

The Toolkit Reference Manual

Table of Contents

The Toolkit Reference Manual	1
1 Overview	2
2 Resource location	3
2.1 Packages.....	3
2.2 Services.....	10
3 Install	11
4 Changelog	12
5 Stone gazer	13
5.1 Server Medusa.....	13
5.2 Labor Sisyphus for Android.....	13
5.3 Labor Sisyphus for MS-Windows.....	15
5.4 Record operator Orochi.....	15
5.5 Document uploader Pmlatex.....	17
5.6 Surface with images.....	18
6 Stone measurer Imoko	19
7 Data format CASTEML	20
7.1 CASTEML Introduction.....	20
7.2 CASTEML commands.....	20
7.3 CASTEML polymorph.....	21
8 Data format IONML	23
8.1 Create IONML.....	23
8.2 Read IONML.....	24
9 Barcode operation	26
9.1 Barcode reader.....	26
9.1.1 Setup code device.....	26
9.1.2 Setup AC-2D601W device.....	29
9.1.3 Connect to COM port.....	31
9.2 Barcode printer.....	31
9.2.1 Setup Tepra server.....	31
9.2.2 Configure Android.....	32
9.2.3 Configure web browser.....	32
9.3 Barcode printer (Petit lapin).....	33
9.3.1 Setup SATO PT-208e.....	33
9.3.2 Configure Android for Petit lapin.....	33
9.3.3 Configure PC for Petit lapin.....	33

10	R-library Chelyabinsk	35
11	Okadai utilities	37
11.1	Vstool.....	37
11.2	Image.....	38
11.3	Emacs script.....	39
11.4	Matlab script.....	41
11.5	Ruby/Perl/Shell script.....	43
12	Machine gazer	45
13	Command index	46
14	Project index	48

The Toolkit Reference Manual

This document describes the Toolkit. In addition to this document, you may find useful information on Toolkit's home page (<https://dream.misasa.okayama-u.ac.jp/documentation/>).

1 Overview

For a long term geochemistry, rocks and datasets should be maintained physically and electronically. We have started to construct a rock depository named DREAM.

First, we have developed as a digital sentinel for the depository name Medusa, that is is a database server with web interface.

To manipulate records and process datasets by a personal computer, sets of utilities were developed. See Section 5.4 [Record operator Orochi], page 15. See Chapter 7 [Data format CASTEML], page 20. To correlate documents and rocks, we have developed L^AT_EX based program. See Section 5.5 [Document uploader Pmlatex], page 17. Since datasets are created by analytical instruments such as a mass spectrometer, eventually we want to correlate Medusa and the instruments.

The goal of this project is to build a toolkit that unites rocks, datasets, Medusa, and analytical instruments. We call it as the Fudo toolkit, after Emperor Go-Daigo who escaped Oki and initiated restoration on 1333 AD. Relationship between the Fudo toolkit and Medusa is like GNU and Linux. Medusa is the core but that is included in the Fudo toolkit.

2 Resource location

Packages and services are listed in this section. Most of packages are hosted by following sites.

‘Repository for packages -- public’

- <https://github.com/misasa>

‘Repository for packages -- local’

- <https://gitlab.misasa.okayama-u.ac.jp/>
- <https://database.misasa.okayama-u.ac.jp/repository/>

2.1 Packages

‘rails project -- medusa’

Keep track any stones and boxes. Medusa documents the origin of a rock, its current storage location, the relationship, if any between the original rock and any sub-samples, and correlate any and all geochemical datasets associated with the rock.

See Section 5.1 [Server Medusa], page 13, for detail. See [web app – Medusa for Echo], page 10, for application.

- Repository (<https://github.com/misasa/medusa>)
- Jenkins (<https://database.misasa.okayama-u.ac.jp/jenkins/job/Medusa>)
- Redmine (<https://database.misasa.okayama-u.ac.jp/redmine/projects/medusa/issues>)

‘gem package -- withrecursive’

An ActiveRecord extension to add recursive association.

This extension helps to create hierarchical queries for self-referential models and allows to traverse ancestors or descendants, recursively.

See [rails project – medusa], page 3, that refers this package.

