Polina Shpilker, Lenore J. Cowen,
Alva Couch, Noah M. Daniels
Tufts University and
the University of Rhode Island

MEDFORD Language Specification

Abstract

This document describes the syntax specification and validation requirements for MEDFORD, a human-readable, easily editable and templatable metadata language for scientists to collocate all the details relevant to a singular research project. The MEDFORD syntax is simple enough to allow for human writing and editing on-the-fly, but structured enough to allow for automated parsing, validation, and output into other formats. The ‘medford‘ software is a command-line tool that parses a given MEDFORD file, ensuring it follows all syntax requirements and, if desired, gathers the MEDFORD file and any specified external files into a Bag, as defined in RFC8493.

Contents

1. Introduction

This document describes the syntax and semantics, as well as validation requirements for MEDFORD, a declarative, human-readable markup language for describing the metadata associated with various kinds of dataset, with a goal of enabling scientists to collocate all the details associated with a research project. MEDFORD metadata files are typically associated with a BagIt bag (RFC8493), which can contain associated data files, code, or other materials. MEDFORD's syntax is intended to be simple enough to allow editing by non-programmers, but structured enough to allow for automated parsing, validation, and transformation into other formats. This document also describes the 'medford' command-line tool, which parses a MEDFORD file, checks validation rules, and ultimately constructs the associated BagIt bag.

1.1. Goals

MEDFORD is a language for describing heterogeneous scientific metadata, that is designed to be lightweight, and both human- and machine readable and writable. Our reference implementation of MEDFORD is designed to inter-operate with the BagIt hierarchical file packaging format designed to support storage and transfer of arbitrary digital content. BagIt associates a set of data files with an entity called a bag, preserving directory and file tree information. MEDFORD posits a single additional flat file deposited into the bag written in the MEDFORD language to describe the associated metadata (a MEDFORD-formatted file without an associated bag or with an empty bag are special cases, which we discuss below). MEDFORD stands for "Metadata Format for Open Reef Data," and MEDFORD was initially designed and tested by scientists researching corals. The MEDFORD reference implementation provides several helpful tools, including a syntax validator and a converter that can translate a file written in MEDFORD format into other popular database-based Metadata formats (so far only BagIt itself is implemented in MEDFORD 1.0, but extensions are planned to RDF and BCO-DMO formats).

1.2. Requirements

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

Note that the word ''EXPECTED'' is used when describing minor tokens and is to be interpreted as defined in Section 2.6 below.

1.3. Overview of the MEDFORD file structure and syntax

A MEDFORD file is a plain text file where certain characters are reserved. In particular, @ at the beginning of a line defines a tag as described below. A tag begins a statement, which also includes the metadata. For example:

```
@Contributor Luke Skywalker
```

indicates that Luke Skywalker is a contributor to a data resource associated with the MEDFORD file.

A complete list of reserved characters appears in Section 2.

A statement beginning with a tag may continue onto multiple lines, until the next tag. A statement may begin with a tag describing a minor tag which further expands the metadata being described. See section 1.4 for more detail.

Lines beginning with # are treated as comments. Comments are not validated and will not be propagated to other metadata file types.

A MEDFORD file may include macro definitions and template placeholders as discussed in sections 2.4 and 2.5.

1.4. Terminology

The following terms have precise definitions as are used in this document:

compile The act of the medford software converting a validated MEDFORD file into a Bag containing the MEDFORD file and all related files, following all BagIt specifications.

major tag A word that defines what kind of data is being described. Can either be directly following the '@' character, or follow another major tag and a '_' character.

minor tag A word that defines the type of metadata that is being described for the current major tag. Must always follow a major tag and a '-' character.

medford The software (e.g. the reference implementation written in Python) that either validates or compiles a given MEDFORD file. Note that here we distinguish between the software and the language via capitalization.

MEDFORD The MEDFORD language itself, which can be understood by the medford software or other hypothetical software meeting this specification.

MEDFORD file A file written according to the MEDFORD syntax.

statement A string that contains the '@' character at its head, a collection of major tags, a minor tag, and all following characters until a statement terminator is encountered. Trailing white spaces are purged from the statement.

statement terminator Any content that may terminate a statement. This collection currently includes (as of version 0.9): a novel statement declaration, a comment, or a macro declaration.

tag A series of alphabetical characters that represent either a major or a minor tag, as described in the appropriate terminology entries.

template A MEDFORD-syntax file that need not itself fully conform to the MEDFORD language specification. A template is intended for commonly-reused metadata across a number of projects. Specifically, a template allows for placeholder fields which must be replaced with content in order for the resulting file to be validated.

token A series of alphanumerical characters, not including any whitespaces, underscores, or dashes.

validate The act of the medford software ensuring that a MEDFORD file follows all syntactical requirements, with no modification of the original MEDFORD file.

2. Syntax

MEDFORD places all metadata into a single MEDFORD file. This MEDFORD file can be stand-alone, There are very few critical components of a MEDFORD file. The rules are as follows:

- Any line that consists of only white space is ignored.
- Any line that begins with # is ignored.
- A line that begins with @ is parsed as a metadata line.
- A line with a leading character that is neither #, @ or ‘@ are considered extensions to the metadata being described in the previous line.
- The series of symbols [..] is reserved as a template marker. Upon encountering an instance of this series of symbols, the medford parser shall flag it as an unfinished template and request the user to fill it out; a file containing this token is a MEDFORD template but not a valid MEDFORD file.
- The symbol $$ is reserved as a flag for LaTeXmath mode, and the medford parser will ensure that they are opened and closed correctly, but nothing between the $$ tags is validated. The enclosed sequence of characters is passed through unmodified.
- The series of symbols ‘@ (backtick character followed by the @ character), indicates some form of macro usage, either definition or reference. This is explained in further detail in section 2.4.

2.1. Major and Minor Tokens

MEDFORD defines a **major token** as the first substring after an @ symbol, up to but not including a minus sign (dash) ‘-’ or underscore ‘_‘.

The major token defines the type of metadata being defined; examples are ‘Contributor,’ ‘Data,‘ or ‘Paper.‘ Associated with a major token may be multiple entries that include **minor** tokens, which further define the data attributes associated with that data type. Examples would be ‘Contributor-name,’ ‘Contributor-ORCID,’ or ‘Paper-title.’

2.2. MEDFORD versions

The version of the MEDFORD language specification followed by a particular MEDFORD file can be indicated by the @Version tag. This indicates a revision of the MEDFORD language as opposed to adjustments to the medford parser. The medford parser will automatically add its current version to the BagIt bag if it was not present in the source MEDFORD file.

2.3. Data Provenance Tokens

All MEDFORD files are defined in reference to a BagIt bag, although the special use-case of an empty bag is common and acceptable. The BagIt bag binds a set of files to the MEDFORD file according to the BagIt standard, where these files describe a variety of resources, including source code, scientific papers, or raw data, each represented by a major tag in the MEDFORD file. The versioning and origins of that file are marked using a secondary major tag, where the tag can represent that the bag is considered to be the primary and authoritative source for the data or resource. Other secondary major tags describe the file as either a copy of an existing source, or simply a pointer to a URI where the resource can be obtained.

@Data_Primary	@Code_Primary	@Paper_Primary
@Data_Copy	@Code_Copy	@Paper_Copy
@Data_Ref	@Code_Ref	@Paper_Ref

MEDFORD's place in a BagIt directory structure is that the MEDFORD (.mfd) file is placed at the top level of the bagit directory structure. Any files carried along in the BagIt archive exist as Copy or Original directives (whether Data, Code, or Paper). The BagIt files.txt manifest refers to these files and their checksums. In contrast, any files only referred to using Ref directives are not listed in the BagIt manifest, as they are not considered to be part of the bag.

@Data_Primary and @Data_Copy both refer to resources that have been packaged with the MEDFORD metadata file, and should be available from the bag in a self-contained fashion, without having to visit external sources. From the point of view of the bag itself, there is no difference between these two tags; the difference is based on user context: @Data_Primary means that the BagIt bag is considered to be the primary and authoritative source for the data or resource; @Data_Copy means that BagIt has placed a copy of the data or resource into the bag, but that it does not claim the primary role. Finally @Data_Ref refers to DOIs, URLs, or other pointers to data or resources that are not placed in the bag, but rather represent external databases or resources.

2.4. Macros

MEDFORD supports a simple macro-expansion feature in order to save repetitive typing. In MEDFORD, a macro is defined by specifying a backtick (‘), @, a one-token name, and the macro body (which can contain multiple lines, ending before the next statement terminator, which could begin another macro definition or could be a regular token). For example, in lieu of typing their institute five times to document each of their collaborators at the same institution, they can define a macro as follows: ‘@myinstitute 100 Institute Drive, State, Zip

Subsequent instances of ‘@myinstitute will be replaced with 100 Institute Drive, State, Zip

In the event that a macro expansion needs to be concatenated with either another macro or some static content, MEDFORD supports the use of curly brackets ({}) for string interpolation. For instance, consider the following example:

```
`@location Sabago Isthmus

@Species-Loc `@{location}, Panama
```

will be expanded to

```
@Species-Loc Sabago Isthmus, Panama
```

2.5. Templates

As MEDFORD files are plain text, they can naturally be reused as easily as any other source code. However, for further reuse, MEDFORD provides an ‘‘invalid value’’ template token, which can be used to force users of a template to fill it out with complete information. The MEDFORD parser would require the user to fill in the specific placeholders ([..]) prior to validation. This eliminates the possibility that a researcher could accidentally leave a value for an older version of the template, further error-proofing MEDFORD templates.

An example of this template token is below:

```
@Species Pocillopora damicornis
@Species-Loc Sabago Isthmus, Panama
```

```
@Species-ReefCollection [..]
@Species-Cultured University of Miami Coral Resource Facility
@Species-CultureCollection [..]
```

2.6. Currently Defined Tags

This describes the tags that are considered pre-defined in the current version of MEDFORD (v. 0.9) Pre-defined tags may also have pre-defined minor tags, which are called 'expected' tags. These may or may not be required, only that the MEDFORD parser is aware of their existence and may be able to validate them. For instance, even if a Date minor tag is not required, it can till be validated for formatting. Minor tags which are not pre-defined are still allowed. Note that some tags are required only by certain target formats, such as BCO-DMO, and may only be validated when compiling to such a format.

@Contributor This defines an individual person who has contributed to a resource. A minor token, such as @Contributor-Role, indicates the role this individual has in the resource, such as Author or Maintainer.

Expected minor tokens include:

- ORCID: The ORCID of the contributor, if applicable. In future editions, this will be used to automatically resolve or validate several other @Contributor minor tokens.
- Association: The name and location of an institution the contributor is associated with.
- Role: The contributor's role in the study. Pre-defined values include Author, First Author, and Corresponding Author, but any additional roles may be used in addition to these.
- Email: An email the contributor may be reached at. In the case of a Corresponding Author, this field is required.

@Data This defines a data resource, which might be available online via a URI, or might be contained locally in the BagIt bag.

If a copy of the data is being provided alongside the MEDFORD file, it will require all @File minor tokens. In addition, all @Data major tokens expect the following minor tokens:

- Type: The type of the data, such as 'png'.

@Date This defines a date, such as the time a resource was downloaded or the publication date of a paper. The metadata associated with a @Date must be in UTC format (ISO-8601).

Expected minor tokens include:

- Note: A description of what the Date refers to, such as 'sample collection'.

@Expedition This defines a seagoing voyage or other voyage that typically resulted in data collection; this information is required by the BCO-DMO portal.

Expected minor tokens include:

- ShipName: As defined by BCO-DMO.
- CruiseID: As defined by BCO-DMO.
- MooringID: As defined by BCO-DMO.

- DiveNumber: As defined by BCO-DMO.
- Synonyms: As defined by BCO-DMO.

For an @Expedition block to be valid, it must either have both a ShipName and a CruiseID, or a MooringID, or a DiveNumber for identification purposes.

@File This defines a more general form of @Data, which is a catchall for any file a user may want provided within their bag, despite not necessarily being Data or Software. Examples include publication-quality photographs or research notes.
Expected minor tokens include:

- Path: The location of the file relative to the MEDFORD file, if it is available alongside the current MEDFORD file.
- Destination: An optional description of a bag-relative location to copy the file into when generating an output Bag.
- URI: A URI to the location of the file, if it is not available alongside the current MEDFORD file.

@Funding This defines a funding source for a paper or dataset; it could be a grant from a US federal funding agency such as NSF or NIH, or another nation's funding agency such as Canada's CIHR, or a private agency.
Expected minor tokens include:

- ID: The ID of the grant that funded the research described within the MEDFORD file.

@Journal This defines a journal in which the research described in the relevant MEDFORD file has been published.
Expected minor tokens include:

- Volume: Self-explanatory.
- Issue: Self-explanatory.
- Pages: Self-explanatory.

@Keyword This defines a keyword that is relevant to the research being described in the MEDFORD file, for discovery purposes.
This major tag has no expected minor tags.

@Method This defines a research methodology used to produce a dataset; it might refer to RNASeq technology, mass spectrography, photography, or any other method of collecting research data.
Expected minor tokens include:

- Type: The type of the method being described, such as 'sequencing' for a method used for DNA sequencing.
- Company: The company the method materials were purchased from.
- Sample: A descriptor of what samples were obtained using this method, such as 'healthy' to refer to control samples and 'bleach' to refer to bleached coral samples. This field is only meaningful when combined with the data provided alongside the MEDFORD file.

@Paper This defines a published manuscript, which may include a preprint on a server such as the arXiv, or a peer-reviewed paper in a conference or journal.

If a copy of the paper is being provided alongside the MEDFORD file, it will require all @File minor tokens. In addition, all @Paper major tokens expect the following minor tokens:

- Link: A URI to the published version of the paper that is available online.
- PMID: the paper's pubmed ID, if applicable.
- DOI: the paper's digital object identifier, if it is available.

@Software This defines a software package used in the production or analysis of a dataset, and should specify a version number (or other method of identification such as a Git hash) whenever possible.

If a copy of the software is provided alongside the MEDFORD file, it will require all @File minor tokens. In addition, all @Software major tokens may expect the following minor tokens:

- Type: A descriptor of the type of software being described, such as 'R package' or 'Python script'.
- Version: A version identifier to specify what version of the software was used.

@Species This defines a coral species that was studied, providing information such as reef location and culture growth site.

Expected minor tokens include:

- Loc: A description of the location where the coral species can be found naturally.
- ReefCollection: The date the coral was sampled from the wild.
- Cultured: A description of the location where the coral samples were cultured in a controlled environment.
- CultureCollection: The date the coral was sampled from a controlled culture.

@Version This defines important version information for the MEDFORD file. While not necessarily required for successful validation, it is strongly recommended and will be included in any MEDFORD file generated by the parser.

There are currently no defined expected minor tokens for the Version major tag.

2.7. User-Defined Tags

Any tags not pre-defined by MEDFORD are simply passed through by the medford parser, and not subject to any validation. We refer to these as user-defined tags. Should a user wish to implement custom validation for a user-defined tag, they would need to extend the medford parser (essentially, a language extension). This will be documented in a future version of MEDFORD.

3. Further Reading

The official source for the medford parser is on GitHub at https://github.com/TuftsBCB/medford. Additionally, there is a repository of example MEDFORD files on GitHub at https://github.com/TuftsBCB/MEDFORD-examples/. Finally, a preprint of the MEDFORD paper is on the arXiv at https://arxiv.org/abs/2204.09610

4. Acknowledgements

LC and PS thank the NSF for support under grant OAC-1939263. We thank Hollie Putnam, Jane Greenberg, Jack Freeman, Jay-Miguel Fonticella, and the entire Tufts BCB group for helpful feedback.