missing identifier types
- Case-sensitive matching

Usage

Basic Export

Step 1: Load Set

1. Enter set ID in "Set ID" field
   - Example: `7071087320004641` (DCAP01 set)
   - Or click the DCAP01 set ID link to auto-populate
2. Click "Load Set" button
3. Wait for confirmation: "Set loaded: 2,847 records"

Step 2: Select Function

1. Open function dropdown menu

2. Select "Export Identifier CSV (dg_, *Grinnell:*, Handle)"
3. Function 8 button becomes active

Step 3: Execute Export

1. Click Function 8 button
2. Progress bar appears immediately
3. Watch progress: "Processing record 1 of 2,847"
4. Wait for completion (typical: 30-60 minutes for 2,847 records)

Step 4: Locate Output File

1. Check CABB project directory
2. Find file with pattern: `identifier_export_YYYYMMDD_HHMMSS.csv`
3. Example: `identifier_export_20241204_143022.csv`

Step 5: Open and Analyze

1. Open CSV in spreadsheet application (Excel, Google Sheets, etc.)
2. Review the 4 columns
3. Use filters to find specific patterns
4. Create pivot tables or charts as needed

Kill Switch Usage

When to Use:

- Export taking longer than expected
- Need to stop for system maintenance
- Discovered wrong set was loaded
- Want partial export for testing

How to Use:

1. During export, click "Kill" button
2. Current record completes
3. Remaining records skipped
4. Partial CSV file created with records processed so far

Result:

- CSV contains only successfully processed records
- File still usable for partial analysis
- Can reload set and re-run for complete export

Output File Format

Filename Convention

Pattern: `identifier_export_YYYYMMDD_HHMMSS.csv`

Components:

- `identifier_export`: Fixed prefix
- `YYYYMMDD`: Date (e.g., 20241204 = December 4, 2024)
- `HHMMSS`: Time (e.g., 143022 = 2:30:22 PM)
- `.csv`: File extension

Examples:

- `identifier_export_20241204_143022.csv` - Full DCAP01 export at 2:30 PM
- `identifier_export_20241204_090000.csv` - Morning export at 9:00 AM
- `identifier_export_20241204_150000.csv` - Afternoon export at 3:00 PM

CSV Structure

Header Row:

```
MMS ID,dg_* identifier,Grinnell:* identifier,Handle identifier
```

Data Rows:

```
991234567890104641,dg_12345,Grinnell:12345,http://hdl.handle.net/11084/12345
991234567890204641,dg_12346,,
991234567890304641,,Grinnell:12347,http://hdl.handle.net/11084/12347
991234567890404641,,,http://hdl.handle.net/11084/12348
```

Column Details:

Column	Description	Example Values	Can Be Empty
MMS ID	Alma record identifier	991234567890104641	No (always present)
dg_* identifier	Legacy Digital Grinnell ID	dg_12345	Yes
Grinnell:* identifier	Standardized Grinnell ID	Grinnell:12345	Yes
Handle identifier	Persistent Handle URL	http://hdl.handle.net/11084/12345	Yes

Character Encoding

- **UTF-8 throughout**: All special characters preserved
- **CSV-safe**: Commas in values handled properly
- **No BOM**: Standard UTF-8 without byte order mark
- **Line endings**: Platform-appropriate (CRLF on Windows, LF on Unix/Mac)

Use Cases

1. Migration Progress Tracking

Scenario: Monitor Function 7's addition of Grinnell:* identifiers

Workflow:

1. **Before Function 7**:

   - Load DCAP01 set
   - Run Function 8 to export baseline
   - File shows many records with dg_* but no Grinnell:*

2. **Run Function 7**:

   - Execute "Add Grinnell: dc:identifier Field As Needed"
   - Wait for completion

3. **After Function 7**:

   - Run Function 8 again
   - Compare new export to baseline
   - Verify Grinnell:* column now populated

Analysis:

```
Before: 2,847 records with dg_* identifier, 0 with Grinnell:*
After:  2,847 records with dg_* identifier, 2,847 with Grinnell:*
Result: 100% migration success
```

2. Identify Records Needing Updates

Scenario: Find records that need identifier standardization

Workflow:

1. Export identifier CSV for collection

2. Open in Excel or Google Sheets

3. Filter for specific patterns:

   - *Has dg_ but no Grinnell:***: Needs Function 7
   - *Has Grinnell: but no Handle**: Needs Handle registration
   - **Has none of the three**: Needs manual investigation

4. Create work lists from filtered results

5. Process records accordingly

Example Filters:

Excel Formula - Missing Grinnell:

```
=AND(B2<>"", C2="")
```

Google Sheets Filter:

- Column B (dg_* identifier) is not empty
- Column C (Grinnell:* identifier) is empty

3. Audit Identifier Consistency

Scenario: Verify identifier relationships are correct

Workflow:

1. Export identifier CSV
2. Import to analysis tool or script
3. Verify patterns:
    - dg_12345 should correspond to Grinnell:12345
    - Extract number from both and compare
    - Flag any mismatches

Python Analysis Script:

```python
import csv

mismatches = []

with open('identifier_export_20241204_143022.csv', 'r') as f:
    reader = csv.DictReader(f)
    for row in reader:
        dg_id = row['dg_* identifier']
        grinnell_id = row['Grinnell:* identifier']

        if dg_id and grinnell_id:
            # Extract numbers
            dg_num = dg_id.replace('dg_', '')
            grinnell_num = grinnell_id.replace('Grinnell:', '')

            # Compare
            if dg_num != grinnell_num:
                mismatches.append({
                    'MMS ID': row['MMS ID'],
                    'dg': dg_id,
                    'grinnell': grinnell_id
                })

print(f"Found {len(mismatches)} mismatches")
for m in mismatches:
    print(m)
```

4. Handle System Registration Tracking

Scenario: Track which records have persistent Handle URLs

Workflow:

1. Export identifier CSV
2. Count records with Handle identifiers
3. Identify gaps where Handles should exist
4. Generate list for Handle registration
5. After registration, re-export and verify

Spreadsheet Analysis:

```
Total records: 2,847
Records with Handle: 1,234
Missing Handle: 1,613
Coverage: 43.4%
```

5. External System Integration

Scenario: Provide identifier mapping to external discovery system

Workflow:

1. Export identifier CSV from Alma
2. Send file to web developers
3. They use it to create URL mappings:
   - Legacy URLs (dg_*) redirect to Handle URLs
   - Grinnell:* identifiers used in search indexing
   - Handle URLs as canonical permalinks
4. Update export monthly or after major changes
5. Keep external system synchronized

6. Quality Assurance Reporting

Scenario: Document identifier status for annual report

Workflow:

1. Export identifier CSV at end of fiscal year
2. Generate statistics:
   - Total records
   - Percent with each identifier type

   - Migration completion percentage
   - Year-over-year growth

3. Create visualizations:

   - Pie chart: identifier type distribution
   - Bar chart: migration progress over time

4. Include in annual digital collections report

Technical Details

API Operations

Batch GET Request:

```
GET /almaws/v1/bibs?mms_id=ID1,ID2,ID3,...,ID100&view=full&expand=None
Accept: application/xml
Authorization: apikey {api_key}
```

Parameters:

- `mms_id`: Comma-separated list of up to 100 MMS IDs
- `view=full`: Returns complete record data
- `expand=None`: No additional linked data
- `apikey`: API authentication key

Response:

- Multiple `<bib>` elements in `<bibs>` wrapper
- Each contains full bibliographic record
- Status: 200 on success

XML Parsing

Dublin Core Section:

```xml
<bib>
  <mms_id>991234567890104641</mms_id>
  <anies>
    <record xmlns="http://alma.exlibrisgroup.com/dc/01GCL_INST"
            xmlns:dc="http://purl.org/dc/elements/1.1/"
            xmlns:dcterms="http://purl.org/dc/terms/">
      <dc:identifier>dg_12345</dc:identifier>
      <dc:identifier>Grinnell:12345</dc:identifier>
      <dc:identifier>http://hdl.handle.net/11084/12345</dc:identifier>
      <dc:identifier>https://digital.grinnell.edu/...</dc:identifier>
      <!-- Other Dublin Core fields -->
    </record>
  </anies>
</bib>
```

Extraction Process:

1. Find `<anies>` element in bib record
2. Parse XML string within anies
3. Find all `dc:identifier` elements using namespace
4. Iterate through identifiers and categorize by prefix
5. Store first match of each type

Performance Considerations

Time per Record:

- In 100-record batch: ~0.5 seconds per record
- Individual parsing: negligible (<0.01 seconds)
- CSV writing: negligible (<0.01 seconds)
- Total: ~0.5-1 second average per record

Total Time Estimates:

- 100 records: 1-2 minutes
- 500 records: 5-8 minutes
- 1,000 records: 10-15 minutes
- 2,847 records: 30-45 minutes

File Sizes:

- 100 records: ~5-10 KB
- 1,000 records: ~50-100 KB
- 2,847 records: ~150-200 KB

Comparison to Function 3:

- Function 3: 2,847 records = ~2 hours (65 columns, complex extraction)
- Function 8: 2,847 records = ~30-45 minutes (4 columns, simple extraction)
- **Function 8 is 2-3x faster** for identifier-only needs

Error Handling

Individual Record Failures:

- Error logged with MMS ID and details
- Record skipped in output (no partial row)
- Processing continues to next record
- Failed count incremented

Common Issues:

Error	Cause	Handling
404 Not Found	Invalid MMS ID	Skip, log error
No anies section	Non-DC record	Skip, log warning
Parse error	Malformed XML	Skip, log error
Empty identifiers	No dc:identifier fields	Write row with empty identifier columns

Network Failures:

- Full traceback logged
- User notified via status message
- Can use kill switch to stop
- Partial CSV file preserved

Output Analysis Tips

Spreadsheet Formulas

*Count Records with dg_ Identifier:**

```
=COUNTIF(B:B,"<>")
```

*Count Records with Grinnell: Identifier:**

```
=COUNTIF(C:C,"<>")
```

Count Records with Handle Identifier:

```
=COUNTIF(D:D,"<>")
```

*Find Records Missing Grinnell: but Having dg_*:**

```
=IF(AND(B2<>"",C2=""),"Needs Grinnell ID","OK")
```

*Extract Number from dg_ Identifier:**

```
=SUBSTITUTE(B2,"dg_","")
```

*Extract Number from Grinnell: Identifier:**

```
=SUBSTITUTE(C2,"Grinnell:","")
```

Compare Numbers Match:

```
=IF(SUBSTITUTE(B2,"dg_","")=SUBSTITUTE(C2,"Grinnell:",""),"Match","Mismatch")
```

Pivot Table Analysis

Recommended Pivot Tables:

1. **Identifier Type Distribution:**
    - Rows: Has dg_*, Has Grinnell:*, Has Handle
    - Values: Count of MMS IDs
    - Result: Shows combinations (e.g., "Has all 3", "Has dg_ only", etc.)
2. **Migration Status:**
    - Filters: Has dg_* identifier
    - Columns: Has Grinnell:* identifier (True/False)
    - Values: Count
    - Result: Migration progress percentage
3. **Handle Coverage:**
    - Rows: Has Handle (True/False)
    - Values: Count, Percent of Total
    - Result: Handle registration progress

Data Visualization

Excel/Google Sheets Charts:

Stacked Bar Chart - Identifier Completeness:

```
X-axis: Identifier Type (dg_*, Grinnell:*, Handle)
Y-axis: Count of Records
Series: Present, Missing
```

Pie Chart - Migration Status:

```
Slices:
- Both dg_* and Grinnell:*
- Only dg_* (needs migration)
- Only Grinnell:* (migrated, dg_ removed)
- Neither
```

Line Chart - Progress Over Time:

```
X-axis: Export Date
Y-axis: Count
Lines:
- Records with Grinnell:* identifier
- Records with Handle identifier
```

Comparison with Other Functions

Function 8 vs. Function 3

Aspect	Function 3	Function 8
Columns	65 (all DCAP01 fields)	4 (identifiers only)
File size	Large (~2-5 MB for 2,847 records)	Small (~150-200 KB)
Export time	~2 hours	~30-45 minutes
Use case	Complete metadata export	Identifier analysis only
Complexity	High (many field mappings)	Low (simple categorization)
Best for	Comprehensive backup, migration	Tracking, auditing, reporting

When to Use Function 3:

- Need complete metadata backup
- Preparing for migration to another system
- Comprehensive data quality analysis
- Documentation requiring all fields

When to Use Function 8:

- Only need identifier information
- Tracking Function 7 progress
- Quick identifier audit
- Handle registration planning
- Faster export needed

Function 8 vs. Function 5

Aspect	Function 5	Function 8
Format	JSON (complete XML)	CSV (identifiers only)

Aspect	Function 5	Function 8
Use case	Programmatic processing	Spreadsheet analysis
Output	Machine-readable XML	Human-readable table
Parsing	Requires XML tools	Open in Excel directly

When to Use Function 5:

- Need complete record XML
- Building external applications
- Complex data transformation
- Backup with all original data

When to Use Function 8:

- Quick visual review needed
- Sharing with non-technical staff
- Spreadsheet-based analysis
- Simple identifier tracking

Best Practices

Before Export

1. **Verify set membership**: Ensure set contains intended records
2. **Estimate time**: ~1 second per record (2,847 records ≈ 45 minutes)
3. **Plan analysis**: Know what you'll do with the export
4. **Test with small set**: Try 100 records first if uncertain

During Export

1. **Monitor progress**: Check progress bar periodically
2. **Don't close browser**: Keep application open
3. **Avoid system sleep**: Disable sleep mode for exports >30 minutes
4. **Note any errors**: Check status messages for issues

After Export

1. **Verify file created**: Check CABB directory for CSV file
2. **Open and review**: Quick scan in spreadsheet application
3. **Check record count**: Compare to set size
4. **Spot check data**: Verify a few rows look correct
5. **Save to appropriate location**: Move file to analysis folder

Data Analysis

1. **Use filters**: Excel/Sheets filters for quick insights
2. **Create pivot tables**: Summarize identifier patterns
3. **Compare over time**: Export regularly to track changes

4. **Document findings**: Note anomalies or issues discovered
5. **Share appropriately**: CSV is easy to share with colleagues

Limitations

- **Set-based only**: Cannot export arbitrary MMS ID list without creating set
- **One identifier per type**: If multiple dg_* identifiers exist, only first captured
- **No other metadata**: Only identifiers, no titles, dates, etc.
- **No filtering**: Exports all records in set, cannot selectively process
- **Pattern-specific**: Only recognizes exact prefixes (dg_, Grinnell:, http://hdl.handle.net/)
- **No custom patterns**: Cannot add other identifier types without code changes

Troubleshooting

Empty Identifier Columns

Symptom: All identifier columns empty for some/all records

Possible Causes:

- Records don't have Dublin Core metadata
- Identifiers use different prefixes than expected
- dc:identifier fields missing or empty

Solutions:

- Use Function 1 to inspect sample records
- Verify records have `<anies>` section
- Check actual identifier format in Dublin Core
- Confirm records are digital collection items

Export Slower Than Expected

Symptom: Export taking much longer than estimated

Possible Causes:

- Network latency
- Alma server under heavy load
- Very large records (with many fields)

Solutions:

- Run during off-peak hours (nights/weekends)
- Check network connection
- Use kill switch and retry later
- Contact Alma support if persistent issue

Mismatched Identifier Numbers

Symptom: dg_12345 paired with Grinnell:99999 in same record

Possible Causes:

- Manual editing error in Alma
- Function 7 bug (unlikely)
- Record merged or split incorrectly

Solutions:

- Use Function 1 to view full record
- Manually correct in Alma metadata editor
- Document for further investigation
- Report pattern if multiple records affected

CSV Doesn't Open Correctly

Symptom: File opens with garbled characters or wrong columns

Possible Causes:

- Not opened with UTF-8 encoding
- Excel default encoding issue
- Commas in identifier values (rare)

Solutions:

- Use Excel's "Import Data" feature with UTF-8
- Open in Google Sheets (better UTF-8 support)
- Check file in text editor first
- Verify file wasn't corrupted during creation

Integration with Other Functions

Before Function 7 (Add Grinnell Identifier)

Workflow:

1. Run Function 8 to export baseline
2. Review how many records have dg_* but not Grinnell:*
3. Run Function 7 to add Grinnell:* identifiers
4. Run Function 8 again to verify

Benefits:

- Document pre-migration state
- Calculate expected changes
- Verify Function 7 results
- Track migration completion

After Function 7

Workflow:

1. Function 7 completes adding Grinnell:* identifiers
2. Run Function 8 immediately
3. Compare to pre-Function 7 export
4. Verify all expected records now have Grinnell:*

Validation:

```
Before Function 7: Grinnell:* column mostly empty
After Function 7:  Grinnell:* column populated
Success metric:    Grinnell:* count = dg_* count
```

With Function 1 (Single XML View)

Workflow:

1. Export identifier CSV
2. Find interesting records (e.g., missing identifier)
3. Copy MMS ID from CSV
4. Use Function 1 to view full XML
5. Investigate issue in detail

With Function 3 (Full CSV Export)

Complementary Use:

- Function 8 first for quick identifier overview
- Function 3 later if full metadata needed
- Cross-reference MMS IDs between exports
- Use Function 8 for regular monitoring, Function 3 for comprehensive backup

Related Documentation

- **Function 7**: Add Grinnell: dc:identifier Field As Needed
- **Function 3**: Export Set to DCAP01 CSV
- **Function 1**: Fetch and Display Single XML
- **Dublin Core dc:identifier**: https://www.dublincore.org/specifications/dublin-core/dcmi-terms/#identifier
- **Handle System**: https://www.handle.net/

Version History

- **Initial Implementation**: December 2024
- **Purpose**: Track identifier migration and Handle System registration
- **Status**: Active, production-ready