- Repository (<https://github.com/misasa/withrecursive>)

‘gem package -- actsasmappable’

Generate Google map from ActiveRecord model. Integrate a dynamic map in a Rails application. It can display geographical map tiles loaded from OpenStreetMap. The included map viewer is Leaflet.

See [rails project – medusa], page 3, that refers this package.

- Repository (<https://github.com/misasa/actsasmappable>)

‘titanium project -- sisyphus’

Let Android App be client of Medusa.

See Section 5.2 [Labor Sisyphus for Android], page 13, for detail. See [gem package – sisyphus-for-medusa], page 4.

- Repository (<https://github.com/misasa/sisyphus>)
- Repository (<https://github.com/misasa/sisyphus-2022>)
- Jenkins (<https://database.misasa.okayama-u.ac.jp/jenkins/job/Sisyphus>)
- Redmine (<https://database.misasa.okayama-u.ac.jp/redmine/projects/sisyphus/issues>)

`'gem package -- sisyphus-for-medusa'`

Create stone in Medusa and print barcode label on MS-Windows. This is a Web App that runs on PC. User creates a stone with label by one action using web browser as one does it using See Section 5.2 [Labor Sisyphus for Android], page 13. See Section 5.3 [Labor Sisyphus for MS-Windows], page 15, for detail. See [titanium project – sisyphus], page 3.

See [gem package – tepra], page 5, that is referred by this package.

- Repository (<https://github.com/misasa/sisyphus-for-medusa>)
- Jenkins (<https://database.misasa.okayama-u.ac.jp/jenkins/job/sisyphus-for-medusa>)
- Redmine (<https://database.misasa.okayama-u.ac.jp/redmine/projects/sisyphus-for-medusa/issues>)

`'gem package -- orochi-for-medusa'`

A series of command-line utilities that manipulates records in Medusa. The utilities list, name, and find records in Medusa as GNU utilities do to files.

See Section 5.4 [Record operator Orochi], page 15, for detail.

See [gem package – medusa_rest_client], page 4, that is referred by this package. See [gem package – tepra], page 5, that is referred by this package.

- Repository (<https://github.com/misasa/orochi-for-medusa>)
- Jenkins (<https://database.misasa.okayama-u.ac.jp/jenkins/job/orochi-for-medusa>)
- Redmine (<https://database.misasa.okayama-u.ac.jp/redmine/projects/orochi-for-medusa/issues>)

`'gem package -- casteml'`

A series of comprehensive utilities that handles CASTEML. The utilities download/upload datasets from/to Medusa. The utilities join, split, and convert datasets.

See Chapter 7 [Data format CASTEML], page 20, for detail.

- Repository (<https://github.com/misasa/casteml>)
- Jenkins (<https://database.misasa.okayama-u.ac.jp/jenkins/job/Casteml>)
- Redmine (<https://database.misasa.okayama-u.ac.jp/redmine/projects/casteml/issues>)

`'gem package -- pmlatex'`

Upload PDF file created by `pmlatex.sty` and `pdflatex` to Medusa. Create bib record in Medusa, then upload corresponding PDF file to the record. Also replace PDF and update title, date, and author.

See Section 5.5 [Document uploader Pmlatex], page 17, for detail.

- Repository (<https://github.com/misasa/pmlatex>)
- Jenkins (<https://database.misasa.okayama-u.ac.jp/jenkins/job/pmlatex>)
- Redmine (<https://database.misasa.okayama-u.ac.jp/redmine/projects/pmlatex/issues>)

`'gem package -- medusa_rest_client'`

A series of Ruby interfaces to Medusa REST API. This gem is a client for Medusa REST API (<https://dream.misasa.okayama-u.ac.jp/documentation/MedusaRestAPI/>). Medusa REST API follows the Rails's RESTful conventions, so this gem interacts with the Medusa REST API via ActiveResource.

See [gem package – orochi-for-medusa], page 4, that refers to this package.

- Repository (https://github.com/misasa/medusa_rest_client)
- Jenkins (https://database.misasa.okayama-u.ac.jp/jenkins/job/medusa_rest_client)
- Redmine (<https://database.misasa.okayama-u.ac.jp/redmine/projects/medusarestclient/issues>)

`‘gem package -- tepra’`

Print QR-code by King Jim’s Tepra via command or REST interface. The program also works as label-print server. When the program is launched without argument, it runs as web server.

See [gem package – orochi-for-medusa], page 4, that refers to this package. See [gem package – godigo], page 7, that refers to this package. See [gem package – sisyphus-for-medusa], page 4, that refers to this package. See Section 9.2.1 [Setup Tepra server], page 31, for detail.

- Repository (<https://github.com/misasa/tepra>)
- Jenkins (<https://database.misasa.okayama-u.ac.jp/jenkins/job/Tepra>)
- Redmine (<https://database.misasa.okayama-u.ac.jp/redmine/projects/tepra/issues>)

`‘raspberrypi package -- DREAM-PI’`

DREAM-PI is a device that intermediate PubNub (<https://www.pubnub.com/>) interfaces and barcode reader, balances, and NFC reader/writer. As of January 25, 2018, we use computer (Raspberry Pi) as an infrastructure. As an example of applications of DREAM-PI, we develop a web interface named Imoko. He transfers quantity of a specimen weighted in balance into Medusa using DREAM-PI. As of January 19, 2018, one computer (Raspberry Pi) can communicate with two balances and one barcode reader at once.

See Chapter 6 [Stone measurer Imoko], page 19, for detail.

- Repository (https://github.com/misasa/dream_io)

`‘r package -- chelyabinsk’`

R package for geochemical datasets. This package reads and processes geochemical datasets, downloads CASTEML from Medusa, processes IONML, and plots typical geochemical datasets.

See Chapter 10 [R-library Chelyabinsk], page 35, for detail. See manual (<http://dream.misasa.okayama-u.ac.jp/documentation/chelyabinsk-manual.pdf>). See [gem package – casteml], page 4, that is referred by this package. See [r package – MedusaRClient], page 5, that is referred by this package.

- Repository (<https://github.com/misasa/chelyabinsk>)

`‘r package -- MedusaRClient’`

A series of R interfaces to Medusa. Using this package user can obtain data directly from Medusa.

See [r package – chelyabinsk], page 5, that refers this package.

- Repository (<https://github.com/misasa/MedusaRClient>)

`‘r package -- VisualAnalysis’`

R/Shiny interface to See [rails project – medusa], page 3, for visualizing geochemical data. This app uses Leaflet, which is the leading open-source JavaScript library for interactive maps, and r package – chelyabinsk visualize geochemical dataset served by Medusa.

See [r package – chelyabinsk], page 5. See [r package – MedusaRClient], page 5.

- Repository (<https://github.com/misasa/VisualAnalysis>)

‘python package -- image_mosaic’

A series of image utilities. Use this ImageMosaic to crop, rotate, or enlarge bitmap images. This package took over the project ‘python package – opencvtool’ on 2018-08.

See [gem package – vstool], page 8, that refers to this package. See [gem package – opencvtool], page 7, that refers to this package. See [rails project – medusa], page 3, that refers to this package. See [rake project – mosaic-sem], page 9, that refers to this package. See also spots-warp in See [gem package – multi_stage], page 7,

- Repository (https://github.com/misasa/image_mosaic)

‘rails project -- machine_time’

A machine-gazer who monitors ‘machine-time’ of analytical instruments. This program is Web App. The status is modified by this Web App or programs by ‘gem package – godigo’.

See Chapter 12 [Machine gazer], page 45, for detail. See [gem package – godigo], page 7, for client. See [web app – Session for machines], page 10, for application.

- Repository (https://github.com/misasa/machine_time)
- Redmine (<https://database.misasa.okayama-u.ac.jp/redmine/projects/machine/issues>)

‘www -- dream’

This repository includes files in the html root directory of DREAM’s web-site. There are two sites, that are staging web-site (<https://database.misasa.okayama-u.ac.jp/staging>) and public web-site (<https://dream.misasa.okayama-u.ac.jp/>).

This repository also includes the configuration file (<https://gitlab.misasa.okayama-u.ac.jp/DREAM/dream-site/blob/master/.gitlab-ci.yml>) for jobs to deploy the staging and public website. To update the site, edit index.php (<https://gitlab.misasa.okayama-u.ac.jp/DREAM/dream-site/blob/master/html/index.php>) via GitLab’s UI and push the revision by clicking Commit changes button.

- Repository (<https://gitlab.misasa.okayama-u.ac.jp/DREAM/dream-site>)

‘www -- toolkit’

This repository includes files in the html root directory of Toolkit’s web-site (<https://dream.misasa.okayama-u.ac.jp/documentation/>). To update the site, push to the master. The GitLab system keeps the site up-to-date.

- Repository (<https://gitlab.misasa.okayama-u.ac.jp/orochi/documentation>)
- Jenkins (<https://database.misasa.okayama-u.ac.jp/jenkins/job/Documentation>)

‘rails project -- epma_archiver’

Export maps obtained by EPMA. The maps are shown by web pages. The maps are converted from raw JEOL data. Although target device of this project is JEOL JXA-8530F, the project can handle datasets obtained by JEOL JXA-8800. This project took over See [rails project – jxa], page 7, from 2012-12.

See [web app – EPMA archiver], page 10, that is the application of this project.

- Repository (https://github.com/misasa/epma_archiver)

`'rails project -- jxa'`

Export point and area analyses obtained by EPMA. The maps are shown by web pages (obsolete). The maps are converted from raw JEOL data. Target device of this project is JEOL JXA-8800 only. This project was took over by See [rails project – epma_archiver], page 6, on 2012-12 and under maintenance mode.

See [web app – EPMA archiver for JXA-8800], page 10, that is the application of this project.

See [gem package – jxa1_api], page 7, to export stagelist from Web App using this project.

- Repository (<https://gitlab.misasa.okayama-u.ac.jp/rails/jxa>)
- Redmine (https://database.misasa.okayama-u.ac.jp/redmine/projects/epma_archiver/issues)

`'gem package -- multi_stage'`

Project coordinates to other space. It is useful to convert stage coordinate of certain device such for EPMA into global coordinate in VisualStage 2007.

See also `image-warp` in See [python package – image_mosaic], page 6. See [gem package – opencvtool], page 7, that this package refers to.

- Repository (https://github.com/misasa/multi_stage)
- Jenkins (https://database.misasa.okayama-u.ac.jp/jenkins/job/gem-multi_stage/)
- Redmine (https://database.misasa.okayama-u.ac.jp/redmine/projects/gem-multi_stage/issues)

`'gem package -- scalebar'`

Put scalebar on image. Create a \LaTeX file that includes scale bar or scale grids with indexes.

- Repository (<https://github.com/misasa/scalebar>)
- Jenkins (<https://database.misasa.okayama-u.ac.jp/jenkins/job/scalebar>)
- Redmine (<https://database.misasa.okayama-u.ac.jp/redmine/projects/gem-scalebar/issues>)

`'gem package -- opencvtool'`

Ruby package that provides Ruby API that utilizes python package – `image_mosaic`. See [gem package – multi_stage], page 7, that refers to this package. See [python package – image_mosaic], page 6, that is referred by this package. See [rake project – mosaic-sem], page 9, that refers to this package.

- Repository (<https://github.com/misasa/opencvtool>)
- Redmine (<https://database.misasa.okayama-u.ac.jp/redmine/projects/gem-opencvtool/issues>)

`'gem package -- jxa1_api'`

Obtain stagelist from Web App using `'rails project – jxa'` (obsolete). See [rails project – jxa], page 7, that refers to this package.

- Repository (https://gitlab.misasa.okayama-u.ac.jp/gems/jxa1_api)
- Redmine (https://database.misasa.okayama-u.ac.jp/redmine/projects/jxa1_api/issues)

`'gem package -- godigo'`

Ruby package with command-line client utilities to start and stop a session on a machine-gazer `'rails project – machine_time'`. The utilities help to keep track of

machine status. This package also offers interface for synchronization of datasets. Action can be start, stop, and sync.

See Chapter 12 [Machine gazer], page 45, for detail. See [rails project – machine_time], page 6, for the server.

