Business Requirements Specification (BRS)

Project: DTN ADMLOD Export Prototype  
Prepared by: Jordan Eisenback  
Date: 9/30/25

Project Objective

The objective of this project is to deliver a C# command-line prototype that simulates the integration with the DTN ADMLOD (FTP) interface. The prototype will generate ADMLOD-compliant export files for Products and Product Groups using mock data, simulate FTP upload, and parse mock DTN response files. The prototype will demonstrate both successful and error scenarios, providing a foundation for future development and stakeholder review.

Scope

In Scope

* Command-line interface for triggering export operations.
* Mock data for Products and Product Groups.
* Generation of ADMLOD export files using the correct format and structure.
* Simulation of FTP upload (local file write, not actual FTP).
* Simulation of DTN response parsing using static mock response files.
* Logging of all actions and errors to the console and/or file.
* Simulation of error scenarios, including:
  + Missing required fields
  + FTP failure
  + Invalid command format
* Demonstration of both successful and error flows.

Out of Scope

* Real FTP connectivity or live integration with DTN systems
* Import, add, modify, or delete operations (only export is supported)
* User interface or web interface
* Use of real credentials or sensitive data
* Production deployment or use

Business Rules

* Only export operations are supported in the prototype
* All data (Products and Product Groups) is mocked; no live system dependencies exist
* ADMLOD file format must strictly follow DTN documentation and templates
* Prototype must demonstrate both successful and error scenarios
* All actions and errors must be logged for review and traceability
* Logging must follow structured standards including severity levels (INFO, WARN, ERROR), timestamps, and correlation IDs
* Error handling must categorize failures (e.g., validation, FTP, parsing) and support retry logic where applicable
* The prototype must be implemented in C# and run as a command-line application
* All integrations (FTP, DTN) are simulated/mocked

Success Criteria

* The prototype can generate an ADMLOD export file from mock data for Products and Product Groups
* The prototype can simulate FTP upload and log the action
* The prototype can parse and display results from a mock DTN response file, including both success and error cases
* The prototype can simulate and log error scenarios (e.g., missing fields, FTP failure, invalid command)
* Future iterations must demonstrate structured logging and error categorization, including retry behavior and traceability via correlation IDs
* All steps are triggered and visible via the command line
* The prototype is easy to run and review by stakeholders

Assumptions

* DTN ADMLOD file format and command structure are as per provided documentation and templates
* No real credentials or sensitive data are used at any stage
* Prototype is for demonstration and planning only, not for production use
* All required mock data and response files will be provided or created as part of the project

Constraints

* Must be implemented in C# (.NET 6 or later recommended)
* Must run as a command-line application on Windows or cross-platform
* All integrations (FTP, DTN) must be mocked/simulated
* Only export operations are in scope; no other ADMLOD commands are supported
* Logging and error handling must conform to structured observability standards

Glossary

* DTN: Data Transmission Network, provider of the ADMLOD interface
* ADMLOD: Automated Data Management Load, DTN’s batch file protocol for data integration
* FTP: File Transfer Protocol, used for file uploads/downloads (mocked in this prototype)
* Product: An individual item to be exported
* Product Group: A logical grouping of products for export
* Mock Data: Simulated data used in place of real system data
* Mock Response: Simulated DTN response file used to demonstrate parsing and error handling