Pipeline

All processing within Cocoon is handled within a *pipeline*. i.e. input, zero or more processing stages, and ultimately some output.

Processing steps are modelled as different types of Components, the simplest pipeline is:

- Input: Generator
- Processing: Transformer
- Output: Serializer

Therefore a pipeline consists of a generator, zero or more transformers and a serializer. A pipeline may also define its own ErrorHandling.

Requests to Cocoon are mapped to pipeline by a Matcher or a Selector. These are configured in the Sitemap.

Aggregator allow pipeline to be configured into a hierarchy, producing a compound document.

Definition

The map:pipeline element can contain:

- map:match – Selects pipeline processing depending on Matcher
- map:select, map:when, map:otherwise – Selector pipeline processing depending on Selector
- map:mount – Mounts a sub Sitemap
- map:redirect-to – Redirects to another pipeline
- map:parameter – Defines additional parameters for the sitemap components
- map:act – Peform Action processing
- map:generate – Defines the Generator step
- map:aggregate, map:part – Defines an alternate generation step by merging pipelines
- map:transform – Defines zero or more Transformer steps
- map:serialize – Defines the final Serializer step
- map:handle-errors – ErrorHandling

Types of Pipeline

- Normal pipelines
- Internal pipelines – can't be accessed externally, can only be addressed from within a sitemap. Specified by `internal-only="true"` attribute on `map:pipeline`
- Resource pipelines – basically normal pipelines that act as reusable definitions that can be called by others. See Resources
- Error pipelines – defined within `handle-errors` section. Fixed error Generator which is provided automatically by Cocoon, subsequent processing under user definition. See ErrorHandling