See [gem package – tepra], page 5, that is referred by this package.

- Repository (<https://github.com/misasa/godigo>)
- Jenkins (<https://database.misasa.okayama-u.ac.jp/jenkins/job/Godigo>)
- Redmine (<https://database.misasa.okayama-u.ac.jp/redmine/projects/gem-godigo/issues>)

`'gem package -- machine_time_client'`

Ruby package to connect machine_time object and machine-gazer's RESTful web service. A client for interacting with a machine-gazer by See [rails project – machine_time], page 6, by REST. The machine-gazer REST API follows the Rails's RESTful conventions, so this gem uses ActiveResource to interact with the machine-gazer REST API. This gem allows Ruby developers to access the API and manage machine_time object on the machine-gazer.

See [gem package – godigo], page 7, that refers to this library.

- Repository (https://github.com/misasa/machine_time_client)
- Redmine (https://database.misasa.okayama-u.ac.jp/redmine/projects/machine_time_client/issues)

`'gem package -- vstool'`

A series of VisualStage 2007 related utilities. Obtain stage coordinate from VisualStage 2007, transform and export the image, and attach the image to VisualStage 2007.

See Section 11.1 [Vstool], page 37, for detail.

See [python package – image_mosaic], page 6, that this package refers to. See [python package – vs2007], page 8, that this package refers to.

- Repository (<https://github.com/misasa/vstool>)
- Jenkins (<https://database.misasa.okayama-u.ac.jp/jenkins/job/gem-vstool>)
- Redmine (<https://database.misasa.okayama-u.ac.jp/redmine/projects/vstool/issues>)

`'gem package -- visual_stage'`

A Ruby wrapper for 'python package – vs2007'. To control and refer to VisualStage 2007 from Ruby, call this Ruby gem package.

See [python package – vs2007], page 8, that this package refers to.

- Repository (https://github.com/misasa/visual_stage)
- Jenkins (https://database.misasa.okayama-u.ac.jp/jenkins/job/gem-visual_stage)
- Redmine (https://database.misasa.okayama-u.ac.jp/redmine/projects/visual_stage/issues)

`'python package -- vs2007'`

Control and refer to the Windows application 'VisualStage 2007'. This package offers two programs (`vs` and `vs-api`) that communicate with the program 'VisualStage 2007'. With the two commands, one can write a program that talks to VisualStage

2007. The command `vs` is for starting/stopping the program `VisualStage 2007`, opening/closing its data file, and outputting its address and attachment. The program `vs-api` is for sending local command to `VisualStage-API` with an argument.

See [gem package – `visual_stage`], page 8, and `vs_attach_image.m` in See [matlab script – `VisualSpots`], page 9, that refers to this package.

- Repository (<https://github.com/misasa/vs2007>)
- Redmine (<https://database.misasa.okayama-u.ac.jp/redmine/projects/vs2007/issues>)

‘python package -- `vs2020`’

Read stage position and move stage through the Windows application ‘`VisualStage 2020`’. This package offers a program (`vs2020-sentinel`) that communicates with the program ‘`VisualStage 2020`’.

- Repository (<https://github.com/misasa/vs2020>)

‘python package -- `jxmap`’

Convert a map file created by JEOL’s EPMA to a tiff file. Datasets of area analyses using JEOL’s EPMA are stored in file with extension `*.map`. A program by this package reads `*.map` file and writes `*.tiff` file. Also, a program by this package identifies phases using algorithm k-mean.

See [web app – `EPMA archiver`], page 10, that refers to this package. See [rails project – `jxa`], page 7, that refers to this package.

This project took over rake project – `jxa1` from 2012-12.

- Repository (<https://github.com/misasa/jxmap>)
- Redmine (<https://database.misasa.okayama-u.ac.jp/redmine/projects/jxmap/issues>)

‘rake project -- `mosaic-sem`’

Create a mosaic from images by JEOL JSM-7001F accompanied with `imajeometry` files (obsolete). Images are typically obtained by JEOL JSM-7001F. This is a rake project with templates. This works only when local environment is setup properly.

