# PASTA Population Preparation Plan

## Readiness Review

* Data set inventory – consider if you expect to exclude datasets from your existing inventory. Due to quality issues, or metadata deficiencies your final proposed inventory (PASTA submission list) may be different to the inventory you currently have in Metacat.
* Dataset re-packaging – consider if you will repackage datasets to make them more relevant for ingestion into the NIS.
* Decide on your approach to the data upload, e.g. incremental upload over time or bulk upload for similar data products. You may wish to upload at least some datasets manually into the staging system so you become familiar with the upload mechanics before running a larger test upload to the same system.
* Identify and document issues, risks and dependencies that may affect the success of your data migration. These are defined as:
  + **Issues** – issues are existing situations or circumstances that will inhibit your ability to migrate your data into the NIS. These should be managed by an owner to ensure they are addressed in a timely manner.
  + **Risks** – these are situations that may occur that will inhibit your ability to complete the migration. Risks can be categorized by the likelihood of their occurrence and their impact if they do. You should give plan how you might mitigate (reduce) the risks you identify.
  + **Dependencies** – these are the relationships between the tasks that need to be completed and will include tasks that must be finished before others can either begin, or be finished.
* Periodically review your progress, paying attention to the risks, issues and dependencies to ensure that they do not impact your migration schedule. This way, you will avoid a situation developing that takes you by surprise.

## PASTA Population Checklist

This checklist provides a start point for migration planning and is not intended to be exhaustive. You may wish to add additional tasks that are relevant to your site specific circumstances. Dependencies can be logged here and an owner assigned to ensure that dependent tasks are addressed in an appropriate timeframe. (See notes for additional tips, courtesy of Gastil B.)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Implementation Task | Dependencies | Deliverables | Owner | Target Date | Actual Date |
|  | Inventory existing site datasets (a) |  |  |  |  |  |
|  | Refine dataset submission list |  |  |  |  |  |
|  | Identify and complete pre-requisite tasks (b, c) |  |  |  |  |  |
|  | Run congruency checks against target datasets (c) |  |  |  |  |  |
|  | Fix structural dataset and metadata errors |  |  |  |  |  |
|  | Fix content-related dataset and metadata errors (d) |  |  |  |  |  |
|  | Identify and address options for non-PASTA capable datasets |  |  |  |  |  |
|  | Run test upload to staging area |  |  |  |  |  |
|  | QA/QC staged datasets (e) |  |  |  |  |  |
|  | Upload datasets to production area |  |  |  |  |  |

**Notes:**

The between tasks are not just site-specific. The above list is just the first level of the outline of tasks.

1. Inventory: This is an iterative process, as it may be necessary to add more fields as the inventory and upload process proceeds.
2. There are many steps prior to running congruency checks. (And not all checks are congruency checks.). Before putting a data set through the checke at all, several basic steps will be required, including upgrades from EML 2.0.1 and all that entails (esp all that surfaces), having a data url at the entity level, and having a data table description.
3. Steps 3 and 4 are also iterative. It is a cycle.
4. The PASTA checker only looks at a metadata structure, not content. And not all structure is even checked yet. Only a human can judge if a title is adequate and will work well in a result set of mixed sites and beyond LTER.
5. Some QA/QC will be done at earlier stages. Everyone needs a place to see drafts at a much finer iteration level than after uploading to PASTA, or even PASTA staging. You would not want to have to upload to pasta staging for every metadata edit. It may be useful useful to keep revision numbers synchronized by observing the rule that one only gets to upload a revision once.

## Risk Log

Risk can be logged here along with their mitigation plans, i.e. the plan to reduce either the likelihood and/or the impact of occurrence. Risks can be flagged with severity of impact and the probability of them occurring (high, medium, low). Risk mitigation should be prioritized based on their rating. Mitigation plans should be owned and the risk can be closed if it ceases to be relevant.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Item | Impact (H, M, L) | Probability (H, M, L) | Mitigation Plan | Owner | Status (O/C) |
|  |  |  |  |  |  | O |
|  |  | H |  |  |  | O |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

## Issues Log

Issues can be logged here with their resolution plans, i.e. the action plan expected to address the issue. Issues can be prioritized based on their impact. Issues should be owned and can be closed once they are resolved. The list below provides examples of known issues based on user experiences to date. These may or may not be relevant to your own implementation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Description | Date logged | Priority (H, M, L) | Owner | Resolution | Date Resolved |
|  |  |  |  |  |  |  |
|  |  |  |  |  | . |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |