Helmholtz Centre Potsdam
GFZ German Research Centre
for Geosciences

Oliver Ritter, Gerard Muñoz, Roxana Barth,
Kristina Tietze, Paula Rulff, Sophie Stephan

MT Repository

User manual

Scientific Technical Report STR19/06

www.gfz-potsdam.de

HELMHOLTZ
RESEARCH FOR GRAND CHALLENGES

Recommended citation:

Ritter, O., Muñoz, G., Barth, R., Tietze, K., Rulff, P., Stephan, S. (2019): MT Repository - user manual, (Scientific Technical Report STR ; 19/06), Potsdam: GFZ German Research Centre for Geosciences.
DOI: http://doi.org/10.2312/GFZ.b103-19065

Supplementary datasets:

Ritter, O., Muñoz, G., Klose, R. (2019): Exemplary Data and Code Snippets for the Magnetotelluric Repository of the Geophysical Instrument Pool Potsdam (V. 1.0), GFZ Data Services.
DOI: http://doi.org/10.5880/GIPP-MT.0001

Imprint

HELMHOLTZ CENTRE POTSDAM
GFZ GERMAN RESEARCH CENTRE
FOR GEOSCIENCES

Telegrafenberg
D-14473 Potsdam

Published in Potsdam, Germany
May 2019

ISSN 2190-7110

DOI: http://doi.org/10.2312/GFZ.b103-19065
URN: urn:nbn:de:kobv:b103-19065

This work is published in the GFZ series Scientific Technical Report (STR)
and electronically available at GFZ website www.gfz-potsdam.de

This work is licensed under a Creative Commons Attribution 4.0 International License.
(CC BY 4.0) http://creativecommons.org/licenses/by/4.0/

Oliver Ritter, Gerard Muñoz, Roxana Barth, Kristina Tietze, Paula Rulff, Sophie Stephan

MT Repository

User manual

Scientific Technical Report STR19/06

MT Repository - user manual

Authors

Oliver Ritter, Gerard Muñoz, Roxana Barth, Kristina Tietze, Paula Rulff, Sophie Stephan

GFZ German Research Centre for Geosciences

Revision History

- 0.1 - April 2016
- 0.2 - February 2017 (major restructuring)
- 0.3 - March 2018 (inclusion of instrumentation.xml)
- 0.4 - April 2018 (introduction of the 2.0 versions of project.xml and sites.xml)
- 0.5 - October 2018 (introduction of the 2.0 versions of "emerald".xml, expansion of the "recorder".xml files)
- 0.6 - December 2018 (major rewrite and inclusion of report.html)
- 1.0 - April 2019 (Publication as Scientific Technical Report 19/06)

Table of Contents

List of Figures

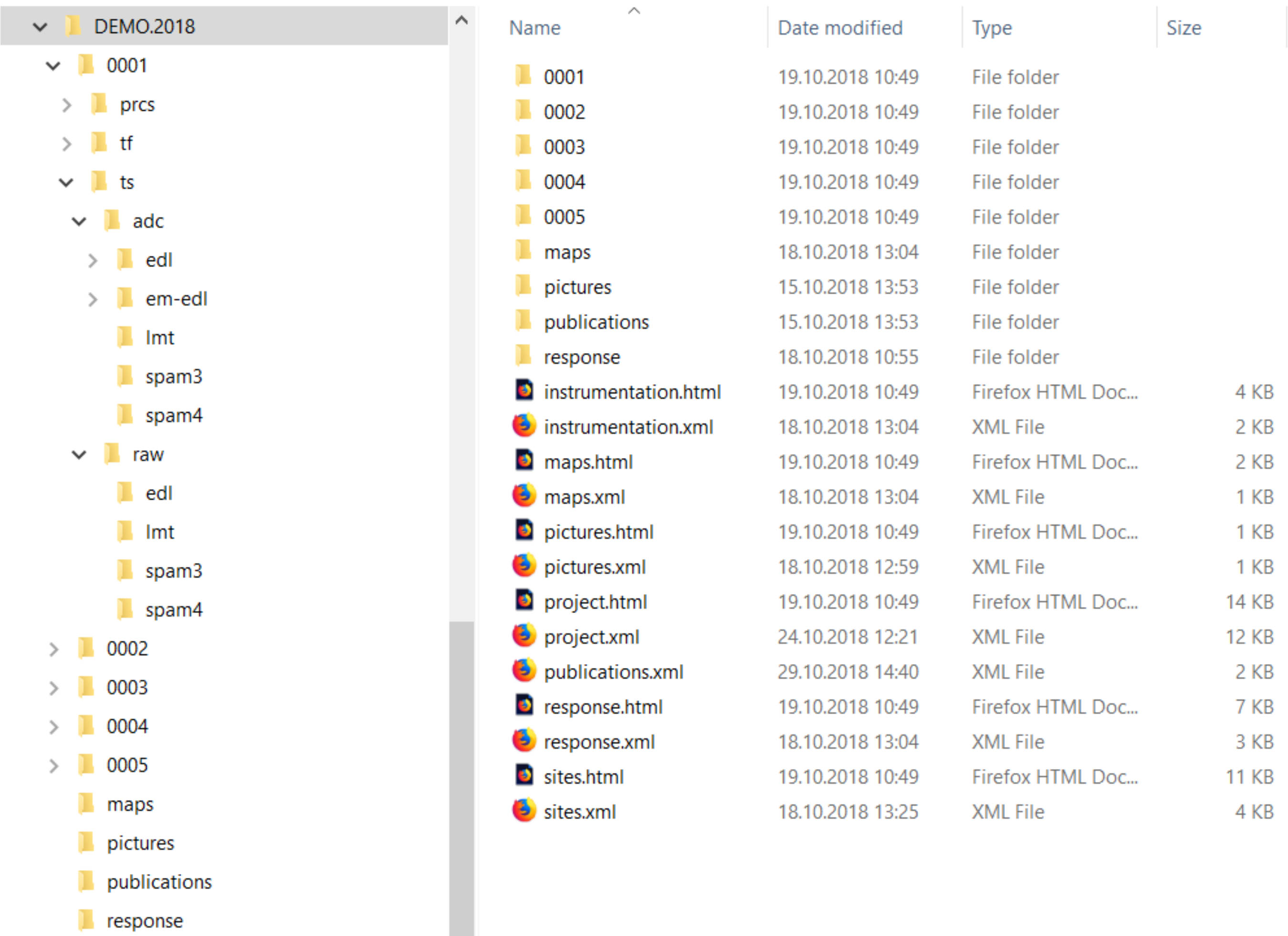
---

1 Introduction

The MT repository contains geophysical data sets collected in field experiments from all over the world. The acronym MT stands for magnetotelluric, a geophysical method used to probe the Earth's deep interior for its electrical conductivity distribution through electromagnetic (EM) induction. MT is based on EM fields generated by natural processes in the Earth's atmosphere and magnetosphere. But the repository also contains data from Controlled Source Electromagnetic (CSEM) projects, for which man-made EM sources are used.

The principle form of data in the repository are time-series of EM field components acquired with heterogeneous sets of sensors, recording instruments, and sampling rates. It is the main purpose of this archive or repository to provide the links between the data and their physical meaning by means of meta-data. To achieve this, the repository is organized as a combination of data files and associated meta-data in a well defined folder (directory) structure, with the data files being sorted into subfolders. Meta-data are provided as *XML* (Extensible Markup Language) formatted file.

The MT repository was originally developed to archive data collected with instrumentation provided by the Geophysical Instrument Pool Potsdam (GIPP) of the GFZ - German Research Centre for Geosciences. GIPP instruments are provided to the international research community free of charge but with the obligation to archive and eventually publish the data. Therefore, the archive contains data of almost all projects carried out by the Geo-Electromagnetic and Magnetotelluric working groups of GFZ Potsdam, together with projects from external universities and organizations also using GIPP instruments.



Folder structure of the MT data repository. Data files and associated meta-data are sorted into subfolders. A web browser is used to navigate through the folder structure and to access the meta-data.

The contents of the entire data repository, including the meta-data, can conveniently be viewed and accessed using a web browser. This document is also optimized to be used with a web browser: Use the internal links to navigate through the document. To return to the original position of the text after following an internal link, use the back-key of your browser. Clicking the [toc] symbol will lead you to the table of contents, selecting the [up] symbol will bring you to the top of the current section.

We also provide exemplary data with this documentation to demonstrate the structure and organization of the repository. All files and folders of project *DEMO.2018* are published by Ritter et al. (2019) and can be downloaded from our GitLab server.

--

2 Quick guide (getting your data into a repository)

2.1 General considerations - starting up

- Clean your data: Keep only those data, which are useful for data processing. Remove data from test runs, with wrongly installed sensors, etc.
- Create a folder for your project, following the naming convention: *ShortNameOfProject.YEAR* (for example: *BARBERTON.2010*).

2.2 Copying your data

2.2.1 *SPAM4* data in *EMERALD* format

- If the data were collected with a SPAM4 system, they are already organized in the correct folder structure, i.e. *.../site/ts/adc/spam4/YYYYdoy/data-files* (with *doy*=day of the year, *YYYY*=year) and the data files are in EMERALD format. Copy the entire folder-tree with the times series data of all sites into the project folder.
- Optionally (but recommended) copy the map and picture files to the corresponding maps and picture folders. Use descriptive filenames rather than what was created by cameras or e.g. *map1.jpg* as the links in the *pictures.xml* and *maps.xml* files will show these names. You can change them later in the

corresponding *.xml* file by modifying the <info> tag.

- Once all data files have been copied you must create or supply the meta-data as *.xml* files. Typically you will use PowerShell script *ArchiveCreateXMLs.ps1* to create the following files: *project.xml*, *sites.xml*, *pictures.xml*, *maps.xml*, *instrumentation.xml*, *response.xml*, *publications.xml* and for CSEM projects also *transmitters.xml*. These files contain useful and necessary information describing the data and give references to associated information in subfolders. Most of the generated *.xml* files are only templates which must be edited to provide relevant information, i.e. copy images of maps to the maps folder, then include respective tags in the *maps.xml* file.

- In each subfolder of a site, the script *ArchiveCreateXMLs.ps1* also generates *comment.xml* and *config.xml* files. Edit *comment.xml* to provide additional information specific for that site. For known data types (e.g. SPAM4 and EDL, the script creates additional files automatically. *config.xml* lists all hardware configurations and recording times used for that particular site. Within the *.../ts/adc* subfolder, the script creates an inventory of all recorded data (e.g. spam4.xml).

2.2.2 *SPAM4* data for Controlled Source EM projects

- For the receivers proceed as described above.

- For CSEM projects, the script *ArchiveCreateXMLs.ps1* generates the file *jobs.xml* to provide additional transmitter specific information from the transmitter files.

2.2.3 *EDL* data in *MINISEED* format

- If the data are collected with another recording system, e.g. the EarthData logger (EDL), you must create an according folder structure for each site: *.../site/ts/adc/edl*. Alternatively use the Powershell script *ArchiveMakeStructure.ps1*. If the data are organized as daily records, create a subfolder for each day as *YYYYdoy*, with *doy*=day of the year and *YYYY*=year. Copy your time series data into the respective folders and repeat until the data of all sites are copied. See also the information provided below on the sites.xml files. Note, to archive EDL data, a preliminary *.xtr/.xtrx* file is required which was created in conjunction with the CASTLE sensor boxes and which contains information on station setup. This *xtr/.xtrx* file must reside in the *.../ts/adc/edl* folder.

- Optionally (but recommended) copy the map and picture files to the corresponding maps and picture folders. Use descriptive filenames rather than what was created by cameras or e.g. map1.jpg as the links in the *pictures.xml* and *maps.xml* files will show these names. You can change them later in the corresponding *.xml* file by modifying the <info> tag).

- To archive older data which were recorded in MINISEED format such as EDL/LMT/BURST, run *ArchiveCreateXMLs.ps1* with the command line parameter *-generateEmeraldAdc* to create *em-edl*, *em-spam3* or *em-lmt* folders within the *.../ts/adc* folder tree structure. These folders contain converted *.raw* and *.xtrx*-files in the most recent EMERALD format. Corresponding *.xml* files are created automatically.

- Note that MINISEED files do not contain coordinates. In this case, the conversion program used to generate the *em-** files (see recorder.xml) reads the coordinates from the sites.xml file (which must have been created beforehand).

- To make sure the coordinates appear correctly in the *.xtrx* files, it can be necessary to first call *ArchiveCreateXMLs.ps1* without parameter *-generateEmeraldAdc*. After having checked all site coordinates in *sites.xml* (possibly updating them with *ArchiveUpdateSitesXML.ps1* using a coordinates file (*.crd*, see detailed help of script for more information), rerun *ArchiveCreateXMLs.ps1*, but now with the parameter *-generateEmeraldAdc* to convert the data. In this case, it is advisable to use the parameter *-partialRun* for the first call of *ArchiveCreateXMLs.ps1* to avoid creating a partial set of data (the *em-** data have not been created yet!) and therefore incomplete *config.xml* files. Rerun *ArchiveCreateXMLs.ps1* (after making sure *sites.xml* contains the correct coordinates) without the *-partialRun* parameter.

2.2.4 *SPAM3* data

- Follow the instructions for *EDL* data above. Make sure to run *ArchiveCreateXMLs.ps1* with the command line parameter *-generateEmeraldAdc* to create *em-spam3* folders within the *.../ts/adc* folder tree structure.
- The *SPAM3* format files already contain the coordinates, therefore the three-step process described for the *EDL* data is not necessary

2.2.5 *SPAM4* data in old proprietary format

- Follow the instructions for *EDL* data above. Make sure to run *ArchiveCreateXMLs.ps1* with the command line parameter *-generateEmeraldAdc* to create *em-spam4* folders within the *.../ts/adc* folder tree structure.
- Depending on the firmware version used to record the data, the headers may or may not contain coordinates. If the *SPAM4*-headers contain the coordinates, proceed as described for _SPAM3_data (a single step with the parameter *-generateEmeraldAdc* is sufficient). If the headers do not include coordinate information, a three step process (one run with the parameter *-partialRun*, an intermediate call to *ArchiveUpdateSitesXML.ps1* using a coordinates file, and a second run with the parameter *-generateEmeraldAdc*) is necessary.

2.3 Finnishing off - *project.xml* and *instrument.xml*

- Obtain the calibration files for all instruments used in the project. The calibration archive is administered by the GIPP-MT crew (contact Reinhard Klose). Calibration files (*.rspx*) matching the timeframe of the project will be provided. These files should then be copied to the *responses* subfolder. Alternatively, it is also possible to proceed initially without the *responses* folder and use the *instrumentation.xml* file instead, which is created by the *ArchiveCreateXMLs.ps1* script. In this case you will need to call the appropriate function manually (see description of *ArchiveCreateXMLs.ps1* for details).

- You will have to edit some of the *.xml* files to provide information that can not be automatically generated, particularly important is the *project.xml* file. Refer to the detailed description of the *.xml* files in section 5 to see what information is needed.

- If not done so already, copy/create the *xslt* folder with all *.xslt* files in the root of your repository (e.g. the same level as the project folders). If you double click on *project.xml* it should open as a *html* formatted webpage in your webbrowser, showing basic information on the project and links to your sites, maps, etc. Please note, it may be necessary to enable this functionality in your web browser (in Firefox: enter `about:config`, set `security.fileuri.strict_origin_policy` to false).

- Use the Powershell script ArchiveCreateHTMLs.ps1 to create *.html* files for all corresponding *.xml* files, so that project meta data can be also browsed/obtained from *.html* files.

- In case the script *ArchiveCreateXMLs.ps1* does not work and error messages like `incorrect encoding` occur, call the Powershell script *ArchiveUpdateEncoding.ps1*. The Script changes the encoding of all existing *.xml* files to *UTF-8*.

2.4 Updating an existing project

- Use Powershell script *ArchiveCleanXMLs.ps1* to clean all or selected *.xml* files in case you want to redo an archive.

- Select the project folder you want to update and copy its address. Open a Powershell console and change to this location.

- Use the Powershell scripts *ArchiveUpdateProjectXML.ps1* and *ArchiveUpdateSitesXML.ps1* to update *project.xml* and *sites.xml*, e.g. from version 1.x to version 2.0 (or 2.1 respectively).

- Edit the *project.xml* file to update any relevant information (particularly required information that was not included in the original (version 1.x) files, which cannot be retrieved automatically. Please refer to description in *project.xml*.

- In projects for which the repository structure was created a long time ago, some of the newer *.xml* files can be missing (e.g. *instrumentation.xml*, *response.xml* or even *config.xml* in the site folders). In this case, use Powershell script *ArchiveCreateXMLs.ps1* to create the missing *.xml* files; only missing files will be created, existing ones will remain unchanged.

- It is also possible that the recorder data was not converted into EMERALD format, and therefore the *em-*.xml* files are missing. In this case, use PowerShell script *ArchiveCreateXMLs.ps1*, making sure to activate parameter *-generateEmeraldAdc* to convert the data and to create the corresponding *.xml* files.

- Note that MINISEED or old SPAM4 data files do not contain coordinates. In this case, the conversion program used to generate the *em-** files (see recorder.xml) reads the coordinates from the sites.xml file (which must have been created beforehand). Check first *sites.xml* for correct coordinates, then call *ArchiveCreateXMLs.ps1*. Missing coordinates can be added with *ArchiveUpdateSitesXML.ps1* using a coordinates (*.crd*) file. See the detailed help of the script for more information.

- Check any newly created *.xml* files or existing ones for additional input required. For example you might want to include captions for your maps or change the names of the pictures for better readability.
- Delete any existing *.html* files (you can also use *ArchiveCleanXMLs.ps1* and use PowerShell script *ArchiveCreateHTMLs.ps1* to)(re-)create *.html* files for all corresponding *.xml* files. Then the project meta data can be also obtained from *.html*-files.

---

3 The data repository

The data repository is a collection of data files and associated meta data organized in a folder (or directory) tree structure. A repository consists of one or many projects and each project contains one or many sites. Information and data for each site are stored in subfolders containing time series data, which can be further subdivided into folders per day, etc

Details of the file and folder structure are given below. A number of tools (Powershell scripts) are available to help create and maintain this structure.

3.1 XML (Extensible Markup Language)

All meta-data are stored as plain ASCII files which are internally structured using the Extensible Markup Language (XML).

The characters making up an *XML* document are divided into markup and content. Strings that constitute markup usually begin with the character < and end with a >. Strings of characters that are not markup are content. A `tag` is a markup construct with start-tags `<section>`, end-tags `</section>` or empty-element tags `<line-break />`.

A logical document component begins with a start-tag and ends with a matching end-tag. The characters between the start- and end-tags, are the element's content, and may contain markup, including other elements, which are called child elements. An example of an element is `<Greeting>Hello, world. </Greeting>`.

Attributes are markup constructs consisting of name/value pairs that exists within a start-tag. In the example `<step number="3">Connect A to B.</step>`, the element step has the attribute *number* with the value *3*. Everything between `<!-- this -->` is a comment.

For more information on XML please refer to: https://en.wikipedia.org/wiki/XML and references therein.

3.2 XSLT (Extensible Stylesheet Language Transformations)

XSLT (Extensible Stylesheet Language Transformations) is a language for transforming *XML* documents into other formats such as HTML for web pages. The original document is not changed; rather, a new document is created based on the content of an existing one. The *XSLT* processor (of a web browser) takes *XML* source documents, plus one or more *XSLT* style sheets, and processes them to produce an output document (e.g. in *.html* format).

To automatically apply an *XSL* transformation to an *XML* document on display, an *XML* style sheet processing instruction must be inserted into XML:

```
<?xml-stylesheet href="example2.xsl" type="text/xsl" ?>
```

In the present text there will be several examples of *xml* files (or snippets) and the associated representation in the browser (converted to *html* format). Note that any links shown in the *html* version are created by the corresponding *xslt* stylesheet assuming the underlying file and folder structure of the repository. Clicking on any link in these examples will not open any file or url.

---

4 Main entry to the repository (repository.xml file)

The main entry to the MT/CSEM data repository is an *.xml* file keeping track of all existing projects. For convenience we refer to it as *repository folder* but this data file can be anywhere on a computer which has access to the actual locations of the data, typically a file server (or several of them).

An exemplary *repository.xml* file could look like this:

```xml
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="/////mt42/MT_Projects/xslt/mtprojects.xslt" ?>
<!-- Description of MT Projects in archive: File name convention: PROJECT.YEAR -->
<PROJECTS>
  <VERSION> 2.00</VERSION>

  <years>
  <year>2018</year>
    <project>
      <host>computer_name</host>
      <name>PROJECT1.2018</name>
      <info>project description</info>
      <status>status information</status>
      <modified>dd-mm-yyyy</modified>
      <user>User Name</user>
    </project>
    <project>
      <host>MT43</host>
      <name>DEMO.2018</name>
      <info>Exemplary project used for documentation</info>
      <status>Final</status>
      <modified>30-10-2018</modified>
      <user>Oliver Ritter</user>
  </project>
  </years>
</PROJECTS>
```

But when viewing above *xml* file in your web browser, it should look like this:

Magnetotelluric Data Repository

Helmholtz Centre Potsdam
GFZ German Research Centre for Geosciences
Version: 2.00

Availabe projects:

----- 2018 -----

PROJECT1.2018: **project description**
host: computer_name - **status:** status information - **last modified:** dd-mm-yyyy

DEMO.2018: **Exemplary project used for documentation**
host: MT43 - **status:** Final - **last modified:** 30-10-2018

---

end of repository

The `<?xml-stylesheet` directive the *repository.xml* file refers to: *href="...path.../mtprojects.xslt"* for *XSLT* formatting; *...path...* is a valid path on your computer, e.g. */////mt42/MT_Projects/xslt* on a Windows system (mt42 is a computer name). Any relative paths are accepted but URLs (i.e. on a web server) are not permitted.

Please note, it may be necessary to enable this functionality in your web browser (in Firefox: enter `about:config`, set `security.fileuri.strict_origin_policy` to false).

The style sheet of the example above looks like this:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:strip-space elements="name" />
<xsl:template match="/">
  <html>
    <body>
    <p align="left" style="font-family:Calibri; font-size:24px; color:red">
    <b>Magnetotelluric Data Repository<br></br></b>
    </p>

    <p align="left" style="font-family:Calibri; font-size:16px; color:Teal">
      <b>
        Helmholtz Centre Potsdam<br></br>
        GFZ German Research Centre for Geosciences
      </b>
      <br></br>Version: <xsl:value-of select="PROJECTS/VERSION" />
    </p>

    <p align="left" style="font-family:Calibri; font-size:16px; color:black">
      <b>Availabe projects:</b>
    </p>
    <hr></hr>

    <p align="justify" style="font-family:Calibri; font-size:14px; color:black">
      <xsl:for-each select="PROJECTS/years" >
        <p align="left" style="font-family:Calibri; font-size:14px; color:green">
        ----- <xsl:value-of select="normalize-space(year)" /> -----
        </p>
        <hr></hr>
        <xsl:for-each select="project" >
          <xsl:variable name="_host" select="host" />
          <xsl:variable name="_name" select="name" />
          <xsl:variable name="_status" select="status" />
          <xsl:variable name="_modified" select="modified" />
          <a>
            <xsl:attribute name="href">
            <xsl:text>file://///</xsl:text>
            <xsl:value-of select="normalize-space($_host)" />
            <xsl:text>/MT_Projects/</xsl:text>
            <xsl:value-of select="normalize-space($_name)" />
            <xsl:text>/project.xml</xsl:text>
            </xsl:attribute>
            <xsl:value-of select="$_name" />
          </a>:
          <b><xsl:value-of select="info" /></b> <br></br>
          <b>host: </b><xsl:value-of select="host" /> -
```

```
          <b>status: </b> <xsl:value-of select="status" /> -
          <b>last modified: </b><xsl:value-of select="modified" />
          <br></br><br></br>
        </xsl:for-each>
        <hr></hr>
      </xsl:for-each>
      </p>
    <p align="left" style="font-family:Calibri; font-size:16px; color:black">
      <b>end of repository</b>
    </p>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

The exemplary project used for this documentation, called *DEMO.2018*, is a virtual project. Its sole purpose is to demonstrate the structure and organization of the repository. The data included may have been collected with real instruments and sensors but they are random snippets of larger data sets. All coordinates given are exemplary; they point to the location of the GFZ in Potsdam, Germany. All files and folders of *DEMO.2018* are published in Ritter et al. (2019) and can be downloaded from our GitLab server.

---

5 Folder structure of a project

To add a new project choose a name which follows the naming convention *NAME.YYYY*. It is a combination of a short name for the project and the year of data recording. In accordance with the exemplary *repository.xml* file above, we named the project folder *DEMO.2018*. In combination with the according *repository.xslt*, *repository.xml* is translated into *.html* format and all links displayed in the *repository.xml* file are resolved correctly, i.e. to open the file *project.xml* in folder *DEMO.2018*. If available (see *ArchiveCreateHTMLs.ps1*), you can open the *.html* files instead.

The project folder contains all data files together with the meta-data (*.xml* files). All data are organized in a predefined folder tree structure. The root of the tree or project folder will contain the following files and subfolders:

- *project.xml* - There is one *project.xml* file in each project folder to provide general information on that particular project.
- *sites.xml* - There is one *site.xml* file per project. For each recorded site of a project a unique site folder exists, usually made up of 4-digit site numbers. The data of each site is stored in subfolders under the according site names.
- *pictures.xml* - lists the contents of the *pictures* folder: photos of fieldwork, site locations, packing/cleaning, field office, group photos, data charts, etc.

- *maps.xml* - lists the contents of the *maps* folder: overview map of sites location, regional geology, or any other relevant information. For version 1.2 or later of the *maps.xml* file, the maps folder contains automatically created files *station map.html* and *station map.png* showing an overview of all sites found in the *sites.xml* file.

- *response.xml* - lists the contents in the *response* folder: response files (*.rsp* and *.rspx* formats) required for processing and a *responseinfo* file which lists all required response file.

- *instrumentation.xml* - provides a listing of used instruments for this project (types and IDs of data loggers, induction coils, electrodes, sensor boxes, etc.).

- *publications.xml* - contains relevant publications related to the project.

- *transmitters.xml* - provides an overview of all current transmitter locations used in a project (CSEM only).

- *scripts* folder: (optional) contains scripts used for data processing. This folder is useful if data processing results are to be reproduced at a later time.

Note, a Powershell script *ArchiveMakeStructure.ps1* is available to create this basic underlying folder tree structure.

5.1 project.xml

The *project.xml* file contains a number predefined mandatory and optional tags, which **must** be provided by the user. **The information supplied in the project.xml file is essential for the integrity of the archive and a meaningfull data publication**. Please take the necessary time to properly describe your project. Please also see the comments in the listing below.

Versions:

- 1.00: Original definition.
- 1.10: Included project type.
- 1.20: Included GIPP number, grant numbers and allowed for several responsible scientist(s).
- 2.00: Active since February 2018. Organization in `<proj.section>` tags.

```xml
<?xml version="1.0" encoding="utf-8"?>
  <?xml-stylesheet type="text/xsl" href="../xslt/project2.xslt"?>
  <PROJECT>
    <VERSION>2.00</VERSION>
    <!--The following fields are all mandatory, please fill them in-->
    <USER></USER>
    <proj.name></proj.name>
    <proj.type></proj.type>
    <proj.long.name>Enter project long name</proj.long.name>
    <proj.gipp.number>Enter GIPP number</proj.gipp.number>
    <proj.start.date>Enter project starting date</proj.start.date>
    <proj.end.date>Enter project end date</proj.end.date>
    <proj.country>Enter project country</proj.country>
    <proj.region>Enter project region</proj.region>
```

```
<ref.wgs84.lat.dec>Enter reference latitude</ref.wgs84.lat.dec>
<ref.wgs84.lon.dec>Enter reference longitude</ref.wgs84.lon.dec>
<ref.altitude.m>Enter reference altitude</ref.altitude.m>
<!--At least one keyword is mandatory, add more if needed or
delete the second one-->
<proj.keyword>Keyword 1</proj.keyword>
<proj.keyword>Keyword 2 (copy or delete as necessary)</proj.keyword>
<!--At least one responsible scientist is mandatory, add more if you need.
These will count as authors of the data report-->
<proj.resp.scientist
  <name>Responsible scientist</name>
  <affiliation>
    <institute>Enter Institute</institute>
    <address>Enter address</address>
  </affiliation>
  <!--More than one affiliation is possible, per scientist.
  Add or delete as necessary-->
  <affiliation>
    <institute>Enter Institute</institute>
    <address>Enter address</address>
  </affiliation>
  <email>email@gfz-potsdam.de</email>
  <phone>Phone number</phone>
  <href>http://</href>
</proj.resp.scientist>
<!--The text is organized in several sections, marked by keywords.
Some of them are mandatory others are optional.-->
<!--Each section contains one <header> tag, where you can write the title of a
section. It may or may not coincide with the keyword.-->
<!--It also has one or more <text> tags. Write your text for the section here,
one paragraph per tag.-->
<!--Abstract Section. Mandatory !-->
<proj.section keyword="abstract">
  <header>Abstract</header>
  <text>Enter the abstract</text>
</proj.section>
<!--Introduction Section. Mandatory !-->
<proj.section keyword="introduction">
  <header>Introduction</header>
  <text>Enter the introduction</text>
</proj.section>
<!-- Experimental setup Section. This is optional, if omitted, a default text
 will be used in the report. -->
<!-- <proj.section keyword="setup">
    <header>Experimental setup and schedule</header>
    <text>Enter the text</text>
</proj.section> -->
<!-- Station locations Section. This is optional, if omitted, a default text
will be used in the report. -->
<!-- <proj.section keyword="locations">
  <header>Station Locations</header>
  <text>Enter the text</text>
```

```xml
</proj.section> -->
<!-- Instrumentation Section. This is optional, if omitted, a default text
will be used in the report. -->
<!-- <proj.section keyword="instrumentation">
  <header>Instrumentation</header>
  <text>Enter the text</text>
</proj.section> -->
<!-- Recording Settings Section. This is optional, if omitted, a default text
 will be used in the report. -->
<!-- <proj.section keyword="config">
  <header>Recording settings</header>
  <text>Enter the text</text>
</proj.section> -->
<!-- Transmitter Section (for CSEM projects). This is optional, if omitted,
this section will not be included in the report -->
<!-- <proj.section keyword="transmitters">
  <header>Transmitter settings</header>
  <text>Enter the text</text>
</proj.section> -->
<!-- Data quality section. This is optional, if omitted, this section will
not be included in the report -->
<!-- <proj.section keyword="quality">
  <header>Data quality</header>
  <text>Enter the text</text>
</proj.section> -->
<!-- Data processing section. This is optional, if omitted, this section will
 not be included in the report -->
<!-- <proj.section keyword="processing">
  <header>Data processing</header>
  <text>Enter the text</text>
</proj.section> -->
<!-- References section. If you include any citations in your text (any section
 or map captions - found in maps.xml) you should include the reference here -->
<proj.section keyword = "references">
  <!-- References to EMERALD format and Wittstock already included -->
  <bibitem type = "techreport">
    <citation>Ritter et al. 2015a</citation>
    <doi>10.2312/GFZ.b103-15082</doi>
  </bibitem>
  <bibitem type = "techreport">
    <citation>Ritter et al. 2015b</citation>
    <doi>10.2312/GFZ.b103-15092</doi>
  </bibitem>
  <bibitem type = "misc">
    <citation>Ritter et al. 2015c</citation>
    <doi>10.5880/GFZ.2.2.2015.001</doi>
  </bibitem>
  <!-- You can include just a citation (for reference only) and doi
  (it will be resolved later by ArchiveUpdateProjectXML) -->
  <bibitem>
    <citation></citation>
    <doi></doi>
```

```
  </bibitem>
  <!-- If no doi is available, please introduce all the information in the
   bibitem (bibXML convention) -->
  <bibitem type = "misc">
    <citation></citation>
    <author><firstname></firstname><lastname></lastname></author>
    <title></title>
    <journal></journal>
    <year></year>
  </bibitem>
</proj.section>
<!-- Introduce the project participants according to their role
(copy or delete fields as necessary) -->
<!-- role="scientist" for participating scientists. These count as authors for
the report. Include affiliation just like for responsible scientists -->
<!-- role="fieldword" for participants in the fieldword (scientists excluded).
They will be mentioned in the acknowledgments -->
<!-- role="other" for other types of participation. They will be neither authors
 or included in the acknowledgements -->
<participant role = "scientist">
  <name></name>
  <affiliation>
    <institute></institute>
    <address></address>
  </affiliation>
  <href></href>
</participant>
<participant role = "fieldwork">
  <name></name>
</participant>
<participant role = "other">
  <name></name>
</participant>
<!-- Cooperation partners (include all institutions which participated
in the project) -->
<coop.partner>
  <name></name>
  <href></href>
</coop.partner>
<!-- Web pages related to the project (GFZ project website, international
project websites, etc.) -->
<web.resources>
  <name></name>
  <href></href>
</web.resources>
<!-- Funding agency (or agencies, in this case copy the field as needed).
Very important! Include the grant number if known! -->
<funding.agency>
  <name></name>
  <href></href>
  <grant></grant>
</funding.agency>
```

```
</PROJECT>
```

Most of the predefined tags in the exemplary project.xml file above are self-explanatory. Some additional information is given below:

- <VERSION> The version number is used to keep track of changes to the structure of this file. It can help solving compatibility issues, e.g. if new tags are defined or existing tags are renamed at a later time. For new projects use the version number as specified in the example above. The current Version is 2.0. Older version files can be updated with the script *ArchiveUpdateProjectXML.ps1*.
- <proj.name> Short name for the project. Must be identical to the name of the project folder, e.g. in the format *Project.Year.*
- <proj.type> Predefined project types are: *MT* , *CSEM* or *RMT*
- <proj.long.name> The complete name (not acronym) of the project
- <proj.gipp.number> The grant number provided by the Geophysical Instrument Pool Potsdam (GIPP) for the project. It can also be found at the GIPP website.
- <proj.start / end.date> The project start and end dates refer to data collection in the field.
- <proj.country / region> The geographical context of the project.
- <ref.wgs84.lat.dec / lon.dec> / <ref.altitude.m> Reference coordinates of the project (central point or a relevant reference: nearby city, important geological feature, etc)
- <proj.resp.scientist> At least one responsible scientist or principle investigator of the project should be named and contact details should be specified.
- <proj.keyword> At least one keyword is mandatory. You can include more if necessary. Use single words or very short sentences for the keywords.
- <proj.section keyword="..."> The project description provided in the *project.xml* file is organized as a series of sections identified by keywords, such as *abstract*, *introduction*, etc. Each section contains one <header> tag, where you can define the title of the section. It may or may not coincide with the keyword. Sections also have one or more <text> tags. Write one paragraph per <text> tag. The text in these sections will be displayed through the stylesheet and forms the backbone for the *report.xml*. Mandatory project sections are *abstract* and *introduction*, while *setup*, *locations*, *instrumentation*, *config*, *transmitters*, *quality*, *processing*, and *references* are optional. If omitted, a default text will be displayed in *report.xml*. These default text blocks can be empty; they are defined in template file *reportSnippets.xml* which resides in the *xslt* subfolder:
    - <proj.section keyword="abstract"> **Mandatory**. A concise description or summary of the project (e.g. the abstract of your proposal). The information supplied could include area of research, geographical description, main objectives, geodynamic context, etc.
    - <proj.section keyword="introduction"> **Mandatory**. A more detailed introduction to the project.

- `<proj.section keyword="setup">` **Optional**. A short description of the instrumental setup, station distribution (grids, profiles, ...), etc. Per default, all maps in the *maps.xml* file with the attribute *figure* set to a number are included as figures in this section, make sure to provide proper captions for them (this includes the automatically generated stationMap.png - see *maps.xml* - and any other maps in *maps.xml* that have a number in the @figure attribute). Additionally, a time frame for the experiment is given also per default.

- `<proj.section keyword="locations">` **Optional**. A list or table with the locations of all the sites. Per default, the report generator includes a short sentence introducing the table and the list of locations obtained from the *sites.xml* file.

- `<proj.section keyword="instrumentation">` **Optional**. A short description of the instruments and sensors used. Commonly used types are predefined in the reportSnippets.xml template file. Note, you do not need to worry about serial numbers, they will be collected automatically from the data (instrumentation.xml).

- `<proj.section keyword="config">` **Optional**. A short description of the overall recording configuration: frequency bands, writing schedule (e.g. 10 min every 1h, continuous 1h files, ...). Note, you don't need to worry about details as they are collected from the data (config.xml tables for each site). A summary will be created as an appendix in the *report.xml*.

- `<proj.section keyword="transmitters">` **Optional (no default)**. Only for CSEM projects - locations, distribution, and particularities of the current transmitters.

- `<proj.section keyword="quality">` **Optional (no default)**. A short description of the data quality. You can comment on noise sources, affected frequency bands, ...

- `<proj.section keyword="processing">` **Optional (no default)**. A short description of the processing steps used in the project: digital filters, the parameters for the robust stacking, etc.

- `<proj.section keyword="references">` **Mandatory**. References to any citations in your text. Could include citations to maps (e.g. in *maps.xml*)). Follow the convention for *BibXML* which is also used for the *publications.xml*. References to the EMERALD format and the Wittstock remote reference site are already included (simply delete the Wittstock references if irrelevant). You can simply supply a *doi* and use Powershell script *ArchiveUpdateProjectXML.ps1* to convert the *doi* into a full reference.

- `<participant role="...">` Introduce the project participants according to their role (copy or delete fields as necessary). Supplying names of participants in the project is useful for future reference: to obtain information on the project, particularities of the fieldwork, or simply to acknowledge help in the field. In updated projects, all the participants will have the three roles listed, just delete those which do not apply.

- `<participant role="scientist">` For participating scientists. These count as authors for the report. Include affiliation just like for responsible scientists.

- `<participant role="fieldwork">` For participants in the fieldword (scientists excluded). They will be mentioned in the acknowledgments.

- `<participant role="other">` For other types of participation. They will be neither authors or included in the acknowledgements.

- `<coop.partner>` Cooperation partners. List the institution(s) that participated in the project (other than the main institution, usually GFZ). You can include a web reference in the tag `<href>`.

- `<web.resources>` Web resources. List all the websites related to the project (GFZ project website, international project websites, etc.). Make sure to include the links in the tag(s) `<href>`.

- `<funding.agency>` Funding agencies. List all the funding agencies including personal, expedition, etc. Make sure to include the grant number for a proper reference in the report.

Make sure the reference to the associated *xslt* file can be resolved. Note, only relative paths can be used, e.g. href='../xslt/project.xslt'. If everything is ok, the *project.xml* file will be translated to *.html* and look like this in your web browser:

Details of project DEMO.2018

(Demonstration project to document the MT repository)
GIPP Number: 000000

Abstract

> The MT repository contains geophysical data sets collected with electromagnetic (EM) experiments. The acronym MT stands for magnetotelluric, a geophysical method used to probe the Earth's deep interior for its electrical conductivity distribution through EM induction. MT is based on EM fields generated by natural processes in the Earth's atmosphere and magnetosphere. But the repository also contains data from Controlled Source Electromagnetic (CSEM) projects. The principle form of data in the repository are time-series of EM field components acquired with heterogeneous sets of sensors, recording instruments, and hugely variable sampling rates. It is the main purpose of this archive to provide the links between the data and their physical meaning by means of meta-data. To achieve this, the repository is organized as a combination of data files and associated meta-data in a well defined folder (directory) structure.
>
> The MT repository was originally developed to archive data collected with instrumentation provided by the Geophysical Instrument Pool Potsdam (GIPP) of the German Research Centre for Geosciences - GFZ. GIPP instruments are provided to the (inter)national research community free of charge but with the obligation to archive and eventually publish the data. Therefore, the archive contains data of almost all projects carried out by the Geo-Electromagnetic and Magnetotelluric working groups of GFZ Potsdam, together with projects from external universities and organisations also using GIPP instruments.
>
> The project at hand, called DEMO.2018, is a virtual project. Its sole purpose is to demonstrate the structure and organization of the repository. The data included may have been collected with real instruments and sensors but they are random snippets of larger data sets. All coordinates given are exemplary; they point to the location of the GFZ in Potsdam, Germany.

Follow this link to view available MT sites .
Follow this link to view available maps .
Follow this link to view available pictures .
Follow this link to view all available response files.
Follow this link to view a summary of all instruments used in the project.
Follow this link to view related publications (papers, conference abstracts, academic theses, etc.)
Follow this link to view the revision history of the project report.

Field experiment:

> Country: Germany, Region: Brandenburg
> Field work lasted from 18.10.2018 to 18.10.2018
> Reference latitude: 52.383015
> Reference longitude: 13.064130
> Reference altitude [m]: 92.10
> Overview map of all sites.

Keywords:

> Archive, MT repository, Demonstration, Example

Responsible scientist(s):

Oliver Ritter
Deutsches GeoForschungsZentrum, Telegrafenberg, 14473 Potsdam
oritter@gfz-potsdam.de
0331 288 1257

Gerard Muñoz
Deutsches GeoForschungsZentrum, Telegrafenberg, 14473 Potsdam
gmunoz@gfz-potsdam.de
0331 288 28678

Other participating scientists:

Ute Weckmann, German Research Center for Geociences - GFZ
Kristina Tietze, German Research Center for Geociences - GFZ

Introduction

Welcome to the data repository of the MT component of the Geophysical Instrument Pool Potsdam. This Magnetotelluric (MT) and Controlled Source Electromagnetic (CSEM) data repository is organized as a combination of data files and associated meta-data. All files are stored in a specific folder (directory) structure, with the data files and meta-data being sorted into sub-folders. All meta-data are provided as XML (Extensible Markup Language) formatted files.

It is straight forward to access the repository. When selecting (clicking on) any of the xml- files or their html- formatted counterparts, its contents opens in a web-browser window. Internal links lead the way to all other parts of the repository and the back-key of the web-browser returns to the original position.

As mentioned before, this particular project serves as a practical demonstration example. The data may originate from a variety of instruments and locations around the world. For simplicity, all site locations were modified to show around the GFZ campus (Telegrafenberg) in Potsdam, Germany.

The file you are currently looking at (project.xml) contains the meta-data describing the project. It is perhaps the most important document of a project and it is this file which requires thorough attention of the project's responsible scientists. project.xml also forms the basis for report.xml, a comprehensive compilation of all information found in the repository for a particular project. For a detailed description of the data repository, please refer to Ritter et al. (2018) and references therein.

Over the years numerous colleagues actively participated in the effort developing this data repository and even more in collecting data sets. For the purpose of this demonstration project, the names of the more substantial contributors appear randomly under various headings such as "Responsible scientists, Participant in the field work, etc."

Experimental setup and schedule

Since this is but a virtual experiment, there is no meaningful experimental setup. Expected useful information in this section would include:

Magnetotelluric data were collected between MONTH YEAR and MONTH YEAR along a XX km long profiles / a XX km wide and YY km wide grid.

Data were recorded with RECORDER-TYPE instruments, and SENSOR-TYPE sensors. Data recording lasted typically for XX days per station.

Data acquisition settings

Recording modes of the instruments used for the demo range from high to low frequency and vary from site to site to illustrate how the various data types are organized. Some of the data were recorded in proprietary data formats according to the instruments used. But all data sets are also available in the EMERALD format (Ritter et al., 2015), the reference format for storing time-series in the repository. Please browse the repository for a detailed description of instruments and sensors used.

Participants in the fieldword:

Paula Rulff
Roxana Barth

Other project participants:

Reinhard Klose
Sophie Stephan

We acknowledge contributions from the following funding agencies:

German Research Center for Geociences - GFZ

Further web resources:

MT component of the GIPP

The *.html* formatted output includes the links to additional meta data files, such as *sites.xml*, *maps.xml*, *pictures.xml* and, if available, to *publications.xml*

The reference latitude and longitude values of the project become *clickable* as they can be used to show the overall location of the project with the interactive map generated in the maps folder.

Information on participating scientists and involved cooperation partners are optional but very desirable information. As before, any links to web resources or email addresses will be resolved and opened in additional web-browser windows (or tabs).

5.2 sites.xml, site folder(s), and time-series data

project.xml makes a reference to *sites.xml* and one *sites.xml* file exists in each project folder.

The file *sites.xml* contains information on all sites of a project, their coordinates and available data sets.

Versions:

- 1.00: Original definition.
- 2.00: Changed the hardwired data types to dynamically adopted tags, depending on the equipment used for the project. Introduction of data levels (0=primary time series data from instruments, 1=processed time series data).
- 2.10: No changes in the *xml* file but style sheet file *sites2.1.xslt* is used to link in interactive maps.

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type='text/xsl' href='../xslt/sites2.1.xslt'?>
<SITES>
  <VERSION>2.10</VERSION>
  <proj.name>DEMO.2018</proj.name>
  <USER>Gerard Muñoz</USER>
  <site id="001">
    <name>0001</name>
    <start.date>2008-04-20</start.date>
    <end.date>2008-04-22</end.date>
    <wgs84.lat.dec>52.3808</wgs84.lat.dec>
    <wgs84.lon.dec>13.0655</wgs84.lon.dec>
    <altitude.m>91.5000</altitude.m>
    <rot.to.Magnetic.North>0.00</rot.to.Magnetic.North>
    <declination>0.00</declination>
    <data level="0">
      <burst>true</burst>
      <edl>true</edl>
      <lmt>false</lmt>
      <spam3>false</spam3>
      <spam4>false</spam4>
      <em-edl>true</em-edl>
      <em-lmt>false</em-lmt>
      <em-spam3>false</em-spam3>
      <em-spam4>false</em-spam4>
```

```
    </data>
    <data level="1">
      <raw>false</raw>
    </data>
  </site>
  <site id="002">
    <name>0002</name>
    <start.date>2006-04-21</start.date>
    <end.date>2006-04-27</end.date>
    <wgs84.lat.dec>52.3792</wgs84.lat.dec>
    <wgs84.lon.dec>13.0627</wgs84.lon.dec>
    <altitude.m>86.6000</altitude.m>
    <rot.to.Magnetic.North>0.00</rot.to.Magnetic.North>
    <declination>0.00</declination>
    <data level="0">
      <burst>false</burst>
      <edl>false</edl>
      <lmt>true</lmt>
      <spam3>false</spam3>
      <spam4>false</spam4>
      <em-edl>false</em-edl>
      <em-lmt>true</em-lmt>
      <em-spam3>false</em-spam3>
      <em-spam4>false</em-spam4>
    </data>
    <data level="1">
      <raw>true</raw>
    </data>
  </site>
  <site id="003">
    <name>0003</name>
    <start.date>1997-06-07</start.date>
    <end.date>1997-06-08</end.date>
    <wgs84.lat.dec>52.3768</wgs84.lat.dec>
    <wgs84.lon.dec>13.0665</wgs84.lon.dec>
    <altitude.m>78.9000</altitude.m>
    <rot.to.Magnetic.North>0.00</rot.to.Magnetic.North>
    <declination>0.00</declination>
    <data level="0">
      <burst>false</burst>
      <edl>false</edl>
      <lmt>false</lmt>
      <spam3>true</spam3>
      <spam4>false</spam4>
      <em-edl>false</em-edl>
      <em-lmt>false</em-lmt>
      <em-spam3>true</em-spam3>
      <em-spam4>false</em-spam4>
    </data>
    <data level="1">
      <raw>false</raw>
    </data>
```

```
  </site>
  <site id="004">
    <name>0004</name>
    <start.date>2011-12-03</start.date>
    <end.date>2011-12-07</end.date>
    <wgs84.lat.dec>52.3805</wgs84.lat.dec>
    <wgs84.lon.dec>13.0619</wgs84.lon.dec>
    <altitude.m>88.3000</altitude.m>
    <rot.to.Magnetic.North>0.00</rot.to.Magnetic.North>
    <declination>0.00</declination>
    <data level="0">
      <burst>false</burst>
      <edl>false</edl>
      <lmt>false</lmt>
      <spam3>false</spam3>
      <spam4>true</spam4>
      <em-edl>false</em-edl>
      <em-lmt>false</em-lmt>
      <em-spam3>false</em-spam3>
      <em-spam4>true</em-spam4>
    </data>
    <data level="1">
      <raw>false</raw>
    </data>
  </site>
  <site id="005">
    <name>0005</name>
    <start.date>2015-09-17</start.date>
    <end.date>2015-09-19</end.date>
    <wgs84.lat.dec>52.3830</wgs84.lat.dec>
    <wgs84.lon.dec>13.0641</wgs84.lon.dec>
    <altitude.m>92.10</altitude.m>
    <rot.to.Magnetic.North>0.00</rot.to.Magnetic.North>
    <declination>0.00</declination>
    <data level="0">
      <burst>false</burst>
      <edl>false</edl>
      <lmt>false</lmt>
      <spam3>false</spam3>
      <spam4>true</spam4>
      <em-edl>false</em-edl>
      <em-lmt>false</em-lmt>
      <em-spam3>false</em-spam3>
      <em-spam4>false</em-spam4>
    </data>
    <data level="1">
      <raw>false</raw>
    </data>
  </site>
</SITES>
```

sites.xml (the file above) is used in combination with *sites.xslt*, which resides in the *xslt* folder to generate *html* formatted output (make sure to properly specify the path in *sites.xml* using the `<?xml-stylesheet>` tag):

MT sites recorded for DEMO.2018

Overview map of all sites

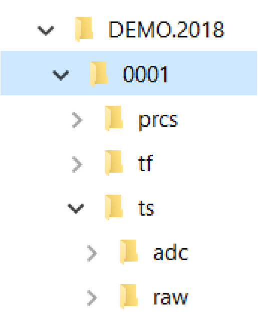
For available data files, please follow the links in the columns of the different data levels.

							level 0								level 1
Site	**Start date**	**End date**	**Latitude**	**Longitude**	**Altitude**	**Declination**	**burst**	**edl**	**lmt**	**spam3**	**spam4**	**em-edl**	**em-lmt**	**em-spam3**	**em-spam4**	**raw**
0001	2008-04-20	2008-04-22	52.3808	13.0655	91.5000	0.00	yes	yes	no	no	no	yes	no	no	no	no
0002	2006-04-21	2006-04-27	52.3792	13.0627	86.6000	0.00	no	no	yes	no	no	no	yes	no	no	yes
0003	1997-06-07	1997-06-08	52.3768	13.0665	78.9000	0.00	no	no	no	yes	no	no	no	yes	no	no
0004	2011-12-03	2011-12-07	52.3805	13.0619	88.3000	0.00	no	no	no	no	yes	no	no	no	yes	no
0005	2015-09-17	2015-09-19	52.3830	13.0641	92.10	0.00	no	no	no	no	yes	no	no	no	no	no

- Clicking on the site name will open up the configuration file for that site (see config.xml)
- Clicking on either latitude or longitude will show the location of this particular site on an interactive map generated in the maps folder. Coordinates are given as decimal numbers and are specified in the WGS84 coordinate frame.
- Altitudes of sites are given in meters above sea-level.
- The last columns of the table indicate available data files. The level-x headings refer to the corresponding data levels. Available data files are listed according to their formats. In the example above *spam4* indicates data recorded with SPAM4 instruments, while *raw* would point to data stored in the EMERALD format, which may have been filtered.

5.2.1 Site subfolders

There is one site folder for each site. Each site folder typically contains a *comment.xml* and a *config.xml* file and the following subfolders:



Folder tree for site 0001

- **prcs**: contains (interim) processing results (optional).
- **tf**: contains transfer function results (optional, level-2 data).
- **ts**: contains time series data. **Mandatory**. This is the most fundamental part of the data archive. Times series data are further subdivided into:

- **adc**: data as generated by the recording devices (level-0 data), typically in a vendor specific data format. It can also contain the recorder specific data converted into EMERALD format without changes.
- **raw**: time series data in EMERALD data format. Apart from reformatting, additional digital filters, re-scaling, or re-sampling may have been applied when compared to the (original) time series (level-1 data).

5.2.2 site/config.xml

There is one *config.xml* file in each site subfolder. The *config.xml* files provides a summary of the recording configuration of the site: sampling frequency for each band, scheduled recording times, filter settings, sensors used, etc.

Internally the *config.xml* files are structured into *runs*. Each *run* corresponds with a particular set of instruments or hardware settings. If, for example, inductions coils were switched between low frequency (LF) and high frequency (HF) modes, their frequency response changes (see response.xml). Therefore, they count as different instruments, which shows in *config.xml* as another run.

For each run the *config.xml* files provide a table with a detailed description of the sampling frequency and the instruments used. The information is given for each chanel and includes IDs and settings of data loggers, sensor boxes, and sensors. The header of the table summarizes the recording periods for which this particular configuration was active.

Versions:

- 1.00: Original definition. Initially only SPAM4 and SB4 sensor boxes were supported and were hardwired into the hardware configuration.
- 2.00: The logger, sensor box, and sensor are stored in a corresponding tag with the device number as an attribute.
- 2.10: Stub version: It is used only in cases when one or more files include several sites (e.g. synchronized time series). Then *config.xml* includes a `<Site>` tag for each site. Within each site, the organization is the same as in the other versions. Backwards compatible with the stylesheet *config2.xslt*.

The contents of an exemplary *config.xml* file could look like this:

```xml
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type='text/xsl' href='../../xslt/config2.xslt'?>
<XTR>
  <VERSION>2.00</VERSION>
  <ScriptDate Version="2.32">15.02.2018</ScriptDate>
  <RunTime>18.10.2018</RunTime>
  <Run id="001">
    <HardwareConfiguration>
      <Project>DEMO.2018</Project>
      <SiteNumber>1</SiteNumber>
      <Channel id="001">
        <Name>Bx</Name>
        <DipoleLength>
        </DipoleLength>
        <HorizontalOrientation>0.000000</HorizontalOrientation>
        <VerticalOrientation>0.000000</VerticalOrientation>
```

```xml
      <StaticGain>0.00000125</StaticGain>
      <Logger number="25">EDL</Logger>
      <SBx number="65">CASTLE</SBx>
      <Sensor number="238">Metronix_Coil-----TYPE-006_LF</Sensor>
    </Channel>
    <Channel id="002">
      <Name>By</Name>
      <DipoleLength>
      </DipoleLength>
      <HorizontalOrientation>90.000000</HorizontalOrientation>
      <VerticalOrientation>0.000000</VerticalOrientation>
      <StaticGain>0.00000125</StaticGain>
      <Logger number="25">EDL</Logger>
      <SBx number="65">CASTLE</SBx>
      <Sensor number="136">Metronix_Coil-----TYPE-006_LF</Sensor>
    </Channel>
    <Channel id="003">
      <Name>Bz</Name>
      <DipoleLength>
      </DipoleLength>
      <HorizontalOrientation>0.000000</HorizontalOrientation>
      <VerticalOrientation>-90.000000</VerticalOrientation>
      <StaticGain>0.00000125</StaticGain>
      <Logger number="25">EDL</Logger>
      <SBx number="65">CASTLE</SBx>
      <Sensor number="24">Metronix_Coil-----TYPE-005_LF</Sensor>
    </Channel>
    <Channel id="004">
      <Name>Ex</Name>
      <DipoleLength>55.600000</DipoleLength>
      <HorizontalOrientation>0.000000</HorizontalOrientation>
      <VerticalOrientation>0.000000</VerticalOrientation>
      <StaticGain>-0.001</StaticGain>
      <Logger number="25">EDL</Logger>
      <SBx number="12">CASTLE</SBx>
      <Sensor number="0">TelluricElectrode-TYPE-AgAgCl</Sensor>
    </Channel>
    <Channel id="005">
      <Name>Ey</Name>
      <DipoleLength>57.500000</DipoleLength>
      <HorizontalOrientation>90.000000</HorizontalOrientation>
      <VerticalOrientation>0.000000</VerticalOrientation>
      <StaticGain>-0.001</StaticGain>
      <Logger number="25">EDL</Logger>
      <SBx number="12">CASTLE</SBx>
      <Sensor number="0">TelluricElectrode-TYPE-AgAgCl</Sensor>
    </Channel>
  </HardwareConfiguration>
  <RecordingMode id="001">
    <FrequencySettings>
      <SamplingFrequency>50.00 Hz</SamplingFrequency>
      <LowPass>0.00 s</LowPass>
```

```
      <HighPass>0.00 s</HighPass>
    </FrequencySettings>
    <RecordingPeriod>
      <PeriodStart>19 Apr 2008 (110) 00:00:00</PeriodStart>
      <PeriodStop>21 Apr 2008 (112) 00:29:59</PeriodStop>
      <For>30min</For>
      <Every>24h</Every>
      <Continuous>False</Continuous>
    </RecordingPeriod>
    <TimeWindow id="0001">
      <StartSec>1208563200</StartSec>
      <StartuSec>0</StartuSec>
      <StopSec>1208564999</StopSec>
      <StopuSec>980000</StopuSec>
      <StartDate>2008-04-19_00-00-00.000000</StartDate>
      <StopDate>2008-04-19_00-29-59.980000</StopDate>
      <Duration>30.00 min</Duration>
      <XTRFile>em-
edl\2008110\0001_SR0000050Hz000_LP0000000s000_HP0000000s000_D2008110_T000000.XTRX</XTRFile>

    </TimeWindow>
    <TimeWindow id="0002">
      <StartSec>1208649600</StartSec>
      <StartuSec>0</StartuSec>
      <StopSec>1208651399</StopSec>
      <StopuSec>980000</StopuSec>
      <StartDate>2008-04-20_00-00-00.000000</StartDate>
      <StopDate>2008-04-20_00-29-59.980000</StopDate>
      <Duration>30.00 min</Duration>
      <XTRFile>em-
edl\2008111\0001_SR0000050Hz000_LP0000000s000_HP0000000s000_D2008111_T000000.XTRX</XTRFile>

    </TimeWindow>
    <TimeWindow id="0003">
      <StartSec>1208736000</StartSec>
      <StartuSec>0</StartuSec>
      <StopSec>1208737799</StopSec>
      <StopuSec>980000</StopuSec>
      <StartDate>2008-04-21_00-00-00.000000</StartDate>
      <StopDate>2008-04-21_00-29-59.980000</StopDate>
      <Duration>30.00 min</Duration>
      <XTRFile>em-
edl\2008112\0001_SR0000050Hz000_LP0000000s000_HP0000000s000_D2008112_T000000.XTRX</XTRFile>

    </TimeWindow>
  </RecordingMode>
  <RecordingMode id="002">
    <FrequencySettings>
      <SamplingFrequency>500.00 Hz</SamplingFrequency>
      <LowPass>0.00 s</LowPass>
      <HighPass>0.00 s</HighPass>
    </FrequencySettings>
    <RecordingPeriod>
```

```xml
        <PeriodStart>19 Apr 2008 (110) 00:00:00</PeriodStart>
        <PeriodStop>21 Apr 2008 (112) 00:11:59</PeriodStop>
        <For>12min</For>
        <Every>24h</Every>
        <Continuous>False</Continuous>
      </RecordingPeriod>
      <TimeWindow id="0001">
        <StartSec>1208563200</StartSec>
        <StartuSec>0</StartuSec>
        <StopSec>1208563919</StopSec>
        <StopuSec>998000</StopuSec>
        <StartDate>2008-04-19_00-00-00.000000</StartDate>
        <StopDate>2008-04-19_00-11-59.998000</StopDate>
        <Duration>12.00 min</Duration>
        <XTRFile>em-
edl\2008110\0001_SR0000500Hz000_LP0000000s000_HP0000000s000_D2008110_T000000.XTRX</XTRFile>

      </TimeWindow>
      <TimeWindow id="0002">
        <StartSec>1208649600</StartSec>
        <StartuSec>0</StartuSec>
        <StopSec>1208650319</StopSec>
        <StopuSec>998000</StopuSec>
        <StartDate>2008-04-20_00-00-00.000000</StartDate>
        <StopDate>2008-04-20_00-11-59.998000</StopDate>
        <Duration>12.00 min</Duration>
        <XTRFile>em-
edl\2008111\0001_SR0000500Hz000_LP0000000s000_HP0000000s000_D2008111_T000000.XTRX</XTRFile>

      </TimeWindow>
      <TimeWindow id="0003">
        <StartSec>1208736000</StartSec>
        <StartuSec>0</StartuSec>
        <StopSec>1208736719</StopSec>
        <StopuSec>998000</StopuSec>
        <StartDate>2008-04-21_00-00-00.000000</StartDate>
        <StopDate>2008-04-21_00-11-59.998000</StopDate>
        <Duration>12.00 min</Duration>
        <XTRFile>em-
edl\2008112\0001_SR0000500Hz000_LP0000000s000_HP0000000s000_D2008112_T000000.XTRX</XTRFile>

      </TimeWindow>
    </RecordingMode>
  </Run>
</XTR>
```

The *.html* formatted version of *config.xml* is obtained using the correponding *.xslt* stylesheet:

Comments, Hardware Configurations and Recording Modes

Project: DEMO.2018

Site number: 1 - Comments

Run: 001

Recording Period 19 Apr 2008 (110) 00:00:00 - 21 Apr 2008 (112) 00:29:59 (For 30min every 24h)						
Logger (EDL)	**SBx (CASTLE)**	**Sampling Frequency**	**Channel Nr.**	**Name**	**Sensor Type**	**Sensor Number**
25	65	50.00 Hz	001	Bx	Metronix_Coil-----TYPE-006_LF	238
			002	By	Metronix_Coil-----TYPE-006_LF	136
			003	Bz	Metronix_Coil-----TYPE-005_LF	24
	12		004	Ex	TelluricElectrode-TYPE-AgAgCl	0
			005	Ey	TelluricElectrode-TYPE-AgAgCl	0

Recording Period 19 Apr 2008 (110) 00:00:00 - 21 Apr 2008 (112) 00:11:59 (For 12min every 24h)						
Logger (EDL)	**SBx (CASTLE)**	**Sampling Frequency**	**Channel Nr.**	**Name**	**Sensor Type**	**Sensor Number**
25	65	500.00 Hz	001	Bx	Metronix_Coil-----TYPE-006_LF	238
			002	By	Metronix_Coil-----TYPE-006_LF	136
			003	Bz	Metronix_Coil-----TYPE-005_LF	24
	12		004	Ex	TelluricElectrode-TYPE-AgAgCl	0
			005	Ey	TelluricElectrode-TYPE-AgAgCl	0

A recording period is described as follows:

- Start Date (Day of Year) Start Time - End Data (Day of Year) End Time (file length), e.g.
    - 23 Sep 2015 (266) 11:00:00 - 24 Sep 2015 (267) 23:12:24 (continuous 1h)
    - 25 Sep 2015 (268) 12:00:00 - 26 Sep 2015 (269) 06:09:59 (For 10min every 1h)
- The meaning of the recording intervals is as follows:
    - *continuous 1h* - a set of data files each recorded continously for one hour. There are no time gaps between the data files, with the exception of the first and last data file which may not last for the full hour.
    - *For 10 min every 1h* - a 10-min long file was recorded every hour.
    - If a recording consists only of one file, the recording period is just the duration of the file and the word "once".

The *config.xml* file contains a link to the *comment.xml* and for each sensor a link to the corresponding *.rspx* file (see response.xml).

5.2.3 site/comment.xml

There is one *comment.xml* file in each site subfolder. The *comment.xml* file provides additional information for this particular site, for example like this:

Versions:

- 1.00: Original definition.

```xml
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type='text/xsl' href='../../xslt/comment.xslt'?>
<COMMENT>
  <VERSION>1.00</VERSION>
  <comment.text>Very high winds.</comment.text>
  <comment.text>Operator suffered from severe brain drain due to strong
 sun.</comment.text>
  <comment.text>No additional information</comment.text>
</COMMENT>
```

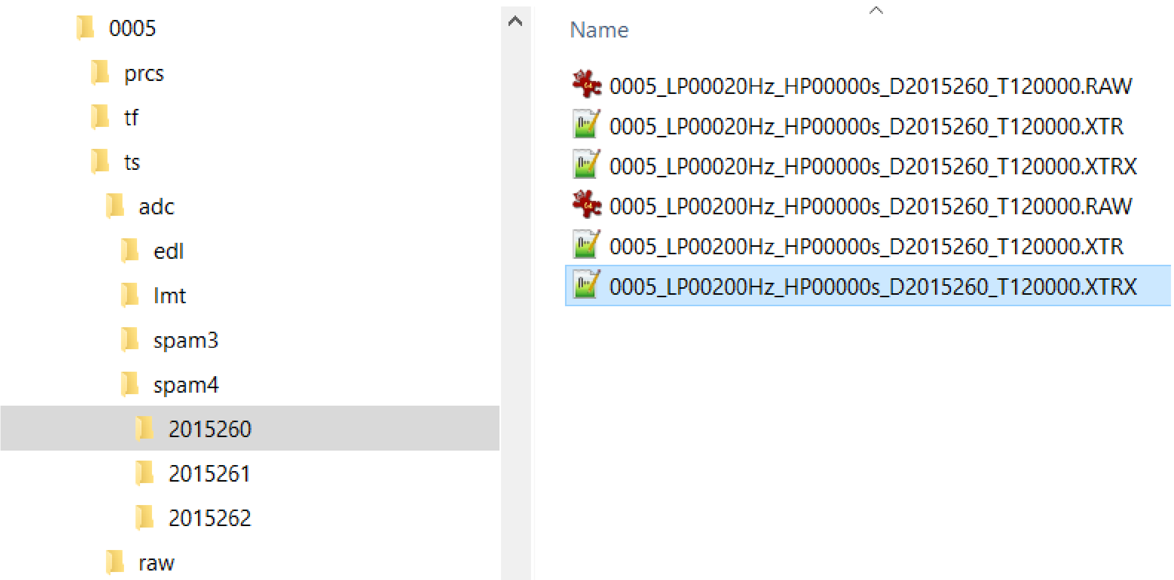
Use the <comment.text> tags to structure your text (one paragraph per tag). You can include all kind of information relevant for (re-)processing the data, such as problems with instruments, a wrong hardware settings, broken cables, wrong orientation of sensors, etc.

5.2.4 site/ts/adc/recorder.xml

Actually, a file named *recorder.xml* file does not exist as *recorder* is synonymous for a piece of equipment (see chapter 6). Examples of existing *recorder.xml* files would be *spam4.xml* or *em-spam3.xml*. The *recorder.xml* files are located in the *.../ts/adc* folder. The exemplary files (*.xml* and *.html*) shown in this section reflect the files associated to data in EMERALD format, for other types of *recorder.xml* files (e.g. *spam3.xml* or *edl.xml*) refer to the recorder types in chapter 6.

The EMERALD format forms the basis for all data exchange and ensures that older data sets can be read (in future) or converted to other data formats. Only the SPAM4 instruments write the data directly in EMERALD format. For other recording equipment, the *ts/adc* folders typically contain an *ts/adc/recorder* subfolder together with an associated *.../ts/adc/em-recorder* subfolder. The former contains the data in the vendor specific data format of a recording device, while the latter holds data which were converted (1:1) into the EMERALD format.

The *.../ts/adc/recorder* folder contains the time-series data, which can be sorted into daily subfolders (for EMERALD and EDL type data) or not (for SPAM3 or older SPAM4 type data):



Subfolders of ts/adc and ts/raw folders

The naming convention for the daily data folders is *YYYYddd*, where *YYYY* is the year of recording, *ddd* specifies

DOI: 10.2312/GFZ.b103-19065

the day of the year when the data were collected (starting with 1 at January 1st and ending with 365/366 at December 31st). The example shows data collected with a SPAM4 system, which are already written in the EMERALD type format for time series (as *.raw* / *.xtrx* files).

Contrary to the config.xml files, the EMERALD type *recorder.xml* files (from version 2.0 of Oct 2018) are organized according to their frequency settings: sampling frequency, low pass and high pass filters. Following the definition of the *EMERALD* format each data file contains only continuous sets of samples. New files can be created if desired, e.g. to organize the data as files of one hour length, but any gaps in the data requires a new file to be written.

Below is an exemplary *spam4.xml* file in *EMERALD* format (*.xml* source and the according *.html* web browser representation via *.xslt* stylesheet). Both files have been edited to contain fewer time windows for the sake of simplicity. For the complete files, check the DEMO.2018 project folders. All EMERALD type *recorder.xml* files (e.g. *em-edl.xml*, *em-spam3.xml*, etc.) have the same structure.

Versions:

- 1.00: Original definition. Files were organized into runs (or days).
- 1.50: Used temporarily to store EMERALD-type SPAM4 data. There is no actual version 1.50 of the EMERALD-type *recorder.xml* files. This version is therefore only included for completeness.
- 1.60: The tag `<FILENAME>` was added to distinguish the files from the content, since the root tag is `<EMERALD>` for all of them. This tag was also added to *spam4.xml* but has been now discontinued.
- 2.00: The run nodes were removed (together with hardware information). Runs are sorted according to the frequency settings. A link to the *config.xml* file was added in the stylesheet to easily obtain the hardware configuration.

```xml
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type='text/xsl' href='../../../../xslt/emerald2.xslt'?>
<EMERALD>
  <VERSION>2.00</VERSION>
  <PROJECT>DEMO.2018</PROJECT>
  <SITE>0005</SITE>
  <USER>Gerard Muñoz</USER>
  <DATE>Thu, 18 Oct 2018 12:59:50 </DATE>
  <FILENAME>spam4.xml</FILENAME>
  <XTR2XML>ArchiveCreateXML.ps1</XTR2XML>
  <frequency.settings>
    <sampl.rate>-500.000000</sampl.rate>
    <lp>-200.000000</lp>
    <hp>0.000000</hp>
    <file.identifier>
      <file.names>
        <name>2015260/0005_LP00200Hz_HP00000s_D2015260_T120000.RAW</name>
        <xtrname>2015260/0005_LP00200Hz_HP00000s_D2015260_T120000.XTRX</xtrname>
      </file.names>
      <year>2015</year>
      <day>260</day>
      <time.window id="001">
        <start>1442491200</start>
        <startms>1580</startms>
        <stop>1442491799</stop>
```

```
        <stopms>999580</stopms>
        <start.date>Thu, 17 Sep 2015 12:00:00 </start.date>
        <stop.date>Thu, 17 Sep 2015 12:09:59 </stop.date>
      </time.window>
    </file.identifier>
  </frequency.settings>
  <frequency.settings>
    <sampl.rate>-50.000000</sampl.rate>
    <lp>-20.000000</lp>
    <hp>0.000000</hp>
    <file.identifier>
      <file.names>
        <name>2015260/0005_LP00020Hz_HP00000s_D2015260_T120000.RAW</name>
        <xtrname>2015260/0005_LP00020Hz_HP00000s_D2015260_T120000.XTRX</xtrname>
      </file.names>
      <year>2015</year>
      <day>260</day>
      <time.window id="001">
        <start>1442491200</start>
        <startms>580</startms>
        <stop>1442494799</stop>
        <stopms>980580</stopms>
        <start.date>Thu, 17 Sep 2015 12:00:00 </start.date>
        <stop.date>Thu, 17 Sep 2015 12:59:59 </stop.date>
      </time.window>
    </file.identifier>
  </frequency.settings>
</EMERALD>
```


Emerald files for site 0005 in DEMO.2018

File created by Gerard Muñoz at Thu, 18 Oct 2018 12:59:50 using ArchiveCreateXML.ps1

For a summary of the recording settings, check the config.xml file.

- **Frequency settings**

Frequency band information (negative in [Hz], positive in [s], 0=OFF):

sampling rate:	low-pass filter:	high-pass filter:
-500.000000	-200.000000	0.000000

Year 2015 Day 260 - RAW and XTR files: 2015260/0005_LP00200Hz_HP00000s_D2015260_T120000.RAW and 2015260/0005_LP00200Hz_HP00000s_D2015260_T120000.XTRX

Time windows:

no	start.date	stop.date	duration [s]
001	Thu, 17 Sep 2015 12:00:00	Thu, 17 Sep 2015 12:09:59	599

- **Frequency settings**

Frequency band information (negative in [Hz], positive in [s], 0=OFF):

sampling rate:	low-pass filter:	high-pass filter:
-50.000000	-20.000000	0.000000

Year 2015 Day 260 - RAW and XTR files: 2015260/0005_LP00020Hz_HP00000s_D2015260_T120000.RAW and 2015260/0005_LP00020Hz_HP00000s_D2015260_T120000.XTRX

Time windows:

no	start.date	stop.date	duration [s]
001	Thu, 17 Sep 2015 12:00:00	Thu, 17 Sep 2015 12:59:59	3599

Note, clicking the link to a particular data or meta data file will download that file.

In section 6 we provide more information on predefined recording devices and the associated *recorder.xml* files. Below is a summary in alphabetical order:

- Earth Data logger (EDL). MINISEED data format.
    - edl: usually continuous data streams of broad-band recordings, in combination with induction coils.
    - lmt: usually continuous data streams with low sampling rates, in combination with fluxgate magnetometers.
    - burst: repeated recordings of short segments of continuous data recorded with high sampling rates, in combination with induction coils.
- SPAM3 recorders. Proprietary data format (see spam3.xml). Superseded by the SPAM4 data loggers (~2010). Usually used for broadband recordings in combination with induction coils. Data are stored as continuous streams of low-pass or band-pass filtered time series.
- SPAM4 recorders. Initially data was written in a proprietary format. After a major firmware upgrade in 2012 EMERALD- type data files are written. Usually used for broadband recordings in combination with induction coils. Data are stored as continuous streams of low-pass (or band-pass) filtered time series. For reference, if both a *spam4.xml* and *em-spam4.xml* file are present, data was recorded in old proprietary format (spam4) and later converted into EMERALD format; if only a *spam4.xml* file is present, data was recorded natively in EMERALD format. Looking into the *.xml* files also reveals more information about the formats. For instance, the old spam4 format has the tag `<SPAM4>` associated, while the spam4 data in EMERALD format has the tag `<EMERALD>` associated.
- SM25 recorders. Proprietary data format (see sm25.xml). The SM25 loggers are used to collect data at high frequencies (10 kHz - 1000 kHz), so-called Radio Magnetotelluric (RMT) applications. Data are stored as segments of low-pass filtered time series.

Note, all of the above data formats can be converted into the most recent version of the EMERALD format. The corresponding recorder.xml file uses then the prefix *em-*, e.g. *em-spam3*.

5.2.5 raw.xml - processed time series

The *raw.xml* file summarizes all the level 1 data (found in any of the data subfolders of .../ts/raw). It is subdivided according to the data folders (called `<RAWDIRS>`) and within it, ordered by sampling frequency. Since the files in the raw folder are in *EMERALD* format, each file corresponds to a single time window

Versions:

- 1.00: Original definition. The xml files were organized into raw folders (`<RAWDIRS>`) and into runs (or days).
- 2.00: The run structure (and hardware information) was removed. The files are now sorted according to

the frequency settings.

This is how the *raw.xml* file and its associated browser view could look like:


```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type='text/xsl' href='../../../../xslt/raw2.xslt'?>
<RAW>
  <VERSION>2.00</VERSION>
  <PROJECT>DEMO.2018</PROJECT>
  <SITE>0002</SITE>
  <USER>Gerard Muñoz</USER>
  <DATE>Thu, 18 Oct 2018 13:20:51</DATE>
  <XTR2XML>ArchiveCreateXML.ps1</XTR2XML>
  <RAWDIRS id="001">
    <raw.dir>.\lmt</raw.dir>
    <file.identifier id="001">
      <name>0002_SR0000002Hz000_LP0000002s000_HP0000000s000_D2006111_T000006.RAW</name>

  <xtrname>0002_SR0000002Hz000_LP0000002s000_HP0000000s000_D2006111_T000006.XTRX</xtrname>
      <sampl.rate>-2.000000</sampl.rate>
      <lp>2.000000</lp>
      <hp>0.000000</hp>
      <time.window id="001">
        <start>1145577606</start>
        <startms>0</startms>
        <stop>1146182393</stop>
        <stopms>500000</stopms>
        <start.date>Fri, 21 Apr 2006 00:00:06 </start.date>
        <stop.date>Thu, 27 Apr 2006 23:59:53 </stop.date>
      </time.window>
    </file.identifier>
    <file.identifier id="002">
      <name>0002_SR0000004s000_LP0000016s000_HP0000000s000_D2006111_T000054.RAW</name>

  <xtrname>0002_SR0000004s000_LP0000016s000_HP0000000s000_D2006111_T000054.XTRX</xtrname>
      <sampl.rate>4.000000</sampl.rate>
      <lp>16.000000</lp>
      <hp>0.000000</hp>
      <time.window id="001">
        <start>1145577654</start>
        <startms>0</startms>
        <stop>1146182342</stop>
        <stopms>0</stopms>
        <start.date>Fri, 21 Apr 2006 00:00:54 </start.date>
        <stop.date>Thu, 27 Apr 2006 23:59:02 </stop.date>
      </time.window>
    </file.identifier>
    <file.identifier id="003">
      <name>0002_SR0000032s000_LP0000128s000_HP0000000s000_D2006111_T000718.RAW</name>

  <xtrname>0002_SR0000032s000_LP0000128s000_HP0000000s000_D2006111_T000718.XTRX</xtrname>
      <sampl.rate>32.000000</sampl.rate>
      <lp>128.000000</lp>
```

```
      &lt;hp&gt;0.000000&lt;/hp&gt;
      <time.window id="001">
        &lt;start&gt;1145578038&lt;/start&gt;
        &lt;startms&gt;0&lt;/startms&gt;
        &lt;stop&gt;1146181942&lt;/stop&gt;
        &lt;stopms&gt;0&lt;/stopms&gt;
        <start.date>Fri, 21 Apr 2006 00:07:18 </start.date>
        <stop.date>Thu, 27 Apr 2006 23:52:22 </stop.date>
      </time.window>
    </file.identifier>
  &lt;/RAWDIRS&gt;
&lt;/RAW&gt;
```

EMERALD type RAW time series data for site 0002 in project DEMO.2018

File created by Gerard Muñoz at Thu, 18 Oct 2018 13:20:51 using ArchiveCreateXML.ps1

For a summary of the recording settings, check the config.xml file.

Listing data directory: .\lmt

Listing of RAW / XTR files for this run:

001.) .\lmt/0002_SR0000002Hz000_LP0000002s000_HP0000000s000_D2006111_T000006.RAW and .\lmt/0002_SR0000002Hz000_LP0000002s000_HP0000000s000_D2006111_T000006.XTRX

Frequency band information (negative in [Hz], positive in [s], 0=OFF):

start.date	stop.date	duration [s]	sampling rate:	low-pass filter:	high-pass filter:
Fri, 21 Apr 2006 00:00:06	Thu, 27 Apr 2006 23:59:53	604787	-2.000000	2.000000	0.000000

002.) .\lmt/0002_SR0000004s000_LP0000016s000_HP0000000s000_D2006111_T000054.RAW and .\lmt/0002_SR0000004s000_LP0000016s000_HP0000000s000_D2006111_T000054.XTRX

Frequency band information (negative in [Hz], positive in [s], 0=OFF):

start.date	stop.date	duration [s]	sampling rate:	low-pass filter:	high-pass filter:
Fri, 21 Apr 2006 00:00:54	Thu, 27 Apr 2006 23:59:02	604688	4.000000	16.000000	0.000000

003.) .\lmt/0002_SR0000032s000_LP0000128s000_HP0000000s000_D2006111_T000718.RAW and .\lmt/0002_SR0000032s000_LP0000128s000_HP0000000s000_D2006111_T000718.XTRX

Frequency band information (negative in [Hz], positive in [s], 0=OFF):

start.date	stop.date	duration [s]	sampling rate:	low-pass filter:	high-pass filter:
Fri, 21 Apr 2006 00:07:18	Thu, 27 Apr 2006 23:52:22	603904	32.000000	128.000000	0.000000

The file names of the *RAW / XTR(X)* files in the ts/raw/*recorder* subfolders can vary quite significantly since there have been a number of naming conventions for the EMERALD files, however the *raw.xml* file structure remains the same.

The folder structure of the ts/raw/*recorder* folder can also vary. While the vast majority of projects have all the level-1 data directly in the corresponding *recorder* subfolder of the ts/raw folder, for some projects this *recorder* folder can be further subdivided into daily folders (either following the YYYYddd - Year in 4 digits plus day of the year in 3 digits - naming convention of the EMERALD type data folders or a now obsolete ddd naming convention -only the day of the year in 3 digits-). In any case, the complete path of the data files is shown in the *raw.xml* file (and browser representation or *.html* file).

5.3 transmitters.xml and transmitter folder(s)

Current transmitters are used for controlled source EM projects. The *transmitters.xml* file provides an overview of the locations and settings of all transmitters used in a CSEM project. The file can be created manually but there are also Powershell scripts available to generate or modify the *transmitters.xml* file. If applicable, the script *ArchiveCreateXMLs.ps1* fills the *transmitters.xml* file with information found in *.xml* files which were created by the Metronix CSEM transmitter, so called ADU header files. The corresponding time series data files are stored in the *.../ts/adc/aduxtr* folder (see below). Additional information such as contact resistances, positions and depths of steel rods, or wire layout can be edited using *ArchiveEditTransmitterXML.ps1*.

Versions:

- 1.00: Original definition.
- 2.00: The data inclusion tags were reorganized into data levels, similar to the *sites.xml* file.

A typical *transmitters.xml* file could look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type='text/xsl' href='../xslt/transmitters.xslt'?>
<TRANSMITTERS>
  <VERSION>1.00</VERSION>
  <proj.name>PROJECT1.2015</proj.name>
  <USER>User Name</USER>
  <transmitter id="001">
    <name>Transmitter ID (4 digit number)</name>
    <start.date></start.date>
    <end.date></end.date>
    <wgs84.lat.dec>Position of Data Logger (usually close to grounding
 electrode)</wgs84.lat.dec>
    <wgs84.lon.dec>0.00</wgs84.lon.dec>
    <altitude.m>0.00</altitude.m>
    <rot.to.Magnetic.North>0.00</rot.to.Magnetic.North>
    <declination>0.00</declination>
    <adc>
      <adutrx>true</adutrx>
      <spam4trx>true</spam4trx>
      <spam4cc>true</spam4cc>
      <raw>false</raw>
    </adc>
    <TransmitterSetup>
      <SteelRods>
        <I1>
          <wgs84.lat.dec> Position of steel rod at end of wire 1</wgs84.lat.dec>
          <wgs84.lon.dec> Position of steel rod at end of wire 1</wgs84.lon.dec>
          <wgs84.altitude.m> Position of steel rod at end of wire 1</wgs84.altitude.m>
          <depth.m> Depth of steel rod 1</depth.m>
          <contactResistance>Contact resistance of steel rod 1 in Ohm </contactResistance>
        </I1>
        <I2>
          <wgs84.lat.dec> Position of steel rod at end of wire 2</wgs84.lat.dec>
```


DOI: 10.2312/GFZ.b103-19065

```
            <wgs84.lon.dec> Position of steel rod at end of wire 2</wgs84.lon.dec>
            <wgs84.altitude.m> Position of steel rod at end of wire 2</wgs84.altitude.m>
            <depth.m> Depth of steel rod 2</depth.m>
            <contactResistance>Contact resistance of steel rod 2 in Ohm </contactResistance>
          </I2>
          <I3>
            <wgs84.lat.dec> Position of steel rod at end of wire 3</wgs84.lat.dec>
            <wgs84.lon.dec> Position of steel rod at end of wire 3</wgs84.lon.dec>
            <wgs84.altitude.m> Position of steel rod at end of wire 3</wgs84.altitude.m>
            <depth.m> Depth of steel rod 3</depth.m>
            <contactResistance>Contact resistance of steel rod 3 in Ohm </contactResistance>
          </I3>
          <ground>
            <wgs84.lat.dec>Position of grounding steel rod </wgs84.lat.dec>
            <wgs84.lon.dec> Position of grounding steel rod</wgs84.lon.dec>
            <wgs84.altitude.m> Position of grounding steel rod</wgs84.altitude.m>
            <depth.m>Depth of grounding steel rod </depth.m>
            <contactResistance>Contact resistance of grounding steel rod
 </contactResistance>
          </ground>
        </SteelRods>
        <Wires>
           <groundToI1>
            <trackpoint id="001">
               <wgs84.lat.dec> </wgs84.lat.dec>
               <wgs84.lon.dec> </wgs84.lon.dec>
               <wgs84.altitude.m> </wgs84.altitude.m>
             </trackpoint>
            <trackpoint id="002">
               <wgs84.lat.dec> </wgs84.lat.dec>
               <wgs84.lon.dec> </wgs84.lon.dec>
               <wgs84.altitude.m> </wgs84.altitude.m>
             </trackpoint>
          </groundToI1>
           <groundToI2>
            <trackpoint id="001">
               <wgs84.lat.dec> </wgs84.lat.dec>
               <wgs84.lon.dec> </wgs84.lon.dec>
               <wgs84.altitude.m> </wgs84.altitude.m>
             </trackpoint>
          </groundToI2>
           <groundToI3>
            <trackpoint id="001">
               <wgs84.lat.dec> </wgs84.lat.dec>
               <wgs84.lon.dec> </wgs84.lon.dec>
               <wgs84.altitude.m> </wgs84.altitude.m>
             </trackpoint>
          </groundToI3>
        </Wires>
      </TransmitterSetup>
    </transmitter>
</TRANSMITTERS>
```

- All `<wgs84.lat/long/altitude>` values are given as decimal numbers.
- The `<wires>` tag is used to specify the field layout of the wires used to inject electrical currents into the ground. Usually the cables cannot be laid out as perfectly straight lines but follow roads or have to avoid infrastructure. Dipole or tri-pole configuration can be described with the `<grounToIx>` tags. `<trackpoint>` tags and are used to define the actual geometry of each wire. These tracking points are usually obtained by walking along the cables and sampling position with a handheld GPS device.

There is one *transmitter* folder for each transmitter and each transmitter folder typically contains subfolders for the time series data, similar to the site folder(s) structure. Depending on the recording system, the following naming convention applies for folders residing in *.../ts/adc* and *.../ts/raw*:

- *adutrx*: data recorded with the Metronix ADU recorder installed on the transmitter trailer.
- *spam4trx*: data recorded with the SPAM4 system installed on the transmitter trailer.
- *spam4cc*: data recorded with current clamps and an external SPAM4 system.

5.3.1 transmitter/jobs.xml

There is one *jobs.xml* file in each transmitter subfolder. The *jobs.xml* file provides an overview of all current transmissions carried out for a particular transmitter installation.

Versions:

- 1.00: Original definition.

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<?xml-stylesheet type='text/xsl' href='../xslt/jobs.xslt'?>
<JOBS>
  <PROJECT>PROJECT1.2015</PROJECT>
  <TRANSMITTER>TransmitterID</TRANSMITTER>
  <USER>User Name</USER>
  <VERSION>1.0</VERSION>
  <job id="000">
    <aduFile>aduHeaderFile.xml</aduFile>
    <start.date unit="UTC">2015-01-01 01:10:43</start.date>
    <end.date unit="UTC">2015-01-01 02:10:43</end.date>
    <JobName>Job Name from ADU</JobName>
    <Waveform>Wave form, e.g. DefaultRect</Waveform>
    <MaxCurrentAmplitude unit="A">max. allowed current amplitude</MaxCurrentAmplitude>
    <RotationFrequency unit="Hz">0</RotationFrequency>
    <Cycle id="001"> <!-- 1st Cycle within transmission-->
      <BaseFrequency unit="Hz">Base Frequency in Hz</BaseFrequency>
      <Polarisation unit="degree">Polarisation of tripole (0, 120, -120)</Polarisation>
      <Iterations>How often the signal was repeated.</Iterations>
    </Cycle>
    <Cycle id="002"> <!-- 2nd Cycle within transmission-->
      <BaseFrequency unit="Hz">Base Frequency in Hz</BaseFrequency>
      <Polarisation unit="degree">Polarisation of tripole (0, 120, -120)</Polarisation>
      <Iterations>How often the signal was repeated.</Iterations>
    </Cycle>
  </job>
</JOBS>
```

- <PROJECT> and <TRANSMITTER> tags must match the definitions given in *project.xml* and *transmitters.xml*.
- <USER> Name of user who was present at or responsible for the transmitter during field operation.
- <adufile> ADU header file (see above), which is expected to reside in subfolder *.../ts/adc/adutrx*
- <BaseFrequency> and <Polarisation> are decimal numbers, <Iterations> is an integer.
- <Waveform> Waveform of the transmitted signal. Defined values are: *DefaultRect*

5.3.2 transmitter/comment.xml

There is one *comment.xml* file in each transmitter subfolder. The *comment.xml* file provides additional information for this particular transmitter installation. It is completely analogous to the *comment.xml* file for the receivers and MT sites for MT projects.

5.4 pictures.xml and pictures folder

The *pictures.xml* file provides a listing of available photos or screenshots which are relevant for the project. File formats of images (bitmaps) should be supported by web browsers, e.g. *.png*, *.jpg*, *.gif*, etc.

Versions:

- 1.00: Original definition.

```xml
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type='text/xsl' href='../xslt/pictures.xslt'?>
<PICTURES>
  <VERSION>1.00</VERSION>
  <proj.name>DEMO.2018</proj.name>
  <USER>Gerard Muñoz</USER>
  <pic id="1">
    <info>group foto</info>
    <file>pic.jpg</file>
  </pic>
  <pic id="2">
    <info>landscape</info>
    <file>img.jpg</file>
  </pic>
  <pic id="3">
    <info>site xxx</info>
    <file>photo.jpg</file>
  </pic>
  <pic id="4">
    <info>screenshot of processing site xxx</info>
    <file>processing_site_xxx.jpg</file>
  </pic>
</PICTURES>
```

- Use the `<info>` tags to supply a descriptive figure caption.
- Make sure the `<proj.name>` tag matches the definition in *project.xml*.

When viewing above *pictures.xml* in your web browser, it should look like this:

Listing of available pictures for project: test.2011

Please follow the links ...
group foto
picture
site xxx
screenshot of processing site xxx

Pictures folder

The image files reside in the *pictures* subfolder of the project folder. Relevant photos could include site locations, packing containers, group photos, data charts, etc. All picture files must be copied directly to the *pictures* subfolder, since other subfolders are not read by *ArchiveCreateXMLs.ps1*. Make sure to ask the photographer and people shown on pictures for permission before including them.

5.5 maps.xml and maps folder

maps.xml provides a listing of relevant maps for a project. *ArchiveCreateXMLs.ps1* creates automatically two maps from the coordinates listed in *sites.xml*: one static map called *stationMap.png* to be used as a figure in the report. An interactive map called *stationMap.html* can be accessed via the links in *sites.xml* file. It is possible to zoomed in and out and also to follow links to the *config.xml* files. Users can still add additional maps (e.g. geological maps, GMT topo maps) to be included in the report or just for reference. Use image (bitmap) file formats supported by web browsers, e.g. *.JPG/.JPEG*, *.TIFF*, *.GIF*, *.BMP*, *.PNG*. Make sure that image resolution is acceptable. For maps intended to be included in the report provide a figure number in the figure attribute of the `<map>` tag and a relevant figure caption in the `<caption>` tag.

Versions:

- 1.00: Original definition.
- 1.10: Added the `<caption>` tag for specifying a figure caption. This figure caption will be used in *report.xml*.
- 1.20: Added the @figure attribute. If set to false a particular figure will be omitted from the report, while a number is used as the figure number in *report.xml* .

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type='text/xsl' href='../xslt/maps.xslt'?>
<MAPS>
  <VERSION>1.20</VERSION>
  <proj.name>DEMO.2018</proj.name>
  <USER>Gerard Muñoz</USER>
  <map id="1" figure="false">
    <info>Interactive station map</info>
    <file>station map.html</file>
    <caption>Interactive map with all MT station locations.</caption>
  </map>
  <map id="2" figure="1">
    <info>Station map</info>
    <file>station map.png</file>
    <caption>Location of the MT stations (blue dots).</caption>
  </map>
</MAPS>
```

Maps folder

The map (image) files reside in the *maps* subfolder of the project folder. Relevant images could include profiles, geological overview, tectonic features, etc. Use the `<info>` tag to provide a descriptive name for the map (per default they will get names like Map 1, Map 2, etc.) and the `<caption>` tag to write a figure caption that can be used in the report.

5.6 response.xml and response folder

Many sensors used for magnetotelluric mesurements have a response in amplitude and phase which depends on frequency and which must be considered when processing the data. A typical example are induction coil magnetometers. The *response.xml* file provides a listing of all required response files for the project.

Versions:

- 1.00: Original definition.

An example is shown below:

```xml
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type='text/xsl' href='../xslt/response.xslt'?>
<RESPONSE>
  <VERSION>1.00</VERSION>
  <proj.name>DEMO.2018</proj.name>
  <USER>Paula Rulff</USER>
  <response id="1">
    <file type="RSPX">Metronix_Coil-----TYPE-005_LF-ID-000022</file>
  </response>
  <response id="2">
    <file type="RSPX">Metronix_Coil-----TYPE-005_LF-ID-000023</file>
  </response>
  <response id="3">
    <file type="RSPX">Metronix_Coil-----TYPE-005_LF-ID-000024</file>
  </response>
</RESPONSE>
```

Response folder

The files with the frequency responses of the sensors reside in the *response* subfolder of the *project* folder. The contents of these files is in *XML* format and the files have the extension *.rspx*. Older versions of these files may also exist with the extensions *.rsp*; they are organized as simple ASCII tables.

Note that the *response.xml* file will only include all *rsp(x)* files that are present in the response folder when *ArchiveCreateXMLs.ps1* is called. In order to obtain the necessary response files you must contact the GIPP team and provide a list of the necessary sensors and time frame. The list of all instruments used in the project can be obatined automatically through the *instrumentation.xml* file (see below).

5.7 instrumentation.xml

instrumentation.xml provides a listing of all used instruments in the project: Types and IDs of data loggers, coils, electrodes and sensor boxes. The information is compiled from all of the *config.xml* files from all *sites* subfolders.

Versions:

- 1.00: Original definition.

A typical *instrumentation.xml* file looks like this:

```xml
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type='text/xsl' href='../xslt/instrumentation.xslt'?>
<INSTRUMENTATION>
  <VERSION>1.00</VERSION>
  <proj.name>DEMO.2018</proj.name>
  <USER>Gerard Muñoz</USER>
  <Instrument type="Logger">
    <Loggers id="1" name="SPAM4">
      <Logger id="1">35</Logger>
      <Logger id="2">62</Logger>
    </Loggers>
    <Loggers id="2" name="EDL">
      <Logger id="1">1</Logger>
      <Logger id="2">25</Logger>
    </Loggers>
    <Loggers id="3" name="SPAM III">
      <Logger id="1">30</Logger>
    </Loggers>
  </Instrument>
  <Instrument type="SensorBox">
    <SensorBoxes id="1" name="SP4">
      <SensorBox id="1">3</SensorBox>
      <SensorBox id="2">111</SensorBox>
      <SensorBox id="3">127</SensorBox>
    </SensorBoxes>
    <SensorBoxes id="2" name="CASTLE">
      <SensorBox id="1">4</SensorBox>
      <SensorBox id="2">12</SensorBox>
      <SensorBox id="3">65</SensorBox>
    </SensorBoxes>
  </Instrument>
  <Instrument type="Sensor">
    <Sensors id="1" name="AgAgCl electrode" />
    <Sensors id="2" name="Metronix Coil MFS05">
      <Sensor id="1">22</Sensor>
      <Sensor id="2">23</Sensor>
      <Sensor id="3">24</Sensor>
      <Sensor id="4">27</Sensor>
    </Sensors>
    <Sensors id="3" name="Metronix Coil MFS06">
      <Sensor id="1">126</Sensor>
      <Sensor id="2">134</Sensor>
      <Sensor id="3">136</Sensor>
      <Sensor id="4">238</Sensor>
      <Sensor id="5">400</Sensor>
      <Sensor id="6">439</Sensor>
    </Sensors>
    <Sensors id="4" name="Metronix Coil MFS07">
      <Sensor id="1">123</Sensor>
    </Sensors>
    <Sensors id="5" name="Pulz fluxgate">
      <Sensor id="1">11</Sensor>
```

```
    </Sensors>
  </Instrument>
</INSTRUMENTATION>
```

In the web browser, the instrumentation file appears as:

Summary of all instruments used in project: DEMO.2018

The following **SPAM4** loggers were used:
35
62

The following **EDL** loggers were used:
1
25

The following **SPAM III** loggers were used:
30

The following **SP4** sensor boxes were used:
3
111
127

The following **CASTLE** sensor boxes were used:
4
12
65

AgAgCl electrodes were used

The following **Metronix Coil MFS05** sensors were used:
22
23
24
27

The following **Metronix Coil MFS06** sensors were used:
126
134
136
238
400
439

The following **Metronix Coil MFS07** sensors were used:
123

The following **Pulz fluxgate** sensors were used:
11

5.8 publications.xml

The file *publications.xml* contains relevant publications related to the project. The format of *publications.xml* follows the definitions for the *BibTeX*- format (see e.g. https://en.wikipedia.org/wiki/BibTeX). Bibliography items can be of the usual types, e.g. *article*, *book*, *phdthesis*, etc, and include fields such as *author*, *journal*, *volume*, etc. But instead of the *TeX*/*LaTeX* style of commands, e.g. `\bibitem`, the organisation of the bibliography files is in *XML* format, sometimes referred to as *BibXML*.

While it is possible to enter a complete publication list (*bibitems*) per hand, it is usually more convinient to add publications using the PowerShell script *ArchiveAddPublication.ps1*, if the publication has a doi, or *ArchiveQueryPublication.ps1*, if the publication can be found in the database of the GFZ library.

Some of the `<bibitem>` objects may still contain additional tags labelled `<project>`. This feature is now obsolete as usage and organization of *publications.xml* was changed during the development. Initially only one (huge) *publications.xml* file existed for the entire archive, which contained all `<bibitem>` entries for all projects. This file (together with the `<project>` tags) was extremely difficult to maintain and is now obsolete. Instead one *publications.xml* file exists for each project.

Versions:

- 1.00: Original definition.

A typical entry in *publications.xml* could look like this:

```xml
<bibitem type = "article">
  <author> <lastname>Becken</lastname> <firstname> Michael</firstname></author>
  <author> <lastname>Ritter</lastname> <firstname>Oliver</firstname></author>
  <author> <lastname>Bedrosian</lastname> <firstname>Paul A.</firstname></author>
  <author> <lastname>Weckmann</lastname> <firstname>Ute</firstname></author>
  <journal>Nature</journal>
  <volume>480</volume>
  <issue>7375</issue>
  <title>Correlation between deep fluids, tremor and creep along the central San Andreas
 fault</title>
  <pages>87-90</pages>
  <year>2011</year>
  <url>http://www.nature.com/nature/journal/v480/n7375/full/nature10609.html</url>
</bibitem>
```

When viewing *publications.xml* in your web browser (remember that a corresponding stylesheet is needed) it might look like this:

PUBLICATIONS related to project ELSAF.2008

Journal articles

Becken, M., Ritter, O., Bedrosian, P., Weckmann, U., 2011. Correlation between deep fluids, tremor and creep along the central San Andreas fault, *Nature*, 480, 87-90. Get paper.

Conference papers

Tietze, K., Ritter, O., Egbert, G., 2014. 3D joint inversion of the magnetotelluric phase tensor and vertical magnetic transfer functions with ModEM and its application to a 250-site MT array data set from the San Andreas fault, California, in: *22nd EM Induction Workshop, Weimar, Germany, August 24-30, 2014*, Weimar (Germany).

Theses

Tietze, K., 2012. Investigating the electrical conductivity structure of the San Andreas fault in the Parkfield-Cholame region with 3D magnetotelluric inversion, *Free University Berlin*, 161 pp. Get paper.

5.9 revision.xml and revision history

Each project contains a *revision.xml* file which describes the revision history of the report. The tag `<last.modified>` indicates when any one of the *xml* files relevant for the report was last modified (even if it does not create a new revision).

Each revision is then listed in the `<revision>` nodes. Revisions numbers 0.X indicate draft (pre-publication) versions, revision 1.0 is reserved for the first published version of the report. Higher numbers indicate new releases of the publication, e.g. to accomodate new information or corrections. If *revision.xml* contains version numbers higher than 1.0, a section with the revision history will appear in the report.

Versions

- 1.00: Original definition

This is how the *revision.xml* file looks like:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type='text/xsl' href='../xslt/revision.xslt'?>
<REVISIONS>
    <VERSION>1.00</VERSION>
    <proj.name>DEMO.2018</proj.name>
    <USER>Gerard Muñoz</USER>
    <last.modified>20.11.2018</last.modified>
    <revision id="001">
        <revision.number>0.00</revision.number>
        <user>Gerard Muñoz</user>
        <date>20.11.2018</date>
        <text>Work in progress</text>
        <text>Create tag with version 1.0 and text "First publication version" when the
 project is ready for publication</text>
    </revision>
</REVISIONS>
```

Revision history for project: DEMO.2018

- **0.00** - Gerard Muñoz - 20.11.2018:
  Work in progress
  Create tag with version 1.0 and text "First publication version" when the project is ready for publicationmm

5.10 report.xml and reportSnippets.xml

Reports accompanying data publications typically follow a well defined organisation. Sections could include, for example, abstract or summary, introduction, experimental setup and schedule, station and transmitter locations, used instrumentation, and available data sets. When considering the wealth of information available in the MT repository data, such a report can be assembled in a semi-automatic way from the existing meta data.

In this respect report.xml does not add much new information to the repository. Instead it is a compilation of all the information provided in the *.xml* files above, which forms the basis for the data publications.

Typically report.xml is created using the Powershell script *ArchiveCreateReport*. In addition to the information from the repository, the report requires headers sections, text blocks decribing default information, etc. This kind of information is stored in the reportSnippets.xml file which resides in the *util* folder as it contains useful information for **all** projects. Please modify this file with care.

When converted into html format using *ArchiveCreateHTMLs* the *report.html* file can be read into a word processing program (e.g. Microsoft Word) for additional editing or further converted into pdf format. This (edited) report constitutes the basis of the STR (Scientific Technical Report) which accompanies the data publications.

Versions:

- 1.00: The file *report.xml* is nothing but a compilation of all other *xml* files related to the project (and it is actually deleted after conversion into HTML), therefore each report section has the version number of the *xml* file forming it.

---

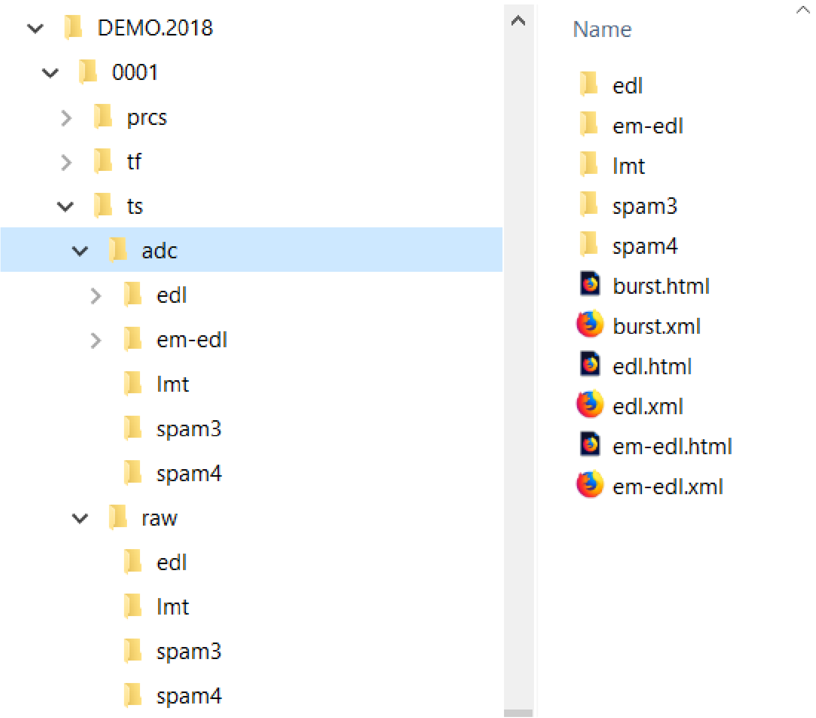
6 Predefined equipment and their *recorder.xml* files

In the following sections we provide a description of some of the hardware mentioned above. Particularly, we describe the recording equipment and their respective *recorder.xml* files.

As already mentioned, common basis for data exchange are time-series data in the EMERALD format. Therefore, for each site one or more data subfolders exist in the *.../ts/adc* tree with associated recorder.xml files. The current EMERALD format (with *.xtrx* files) is directly written by the SPAM4 systems since 2012.

Older versions of the SPAM4 system and any other types of equipment record the time-series data in a vendor specific format (or an outdated version of the EMERALD format). But keeping these data in the repository is desirable as they represent the original recordings. They can always be referred to if in doubt of the converted data. Typically, these data are binary files. To enhance their readability, meta-data are supplied as associated *.xml* files.

To accomodate the data of these other recording devices in the repository the *.../ts/adc* folders can contain subfolders which are named accordingly. Subfolders can be empty if respective equipment was not used at that site, or they can be completely absent. In the figure below the *.../ts/adc* folder contains the subfolders *edl*, *lmt*, and *spam3*, which hold the time series data. Respective *xml* files exist for each one of the recording device subfolders. Since there is no *spam3.xml* file, the *spam3* data folder is empty. *edl.xml* + *burst.xml* and *lmt.xml* describe data recorded with an Earth Data logger (EDL). The *em-edl* folder contains data EDL data converted one-to-one into EMERALD format.



The adc (and raw) subfolders contain .xml files describing the data files for various recording devices (edl, lmt, spam3).

Known formats of such (sometimes outdated) recorder files are summarized below:

6.1 EDL data logger

The Earth Data Logger (EDL, EARTH DATA, U.K.) is a digital recording system predominantly designed for seismic applications. To provide the necessary interface electronics for MT sensors, the EDL loggers are typically used in combination with the CASTLE preconditioning units of the Geophysical Instrument Pool Potsdam. These Sensor-Boxes provide necessary high-impedance amplifiers for electric field recordings and a range of analogue high- and low-pass filters to match typical MT applications.

EDL loggers are available as 3 and 6 channel systems, the latter is normally used for MT applications. EDL systems are GPS synchronized and based on 24-bit analogue to digital converters featuring sampling rates between 1 Hz and 1 kHz.

EDL time series data are stored in the MiniSEED format (https://www.gfz-potsdam.de/en/section/geophysical-deep-sounding/infrastructure/geophysical-instrument-pool-potsdam-gipp/documents/data-format/).

6.1.1 edl.xml, burst.xml, lmt.xml

edl.xml, *burst.xml*, *lmt.xml* are associated *.xml* files for data recorded in MINISEED format. The three recorder files all have the same format as their names only idicate differing recording modes. Broad band MT recordings are often split into continuous sampling modes and a so-called burst modes, in which only discontinuous time segments are recorded, typically at higher sampling rates. In case a *burst.xml* file exists, the *edl.xml* file will only contain references to *.sec* files and the *burst.xml* only to *.pri* files. In case there is no *burst.xml*, the *edl.xml* file contains references to *.pri* files. The *lmt.xml* file, which describes long period MT data sampled at very low rates, contains references to *.pri* files in any case.

The data subfolders can also be related to data converted from the proprietary format of the recording device into *EMERALD* format. In this case the subfolders follow the naming convention *em-recorder*. For example, the subfolder *em-edl* contains EDL data converted into *EMERALD* format. In this case the *ts/adc* folder would also contain an *em-edl.xml* file.

Converted data files in *em-recorder* subfolders contain the exact same information as the original files; they are therefore considered level-0 data.

Versions (apply to all edl-type *xml* files):

- 1.00: Original definition.

The example below shows the (simplified) contents of an *edl.xml* file:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type='text/xsl' href='../../../../xslt/edl.xslt'?>
<EDL>
  <VERSION>1.00</VERSION>
  <PROJECT>DEMO.2018</PROJECT>
  <SITE>0001</SITE>
  <USER>Gerard Muñoz</USER>
  <DATE>Thu, 18 Oct 2018 12:59:26 </DATE>
  <XTR2XML>ArchiveCreateXML.ps1</XTR2XML>
  <RUN id="001">
    <run.identifier>R002</run.identifier>
    <channel.identifier id="001">
      <name>Bx</name>
      <line.len>0.0</line.len>
      <horiz.rot>0.00</horiz.rot>
      <vert.rot>0.00</vert.rot>
      <static.gain>-0.0000006250</static.gain>
      <sensor.name>Metronix MFS05/06</sensor.name>
      <sensor.number>238</sensor.number>
    </channel.identifier>
    <channel.identifier id="002">
      <name>By</name>
      <line.len>0.0</line.len>
      <horiz.rot>90.00</horiz.rot>
      <vert.rot>0.00</vert.rot>
      <static.gain>-0.0000006250</static.gain>
      <sensor.name>Metronix MFS05/06</sensor.name>
      <sensor.number>136</sensor.number>
    </channel.identifier>
    <channel.identifier id="003">
      <name>Bz</name>
      <line.len>0.0</line.len>
      <horiz.rot>0.00</horiz.rot>
      <vert.rot>-90.00</vert.rot>
```

```
      <static.gain>-0.0000006250</static.gain>
      <sensor.name>Metronix MFS05/06</sensor.name>
      <sensor.number>24</sensor.number>
    </channel.identifier>
    <channel.identifier id="004">
      <name>Ex</name>
      <line.len>55.6000</line.len>
      <horiz.rot>0.00</horiz.rot>
      <vert.rot>0.00</vert.rot>
      <static.gain>0.0000025000</static.gain>
      <sensor.name>Ag/AgCl Electrode</sensor.name>
      <sensor.number>0</sensor.number>
    </channel.identifier>
    <channel.identifier id="005">
      <name>Ey</name>
      <line.len>57.5000</line.len>
      <horiz.rot>90.00</horiz.rot>
      <vert.rot>0.00</vert.rot>
      <static.gain>0.0000025000</static.gain>
      <sensor.name>Ag/AgCl Electrode</sensor.name>
      <sensor.number>0</sensor.number>
    </channel.identifier>
    <file.identifier>
      <file.names>
        <edl.file>
          <name>/110/e6025080419000000.sec0</name>
          <name>/110/e6025080419000000.sec1</name>
          <name>/110/e6025080419000000.sec2</name>
          <name>/110/e6025080419000000.sec3</name>
          <name>/110/e6025080419000000.sec4</name>
        </edl.file>
      </file.names>
      <sampl.rate>-50.00</sampl.rate>
      <lp>-12.50</lp>
      <hp>0.00</hp>
      <time.window id="001">
        <start>1208559600</start>
        <startms>0</startms>
        <stop>1208561399</stop>
        <stopms>980000</stopms>
        <start.date>Fri, 18 Apr 2008 23:00:00 </start.date>
        <stop.date>Fri, 18 Apr 2008 23:29:59 </stop.date>
      </time.window>
    </file.identifier>
  </RUN>
</EDL>
```

As before, the *edl.xml* file is accompanied by an *xslt* style sheet (see: `<?xml-stylesheet type="text/xsl" href="../xslt/edl.xslt" ?>`). In your web browser the contents of *edl.xml* may therefore look like this:

CONTINUOUS mode EDL files for site 0001 in DEMO.2018

File created by Gerard Muñoz at Thu, 18 Oct 2018 12:59:26 using ArchiveCreateXML.ps1

Run ID: R002

Channel information:

name	line.len	horiz.rot	vert.rot	static.gain	sensor.name	sensor.no
Bx	0.0	0.00	0.00	-0.0000006250	Metronix MFS05/06	238
By	0.0	90.00	0.00	-0.0000006250	Metronix MFS05/06	136
Bz	0.0	0.00	-90.00	-0.0000006250	Metronix MFS05/06	24
Ex	55.6000	0.00	0.00	0.0000025000	Ag/AgCl Electrode	0
Ey	57.5000	90.00	0.00	0.0000025000	Ag/AgCl Electrode	0

Frequency band information (negative in [Hz], positive in [s], 0=OFF):

sampling rate:	low-pass filter:	high-pass filter:
-50.00	-12.50	0.00

Time windows:

no	start.date	stop.date	duration [s]
001	Fri, 18 Apr 2008 23:00:00	Fri, 18 Apr 2008 23:29:59	1799

EDL (BB) file(s):

channel 0	channel 1	channel 2	channel 3	channel 4
/110/e6025080419000000.sec0	/110/e6025080419000000.sec1	/110/e6025080419000000.sec2	/110/e6025080419000000.sec3	/110/e6025080419000000.sec4

The exemplary *edl.xml* file description contains a lot of information useful for subsequent reprocessing of the data, including but not limited to sensors used, lengths of electric dipoles, start- and stop times of the recordings.

The *edl.xml* files are organized as *runs*; each *run* corresponds to a particular hardware setting. Within a *run*, there can be one or more continuous time windows with defined frequency settings (sampling rate, low pass and high pass filters). The time-series data are stored as MINISEED formatted files (file extensions: **.pri* or **.sec*).

6.2 SPAM4 data logger

S.P.A.M.-systems are **s**hort- **p**eriod **a**utomatic **m**agnetotelluric instruments developed since 1980 by Graham Dawes at the University of Edinburgh. In 2003, Oliver Ritter initiated development of the fourth generation of the instruments in cooperation between GFZ Potsdam and the University of Edinburgh, which was released in 2009.

Generally, a S.P.A.M. Mk IV or SPAM4 systems consist of three principal components:

- Sensors such as induction coils and electrodes.
- Sensor-box for analogue signal preconditioning.
- System-box which contains the analogue-to-digital converters, GPS synchronisation and a miniature personal computer for data storage and processing.

The overall frequency range of the system is from ~10 kHz (max. 25 kHz sampling rate) to DC. The SPAM4 analogue section provides software-controlled amplification with variable gains. There are a range of low- and high- pass filters in the analogue signal path which can be combined to match the sampling rates and sensors used. The last stage of the analogue signal path is the analogue-to-digital conversion. SPAM4 uses a 24-Bit sigma-delta converter. All subsequent data processing is digital.

The time-series data can be continuously filtered and decimated. Low-pass and high-pass corner frequency settings are variable. Any low-pass filtered time series can subsequently be high-pass filtered, to generate band-pass filtered data streams. All data streams can be stored as low-pass filtered time series. Many combinations are possible and all time series are continuous streams of data.

6.2.1 spam4.xml

Since 2012 the SPAM4 loggers write data in the EMERALD format. While the format of the *.raw* files has not changed since release of the SPAM4, there were several changes with respect to file name convention and the associated *.xtr / .xtrx* files. Therefore, the appearance of the *ts/adc/spam4* folders can vary quite significantly between projects. For a summary of the file names and their description, check the documentation of the EMERALD format.

In addition, the folder structure has also experienced some changes. Prior to 2012, when the SPAM4 system wrote data in proprietary format the *ts/adc/spam4* did not have any subfolder and all the data files from all recording days and all frequency bands were located directly in the *spam4* folder. Nowadays data files are organized in daily subfolders.

Versions:

- 1.00: Initial and still active definition.

As a (now discontinued) branch, the following versions were temporarily used:

- (1.50) Temporary solution to store (with the associated *spam4.1.5.xslt* stylesheet) metadata of SPAM4 data in EMERALD format. Since the SPAM4 loggers write data directly in the EMERALD format they no loger use this *xml* file (but the *recorder.xml* files in the EMERALD format).
- (1.60) The `<FILENAME>` tag was added for analogy with the EMERALD-type *em-recorder.xml* files. This version number is also discontinued.

An exemplary (simplified) *spam4.xml* file (and associated browser representation):

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type='text/xsl' href='../../../../xslt/spam4.xslt'?>
<SPAM4>
  <VERSION>1.00</VERSION>
  <PROJECT>DEMO.2018</PROJECT>
  <SITE>0004</SITE>
  <USER>Gerard Muñoz</USER>
  <DATE>Thu, 18 Oct 2018 12:59:48</DATE>
  <XTR2XML>ArchiveCreateXML.ps1</XTR2XML>
  <RUN id="001">
    <run.identifier>002</run.identifier>
```

```xml
<channel.identifier id="001">
  <name>Bx</name>
  <line.len>0.0</line.len>
  <horiz.rot>0.000000</horiz.rot>
  <vert.rot>0.000000</vert.rot>
  <static.gain>0.00000125</static.gain>
  <sensor.name>Metronix_Coil-----TYPE-006_LF-ID-000134.RSPX</sensor.name>
  <sensor.number>134</sensor.number>
</channel.identifier>
<channel.identifier id="002">
  <name>By</name>
  <line.len>0.0</line.len>
  <horiz.rot>90.000000</horiz.rot>
  <vert.rot>0.000000</vert.rot>
  <static.gain>0.00000125</static.gain>
  <sensor.name>Metronix_Coil-----TYPE-006_LF-ID-000400.RSPX</sensor.name>
  <sensor.number>400</sensor.number>
</channel.identifier>
<channel.identifier id="003">
  <name>Bz</name>
  <line.len>0.0</line.len>
  <horiz.rot>0.000000</horiz.rot>
  <vert.rot>90.000000</vert.rot>
  <static.gain>0.0000015625</static.gain>
  <sensor.name>Metronix_Coil-----TYPE-007_LF-ID-000123.RSPX</sensor.name>
  <sensor.number>123</sensor.number>
</channel.identifier>
<channel.identifier id="004">
  <name>Ex</name>
  <line.len>64.900000</line.len>
  <horiz.rot>0.000000</horiz.rot>
  <vert.rot>0.000000</vert.rot>
  <static.gain>-0.001</static.gain>
  <sensor.name>TelluricElectrode-TYPE-AgAgCl-ID-000000.RSPX</sensor.name>
  <sensor.number>0</sensor.number>
</channel.identifier>
<channel.identifier id="005">
  <name>Ey</name>
  <line.len>65.200000</line.len>
  <horiz.rot>90.000000</horiz.rot>
  <vert.rot>0.000000</vert.rot>
  <static.gain>-0.001</static.gain>
  <sensor.name>TelluricElectrode-TYPE-AgAgCl-ID-000000.RSPX</sensor.name>
  <sensor.number>0</sensor.number>
</channel.identifier>
<file.identifier>
  <file.names>
    <name>L004F7B0.002_031211_120000</name>
  </file.names>
  <sampl.rate>-500.000000</sampl.rate>
  <lp>-200.000000</lp>
  <hp>0.000000</hp>
```


DOI: 10.2312/GFZ.b103-19065

```
      <time.window id="001">
        <start>1322913600</start>
        <startms>0</startms>
        <stop>1322914199</stop>
        <stopms>998000</stopms>
        <start.date>Sat, 03 Dec 2011 12:00:00 </start.date>
        <stop.date>Sat, 03 Dec 2011 12:09:59 </stop.date>
      </time.window>
    </file.identifier>
      <file.names>
        <name>L004F7B1.002_061211_120000</name>
      </file.names>
      <sampl.rate>-50.000000</sampl.rate>
      <lp>-20.000000</lp>
      <hp>0.000000</hp>
      <time.window id="001">
        <start>1323172800</start>
        <startms>0</startms>
        <stop>1323176399</stop>
        <stopms>980000</stopms>
        <start.date>Tue, 06 Dec 2011 12:00:00 </start.date>
        <stop.date>Tue, 06 Dec 2011 12:59:59 </stop.date>
      </time.window>
    </file.identifier>
  </RUN>
</SPAM4>
```


SPAM4 files for site 0004 in DEMO.2018

File created by Gerard Muñoz at Thu, 18 Oct 2018 12:59:48 using ArchiveCreateXML.ps1

Run ID: 002

Channel information:

name	line.len	horiz.rot	vert.rot	static.gain	sensor.name	sensor.no
Bx	0.0	0.000000	0.000000	0.00000125	Metronix_Coil-----TYPE-006_LF-ID-000134.RSPX	134
By	0.0	90.000000	0.000000	0.00000125	Metronix_Coil-----TYPE-006_LF-ID-000400.RSPX	400
Bz	0.0	0.000000	90.000000	0.0000015625	Metronix_Coil-----TYPE-007_LF-ID-000123.RSPX	123
Ex	64.900000	0.000000	0.000000	-0.001	TelluricElectrode-TYPE-AgAgCl-ID-000000.RSPX	0
Ey	65.200000	90.000000	0.000000	-0.001	TelluricElectrode-TYPE-AgAgCl-ID-000000.RSPX	0

SPAM4 file(s): L004F7B0.002_031211_120000

Frequency band information (negative in [Hz], positive in [s], 0=OFF):

sampling rate:	low-pass filter:	high-pass filter:
-500.000000	-200.000000	0.000000

Time windows:

no	start.date	stop.date	duration [s]
001	Sat, 03 Dec 2011 12:00:00	Sat, 03 Dec 2011 12:09:59	599

SPAM4 file(s): L004F7B1.002_061211_120000

Frequency band information (negative in [Hz], positive in [s], 0=OFF):

sampling rate:	low-pass filter:	high-pass filter:
-50.000000	-20.000000	0.000000

Time windows:

no	start.date	stop.date	duration [s]
001	Tue, 06 Dec 2011 12:00:00	Tue, 06 Dec 2011 12:59:59	3599

Similar to the *config.xml* files, the *spam4.xml* (and the *spam3.xml* files) are organized in runs. A run consists of a particular hardware configuration. Switching the coils, for example, between high frequency -HF- and low frequency -LF- modes, counts as different hardware setups, since the calibration needed for the coils differs. Each run consists of one or more files and for each file one (in both the SPAM3 and SPAM4 systems) or more (in the SPAM3 system) time windows can be defined.

With the firmware update in 2012, the SPAM4 loggers not only write native EMERALD format data, but also subdivide the *ts/adc/spam4* folder into daily sub-folders, which contain the *raw / xtr(x)* files. The naming convention for the daily data folders is YYYYddd, where YYYY is the year of recording, ddd specifies the day of the year when the data were collected (starting with 1 at January 1st and ending with 365/366 at December 31st). For an example of an EMERALD type *recorder.xml* file (which applies to the EMERALD format *spam4.xml* see above).

6.3 SPAM3 data logger

S.P.A.M. Mk. III systems (or SPAM3 for short) were developed by Graham Dawes between 1990 and 1995 at the Univertsity of Edinburgh, UK. They were superseeded by the SPAM4 systems in 2009. SPAM3 operated as a networked instrument. This network consisted of sensors with distributed analogue and digital processing and control units. Networks could be very simple, e.g. a standard 5-component MT configuration, or very complex systems, limited only by the availability of SPAM3-modules. The network communication was based on digital data transfer via cables, and all devices on the network were synchronised. The computational heart of SPAM3 was the *transputer*, a microprocessor with built in parallel processing. Parallel execution of processes allowed frequency bands to be generated digitally and concurrently, at that time an important improvement over existing instruments.

SPAM3 supported a range of sensors: electrodes, magnetometers and seismometers. Sensors were attached via CASTLE sensor-boxes. They provided general signal preconditioning, power supply for active sensors and high impedance amplifiers for electric field recordings. The highest sampling rate of the SPAM3 systems was 8 KHz using 16-bit analogue to digital converters.

6.3.1 spam3.xml

Versions:

- 1.00: Original definition.

An exemplary (simplified) spam3.xml file is shown below:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type='text/xsl' href='../../../../xslt/spam3.xslt'?>
```

```xml
<SPAM3>
  <VERSION>1.00</VERSION>
  <PROJECT>DEMO.2018</PROJECT>
  <SITE>0003</SITE>
  <USER>Gerard Muñoz</USER>
  <DATE>Thu, 18 Oct 2018 12:59:44</DATE>
  <XTR2XML>ArchiveCreateXML.ps1</XTR2XML>
  <RUN id="001">
    <run.identifier>H01</run.identifier>
    <channel.identifier id="001">
      <name>Bx</name>
      <line.len>0.0</line.len>
      <horiz.rot>0.00</horiz.rot>
      <vert.rot>0.00</vert.rot>
      <static.gain>0.0000012500</static.gain>
      <sensor.name>Metronix_Coil-----TYPE-005_BB-ID-000024.RSP</sensor.name>
      <sensor.number>24</sensor.number>
    </channel.identifier>
    <channel.identifier id="002">
      <name>By</name>
      <line.len>0.0</line.len>
      <horiz.rot>90.00</horiz.rot>
      <vert.rot>0.00</vert.rot>
      <static.gain>0.0000012500</static.gain>
      <sensor.name>Metronix_Coil-----TYPE-005_BB-ID-000023.RSP</sensor.name>
      <sensor.number>23</sensor.number>
    </channel.identifier>
    <channel.identifier id="003">
      <name>Bz</name>
      <line.len>0.0</line.len>
      <horiz.rot>0.00</horiz.rot>
      <vert.rot>-90.00</vert.rot>
      <static.gain>0.0000012500</static.gain>
      <sensor.name>Metronix_Coil-----TYPE-005_BB-ID-000027.RSP</sensor.name>
      <sensor.number>27</sensor.number>
    </channel.identifier>
    <channel.identifier id="004">
      <name>Ex</name>
      <line.len>57.6000</line.len>
      <horiz.rot>0.00</horiz.rot>
      <vert.rot>0.00</vert.rot>
      <static.gain>-0.0000500000</static.gain>
      <sensor.name>TelluricElectrode-TYPE-AgAgCl-ID-000000.RSP</sensor.name>
      <sensor.number>0</sensor.number>
    </channel.identifier>
    <channel.identifier id="005">
      <name>Ey</name>
      <line.len>55.9000</line.len>
      <horiz.rot>90.00</horiz.rot>
      <vert.rot>0.00</vert.rot>
      <static.gain>-0.0000500000</static.gain>
      <sensor.name>TelluricElectrode-TYPE-AgAgCl-ID-000000.RSP</sensor.name>
```


DOI: 10.2312/GFZ.b103-19065

```xml
      <sensor.number>0</sensor.number>
    </channel.identifier>
    <file.identifier>
      <file.names>
        <name>003BPC01.H01</name>
      </file.names>
      <sampl.rate>-1024.00</sampl.rate>
      <lp>-256.00</lp>
      <hp>-32.00</hp>
      <time.window id="001">
        <start>865742504</start>
        <startms>90820</startms>
        <stop>865742508</stop>
        <stopms>964844</stopms>
        <start.date>Sun, 08 Jun 1997 04:01:44 </start.date>
        <stop.date>Sun, 08 Jun 1997 04:01:48 </stop.date>
      </time.window>
      <time.window id="002">
        <start>865742544</start>
        <startms>62500</startms>
        <stop>865742551</stop>
        <stopms>936524</stopms>
        <start.date>Sun, 08 Jun 1997 04:02:24 </start.date>
        <stop.date>Sun, 08 Jun 1997 04:02:31 </stop.date>
      </time.window>
      <time.window id="003">
        <start>865742590</start>
        <startms>62500</startms>
        <stop>865742597</stop>
        <stopms>936524</stopms>
        <start.date>Sun, 08 Jun 1997 04:03:10 </start.date>
        <stop.date>Sun, 08 Jun 1997 04:03:17 </stop.date>
      </time.window>
      <time.window id="004">
        <start>865742636</start>
        <startms>62500</startms>
        <stop>865742643</stop>
        <stopms>936524</stopms>
        <start.date>Sun, 08 Jun 1997 04:03:56 </start.date>
        <stop.date>Sun, 08 Jun 1997 04:04:03 </stop.date>
      </time.window>
    </file.identifier>
  </RUN>
  <RUN id="002">
    <run.identifier>S01</run.identifier>
    <channel.identifier id="001">
      <name>Bx</name>
      <line.len>0.0</line.len>
      <horiz.rot>0.00</horiz.rot>
      <vert.rot>0.00</vert.rot>
      <static.gain>0.0000012500</static.gain>
      <sensor.name>Metronix_Coil-----TYPE-005_BB-ID-000024.RSP</sensor.name>
```


DOI: 10.2312/GFZ.b103-19065

```xml
      <sensor.number>24</sensor.number>
    </channel.identifier>
    <channel.identifier id="002">
      <name>By</name>
      <line.len>0.0</line.len>
      <horiz.rot>90.00</horiz.rot>
      <vert.rot>0.00</vert.rot>
      <static.gain>0.0000012500</static.gain>
      <sensor.name>Metronix_Coil-----TYPE-005_BB-ID-000023.RSP</sensor.name>
      <sensor.number>23</sensor.number>
    </channel.identifier>
    <channel.identifier id="003">
      <name>Bz</name>
      <line.len>0.0</line.len>
      <horiz.rot>0.00</horiz.rot>
      <vert.rot>-90.00</vert.rot>
      <static.gain>0.0000012500</static.gain>
      <sensor.name>Metronix_Coil-----TYPE-005_BB-ID-000027.RSP</sensor.name>
      <sensor.number>27</sensor.number>
    </channel.identifier>
    <channel.identifier id="004">
      <name>Ex</name>
      <line.len>57.6000</line.len>
      <horiz.rot>0.00</horiz.rot>
      <vert.rot>0.00</vert.rot>
      <static.gain>-0.0000500000</static.gain>
      <sensor.name>TelluricElectrode-TYPE-AgAgCl-ID-000000.RSP</sensor.name>
      <sensor.number>0</sensor.number>
    </channel.identifier>
    <channel.identifier id="005">
      <name>Ey</name>
      <line.len>55.9000</line.len>
      <horiz.rot>90.00</horiz.rot>
      <vert.rot>0.00</vert.rot>
      <static.gain>-0.0000500000</static.gain>
      <sensor.name>TelluricElectrode-TYPE-AgAgCl-ID-000000.RSP</sensor.name>
      <sensor.number>0</sensor.number>
    </channel.identifier>
    <file.identifier>
      <file.names>
        <name>003BPC01.S01</name>
      </file.names>
      <sampl.rate>-64.00</sampl.rate>
      <lp>-16.00</lp>
      <hp>-4.00</hp>
      <time.window id="001">
        <start>865659470</start>
        <startms>78125</startms>
        <stop>865741600</stop>
        <stopms>62500</stopms>
        <start.date>Sat, 07 Jun 1997 04:57:50 </start.date>
        <stop.date>Sun, 08 Jun 1997 03:46:40 </stop.date>
```

```
      </time.window>
    </file.identifier>
  </RUN>
</SPAM3>
```

The *html* formatted version looks like this:

SPAM3 files for site 0003 in DEMO.2018

File created by Gerard Muñoz at Thu, 18 Oct 2018 12:59:44 using ArchiveCreateXML.ps1

Run ID: H01

Channel information:

name	line.len	horiz.rot	vert.rot	static.gain	sensor.name	sensor.no
Bx	0.0	0.00	0.00	0.0000012500	Metronix_Coil-----TYPE-005_BB-ID-000024.RSP	24
By	0.0	90.00	0.00	0.0000012500	Metronix_Coil-----TYPE-005_BB-ID-000023.RSP	23
Bz	0.0	0.00	-90.00	0.0000012500	Metronix_Coil-----TYPE-005_BB-ID-000027.RSP	27
Ex	57.6000	0.00	0.00	-0.0000500000	TelluricElectrode-TYPE-AgAgCl-ID-000000.RSP	0
Ey	55.9000	90.00	0.00	-0.0000500000	TelluricElectrode-TYPE-AgAgCl-ID-000000.RSP	0

SPAM3 file(s): 003BPC01.H01

Frequency band information (negative in [Hz], positive in [s], 0=OFF):

sampling rate:	low-pass filter:	high-pass filter:
-1024.00	-256.00	-32.00

Time windows:

no	start.date	stop.date	duration [s]
001	Sun, 08 Jun 1997 04:01:44	Sun, 08 Jun 1997 04:01:48	4
002	Sun, 08 Jun 1997 04:02:24	Sun, 08 Jun 1997 04:02:31	7
003	Sun, 08 Jun 1997 04:03:10	Sun, 08 Jun 1997 04:03:17	7
004	Sun, 08 Jun 1997 04:03:56	Sun, 08 Jun 1997 04:04:03	7

Run ID: S01

Channel information:

name	line.len	horiz.rot	vert.rot	static.gain	sensor.name	sensor.no
Bx	0.0	0.00	0.00	0.0000012500	Metronix_Coil-----TYPE-005_BB-ID-000024.RSP	24
By	0.0	90.00	0.00	0.0000012500	Metronix_Coil-----TYPE-005_BB-ID-000023.RSP	23
Bz	0.0	0.00	-90.00	0.0000012500	Metronix_Coil-----TYPE-005_BB-ID-000027.RSP	27
Ex	57.6000	0.00	0.00	-0.0000500000	TelluricElectrode-TYPE-AgAgCl-ID-000000.RSP	0
Ey	55.9000	90.00	0.00	-0.0000500000	TelluricElectrode-TYPE-AgAgCl-ID-000000.RSP	0

SPAM3 file(s): 003BPC01.S01

Frequency band information (negative in [Hz], positive in [s], 0=OFF):

sampling rate:	low-pass filter:	high-pass filter:
-64.00	-16.00	-4.00

Time windows:

no	start.date	stop.date	duration [s]
001	Sat, 07 Jun 1997 04:57:50	Sun, 08 Jun 1997 03:46:40	82130

Note, the file structure is basically the same as the *spam4.xml* file shown before.

6.4 SM25-RMT instrument

The M-K5-SM25 is a 5-channel radiomagnetotelluric data logger build by St. Petersburg State University and the company Microkor, also based in St. Petersburg. This system allows acquisition of three orthogonal magnetic field components and of two orthogonal horizontal electric field components. The system includes compact induction coils covering a frequency range between 1 kHz and 1000 kHz. The amplifers for the electric field sensors support grounded dipoles (5-10 m) or capacitive-coupled electric field lines (10-20 m). Data are recorded in three frequency bands: D1 with a frequency range of 1-10 kHz and a sampling rate of 39 kHz, D2 with a frequency range of 10-100 kHz and a sampling rate of 312 kHz, and D4 with a frequency range of 100-1000 kHz and a sampling rate of 2496 kHz. Band D3 with a frequency range of 10-300 kHz and a sampling rate of 832 kHz is not working properly and therefore not used.

6.4.1 sm25.xml

The *sm25.xml* files are organized in a similar way as the *spam-type* xml files. A *run* is charaterized by a particular hardware configuration, typically a certain frequency band with according sampling rate and filter settings. The time-series data of a *run* is stored in a binary file in a proprietary format. A *run* can contain one or several time windows.

6.5 Current transmitter (Metronix TXM-22)

The Metronix TXM-22 transmitter is used in connection with the TXB-07 transmitter controller to generate electromagnetic source signals suitable for controlled source applications such as Controlled Source Electro Magnetics (CSEM).

It operates in a wide frequency range between 0.001 Hz up to 8,192 Hz. The output current is adjustable up to a maximum of +/-40A. The nominal output voltage is +/-560V. The TXM-22 is powered by a 50/60Hz, 3-phase, 400V motor generator. In order to unleash the full capabilities a generator of min. 40kVA should be used.

The output currents can be fed into 3 electrodes, offering the possibility to rotate the current vector in any direction. The TXB-22 uses a pulse-width modulation (PWM) to generate the output signals in a range of wave forms such as sine wave, square wave, triangle, saw-tooth or PRBS.

The transmitter is synchronized by GPS and can be programmed using a laptop which is connected by LAN or W-LAN.

6.5.1 transmitter.xml

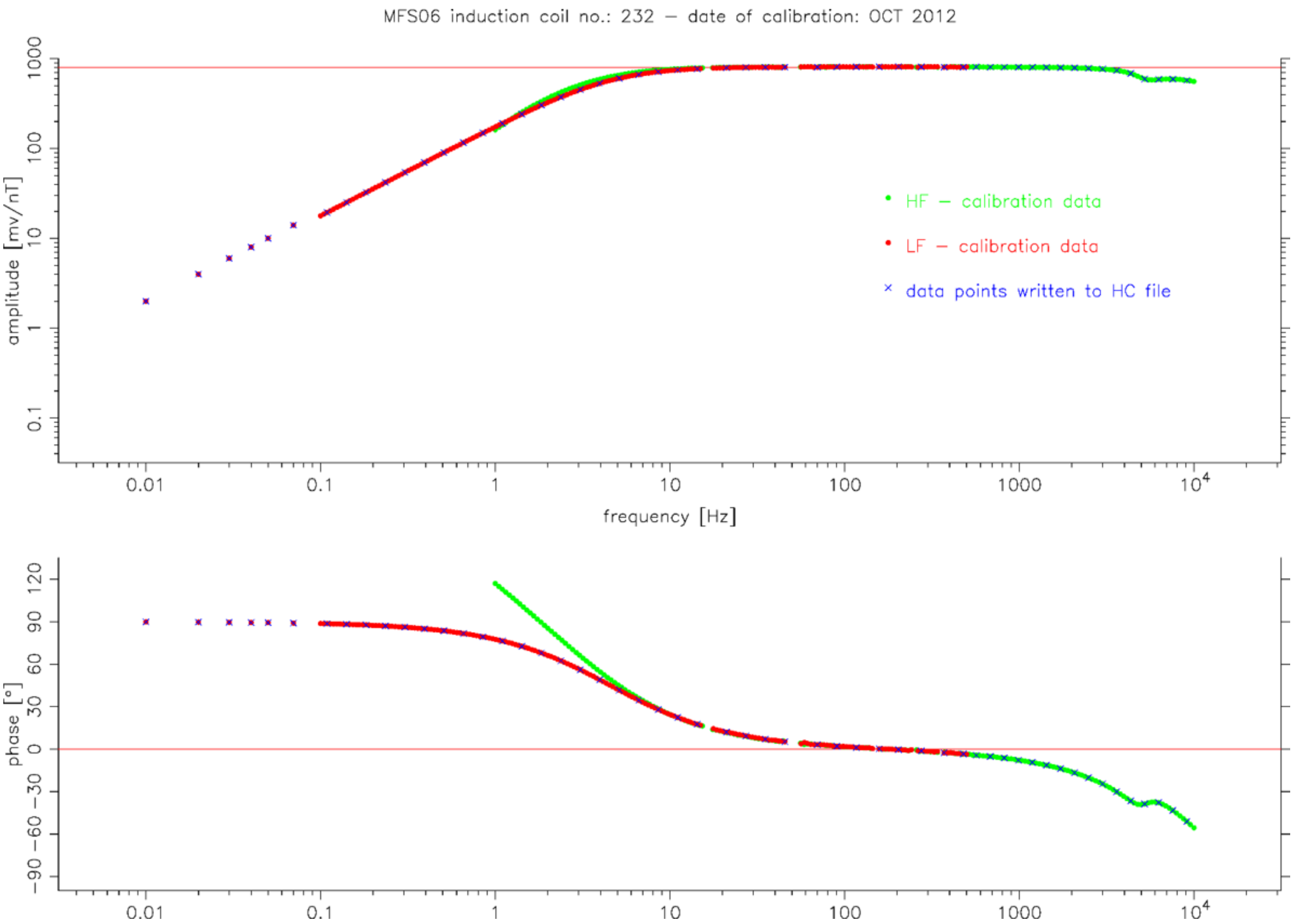
The *transmitter.xml* files are decribed in the corresponding sections above.

6.6 Electric and magnetic sensors

Sensors are devices to measure electric or magnetic fields, typically as continous analogue signals. Sensors are often attached to the data loggers using sensor boxes. As only digitized data is stored, the sensors do not produce data *per se*, and consequently do not have associated *recorder.xml* files. Instead instrument responses are needed to convert any recorded voltages into physically meaningful values. Typically the instrument responses are obtained by calibration measuremnts. The response files of all the sensors used in a project are stored in the responses folder and summarized in the *responses.xml* file (see above).

6.6.1 Induction coils

The induction coil magnetometers MFS-05/06/07/10 (METRONIX, Germany) are designed to measure variations of the Earth´s magnetic field over a wide frequency band (broadband sensor). The sensor coil consists of a highly permeable ferrite core with several thousand copper turns and the magnetometer contains electronics for pre-amplification of the sensor signal. Since induction coil sensors do not measure the magnetic field directly but it's time derivative, their response is highly frequency dependent. The MFSxx sensors cover wide frequency ranges: from approximately 1 mHz to 8 kHz for the MFS06, 1 mHz to 1 kHz for the MFS05, 0.01 Hz to 50 kHz for the MFS07, and 1 mHz to 1 kHz for the MFS10.



Typical amplitude and phase response of the MFS06 induction coil sensor. The nominal amplitude response is approximately 800 mV/nT for frequencies above 4 Hz; for lower frequencies it follows a 1/f behaviour. The sensor can be operated in the so-called high- and low- frequency modes, which hav different frequency responses (red *and green curves).*

7 Other files and folders

7.1 Powershell scripts

PowerShell is a task automation and configuration management framework from Microsoft, consisting of a command-line shell and associated scripting language. Initially a Windows component only, known as Windows PowerShell, it was made open-source and cross-platform in 2016 with the introduction of PowerShell Core.

A number of Powershell scripts have been written to help organize, construct, and maintain the MT repository. Below we summarize briefly their functionality. Please contact GIPP-MT for further information.

7.1.1 ArchiveMakeStructure.ps1

Creates the default folder structure for processing and archiving of MT stations, CSEM receivers and CSEM transmitters. To be called from the project folder at the top of the structure tree.

Example: `ArchiveMakeStructure.ps1 -Site 1`

Creates the folder structure for MT site number 1.

7.1.2 ArchiveCreateXMLs.ps1

The main script to create a MT repository. It generates a number of *.xml* files containing the metadata of the project. Some of the *.xml* files are associated with the entire project, others are associated with each of the sites in the project.

The script only creates the basic structure, the general information as well as the map, picture and response files have to be added by the user.

The script is executed from the project folder (top of the tree) and it will search trough the project structure to identify avalable data types. The script calls other external scripts and programs to generate the corresponding xtr files.

If some xml files to be created exist already, they are skipped (not overwritten).

Created *xml* files include:

- project.xml: general information about the project and links to all other *.xml* files.
- maps.xml: map(s) relevant for the project.
- pictures.xml: pictures relevant for the project.
- sites.xml: a table listing all sites of the project folder with information about recording times, coordinates, and type(s) of available data. Includes links to the configuration.xml files and data type files.
- transmitters.xml: a table listing all transmitters of the project with information about recording times, coordinates, and type(s) of available data. Includes links to jobs.xml files and data type files.
- response.xml: a listing of all calibration files (*.rspx*) found in the associated *response* folder.
- instrumentation.xml: a listing with all serial numbers of all instruments used in the project.
- publications.xml: contains references to publications reelvant for the project.

For each site one or more of the following files are created:

- comment.xml: user comments, information relevant for a particular site.
- recorder.xml: The *.../ts/adc* site-subfolders contain more subfolders which are named after recording devices. One subfolder exists for each data type. If data are converted into *EMERALD* format, they are stored in a subfolder called *em-"data type"* with the corresponding *.xml* files.
- raw.xml: optional. If time-series data is present in the *raw* folder, this file contains a listing of these data (converted into EMERALD format).
- config.xml: see script *ArchiveCreateConfigXML.ps1* for more details.

For each transmitter one or more of the following files are created:

- jobs.xml: information on schedule, form, and duration of transmitted EM signals.
- *adutrx/spam4/spam4trx.xml* : transmitter data, one subfolder for each data type.

Example: `ArchiveCretateXMLs.ps1 -project DEMO.2018 -user "Oliver Ritter" -projectType MT -log`

Creates the xml files for the magnetotellurics project DEMO.2018 and a log file is written. The user is Oliver Ritter.

Example: `ArchiveCreateXMLs.ps1 -project PROJECT1.2015 -user "Kristina Tietze" -projectType CSEM`

Creates all the xml files for the Controled Source Electromagnetic (CSEM) project PROJECT1.2015 (including the transmitter-related files). The user is Kristina Tietze.

For additional details on the use of the ArchiveCreateXMLs.ps1 script in practice, see the Quick guide.

7.1.3 ArchiveUpdateSitesXML.ps1

The script reads file sites.xml from the location specified by the -Path parameter and updates the format to version 2.1.

It can also be used to edit a sites.xml file, e.g. to modify start and stop times, or coordinates. Missing coordinates can be supplied using a *.crd:* file or by reading the information from the *.xtr* files. For older data types (`edl`, `lmt`, `spam3`) the *.xtr* files contain no coordinates, therefore a *.crd* file is needed.

Example: `ArchiveUpdateSitesXML.ps1 -Path .`

Updates the *sites.xml* file in the working folder into version 2.1.

7.1.4 ArchiveUpdateProjectXML.ps1

The script reads the file project.xml from the location specified by the -Path parameter and updates the format to version 2.0.

project.xml should still be edited manually to add any missing information.

7.1.5 ArchiveEditTransmitterXML.ps1

7.1.6 ArchiveCreateConfigXML.ps1

Script creates a config.xml file for a station's time of recording, including information on instruments, sampling frequencies, recording intervals, and hardware modes.

To be called from the project folder (top of tree). Note, the script is internally called for each site by *ArchiveCreateXMLs.ps1*.

7.1.7 ArchiveCleanXMLs.ps1

Running this script can be necessary if something went wrong in the process of creating the archive structure. It deletes all or part of the *.xml* files describing the project in the archive. It can also delete *.html* files or data file (*EMERALD* format).

Example: `ArchiveCleanXMLs.ps1 -project`

Deletes the xml files associated to the project (not the site related).

Example: `ArchiveCleanXMLs.ps1 -data -emData`

Deletes the xml files associated to each site and the generated EMERALD format data as well. The project related xml files (*project.xml*, *sites.xml*, *maps.xml*, etc.) are kept.

The script is executed from the project folder (top of the tree).

7.1.8 ArchiveCreateHTMLs.ps1

The script creates *.html* files for all corresponding *.xml* files, so that all of the project's meta data can be also obtained from *.html* files. The path to *.xslt* files with the *xslt* stylesheets is required for the conversion. The *.html* files are particularly useful if the repository is to be exported as all internal links are maintained.

Note the script requires the command line transformation utility (msxsl.exe) on your computer, e.g. the *util* directory.

The script is executed from the project folder (top of the tree).

Example: `ArchiveCreateHTMLs.ps1 -xsltPath ..\xslt`

Converts all *.xml* files in the current project structure into *.html* by using the stylesheets located in the *xslt* folder at the same level of the project folder.

7.1.9 ArchiveUpdateEncoding.ps1

This scripts changes the encoding of all *xml* files in the folder specified by the parameter *-Path* into UTF-8. This is necessary for the conversion to HTML and to ensure compatibility. If *xml* files were previously created with other encodings, they must be changed. If you are creating a new project structure (and associated *xml* files) from scratch this is not necessary, as all scripts create UTF-8 files.

7.1.10 Other scripts

Some of the MT projects contain *scripts* subfolders with Powershell scripts used to process the recorded dats. This subfolder is optional. The main idea behind this is to ensure that any stored data can be reprocessed, even at much later times.

7.2 Time-series data formats

7.2.1 EMERALD format

The acronym EMERALD was supposed to stand for **E**lectro**M**agnetic **E**quipment, **R**aw data **A**nd **L**ocations **D**atabase. What survived over the years was the EMERALD processing, a set of computer programs to analyse MT time series data, and the EMERALD file format for storing MT data.

EMERALD data files typically come in pairs of two files with the same name but differing file name extensions, sometimes called RAW and XTR files. XTR (extract) files are plain ASCII files, which can be read and modified with text editors. RAW files or more generally, EMERALD -type data files are in most cases binary and used to store all kind of magnetotelluric (MT) data such as time series, cross- and auto spectra and calibration data. The EMERALD -type data files store data in matrix form (any number of channels), but do not contain any description of the data. This information is stored in the according *.xtr* file.

In 2015 the original *.xtr* files were replaced by a modernized version based on the Extensible Markup Language (*XML*). The new files have the extension *.xtrx*.

A brief summary of all the naming conventions is given here:

- Old format (without version number). This old format is a legacy of the old DOS times, when file names could only have 8 characters and extensions 3 characters. In order to pack as much information as possible in only 8 characters a hexadecimal codification and an implicit 2^n decimation scheme was adopted (which is still used by the EMERALD MT processing scheme).

  An example of this naming convention could be: 01H4FA01.XTR. In this case 01 corresponds to the last two digits of the site number (it could be 001 but also 201); H4F corresponds to the filter settings: H means that the frequencies are in Hz, 4 translates into 2^4 = 16, i.e. the low pass filter is set to 16Hz, and F means there was no high pass filter applied; A represents the time window (i.e. this file contains the first continuous recording of the run at the site), subsequent time windows would get B, C, etc.; finally 01 represents the first run (a particular hardware setup).

  This old naming convention is nowadays only found in the ts/raw/ subfolders of some projects before 2010.

- XTR(X) Version 1.0: This format was adopted when Windows allowed larger filenames and to convert SPAM4 data into the EMERALD format. The filenames contained similar information but now without any encoding. Initially the long file names were used with the *.xtr* files but they were later adopted to the *.xtrx* files.

  An example of this naming convention could be: 0001_LP00016Hz_HP00000s_R001_W001.XTR. This file name reflects exactly the same information as the file before: site 001, low pass filter of 16Hz, no high pass filter (conventionally written as 0s), first run and first time window.

- XTRX Version 1.1: This naming convention was adopted to include actual times of measurement instead of counting *Runs* or *Windows*. In this naming convention, the fields run (R001) and window (W001) were substituted by time information in the form of day and time.

  For example, a file following the 1.1 naming convention could be: 0001_LP00016Hz_HP00000s_D2018300_T114800.XTRX. This indicates that the data for site 001 was measured with a low pass filter of 16Hz and no high pass filter. Recording started at 11:48:00 on the 27th of October of 2018 (day 300 of the year).

- XTRX Version 1.2: This naming convention expands the previous version by extending the length of the fields for the low and high pass filters to 7 digit floating points with 3 decimals. Also, the sampling rate appears in the file names in addition to the low and high pass filters.

  An example of this naming convention could be: 0001_SR0000064Hz000_LP0000016Hz000_HP0000000s000_D2018300_T114800.XTRX. It corresponds to the exact same information as in the previous example but now with more significant digits and the sampling rate of 64.00Hz is explicitely shown in the file name.

The EMERALD format for magnetotelluric data is described in detail in Ritter et al. (2015):

7.2.2 MINISEED format

In MINISEED format, the time-series data are usually sorted as individual channels (*.pri / *.sec) into daily subfolders (format *ddd*, with *ddd*=day of the year). Magnetotelluric sensors are connected to EDL recorders via sensor boxes. If CASTLE sensor boxes of the GIPP were used, additional information on the site set-up (sensors, dipole lengths, etc) can be found in the respective EMERALD-type *.xtr/.xtrx* files.

For more information on the MINISEED format see:

https://www.gfz-potsdam.de/en/section/geophysical-deep-sounding/infrastructure/geophysical-instrument-pool-potsdam-gipp/documents/data-format/

7.3 Definition of data levels

- level-0 data: data as generated by the recording device. Data files are typically in a propretriatry binary format.
- level-1 data: data were converted into *EMERALD* format, possibly as a one-to-one copy of the level-0 data. Alternatively additional time-series processing, such as digital filtering, de-trending, or re-scaling may have been applied.
- level-2 data: MT transfer function data after, typically generated by robust, remote-reference processing of the data in the frequency domain.

8 References

Ritter, O., Klose, R., Weckmann, U. (2015): EMERALD Data Format for Magnetotelluric Data, Scientific Technical Report - Data; 15/08, Potsdam: Deutsches GeoForschungsZentrum GFZ. DOI: http://doi.org/10.2312/GFZ.b103-15082

Ritter, O., Munoz, G., Klose, R., Weckmann, U. (2019): Exemplary Data and Code Snippets for the Magnetotelluric Repository of the Geophysical Instrument Pool Potsdam (V. 1.0), GFZ Data Services. http://doi.org/10.5880/GIPP-MT.0001

CC BY

ISSN 2190-7110