See [gem package – `opencvtool`], page 7, that refers to this package. See [python package – `image_mosaic`], page 6, that refers to this package.

- Repository (<https://gitlab.misasa.okayama-u.ac.jp/DREAM/mosaic-sem>)

‘emacs package -- `fudo`’

Scripts written in Emacs.

See Section 11.3 [Emacs script], page 39, for detail.

- Repository (<https://database.misasa.okayama-u.ac.jp/repository/bin/dot.emacs.d>)

‘ruby/perl/shell script -- `xtreeml`’

Scripts written in Ruby, Perl, and Shell.

See Section 11.5 [Ruby/Perl/Shell script], page 43, for detail.

- Repository (<https://database.misasa.okayama-u.ac.jp/repository/xtreeml/bin>)

‘matlab script -- `VisualSpots`’

Scripts written in Matlab.

See Section 11.4 [Matlab script], page 41, for detail.

- Repository (<https://gitlab.misasa.okayama-u.ac.jp/toolkit/matlab>)

2.2 Services

‘web app -- Medusa for Echo’

See [rails project – medusa], page 3.

- Master server (<https://database.misasa.okayama-u.ac.jp/stone>)
- Staging server (<https://database.misasa.okayama-u.ac.jp/staging>)
- Public server for open access (<https://dream.misasa.okayama-u.ac.jp/pub>)

‘ID Query’

Return URL of page for a stone <https://dream/?q=stone-ID>.

- <https://dream.misasa.okayama-u.ac.jp/?q=20131115155414-709544>

‘web app -- EPMA archiver’

See [rails project – epma_archiver], page 6.

Export area analyses on JXA-8530F.

- <https://archive.misasa.okayama-u.ac.jp/JXA-8530F>

‘web app -- EPMA archiver for JXA-8800’

See [rails project – jxa], page 7.

Export point and area analyses on JXA-8800.

- <https://archive.misasa.okayama-u.ac.jp/jxa1>

‘Zabbix for DREAM’

Monitor DREAM services and send messages on failure.

- <https://itokawa.misasa.okayama-u.ac.jp/zabbix/>

‘Jenkins for Fudo’

- <https://database.misasa.okayama-u.ac.jp/jenkins>

‘web app -- Medusa for demo’

See [rails project – medusa], page 3.

- <https://dream.misasa.okayama-u.ac.jp/demo>

‘web app -- Session for machines’

See [rails project – machine_time], page 6.

- <https://database.misasa.okayama-u.ac.jp/machine>

‘Project manager for Fudo development’

- <https://database.misasa.okayama-u.ac.jp/redmine/projects>

3 Install

Most programs are installed as Ruby gem. Install Ruby 2.7 (86 or 64) without Devkit then install gem as following. For MS-Windows, we recommend RubyInstaller for Windows (<http://rubyinstaller.org/>). By installing single gem package, several programs will be placed.

```
$ gem source -a https://dream.misasa.okayama-u.ac.jp/rubygems/  
$ gem install gem-name
```

Locate a configuration file `~/.orochirc` with following lines to access Medusa.

```
---  
uri: https://dream.misasa.okayama-u.ac.jp/demo  
user: admin  
password: admin
```

All commands are listed in Command index. See Chapter 13 [Command index], page 46. Most of them are with online help. Call them with help option as shown below.

```
$ <program-name> --help
```

4 Changelog

‘2018-09-22’

Local server relocated from ‘devel’ to ‘gitlab’.

‘2018-08-12’

Revise structure of manual significantly.

‘2018-05-24’

Typically the server is configured to offer automatic transfer from http to https. However, **Sisyphus for Android** does not work with the transfer. The detail is described.

‘2018-03-31’

Sisyphus: Prevent duplicated registration. Add option to start with camera on registration of a stone. Obtain list of printers from printer server ‘Tepra’.

‘2018-02-16’

Revise structure of manual significantly.

‘2017-03-31’

Sisyphus: Login to Medusa using Okadai ID card. Add capabilities to list history and search online.

‘2016-10-03’

Add [r package – chelyabinsk], page 5, to the list.

‘2016-06-30’

Move Medusa related projects to GitHub (<https://github.com/misasa/>).

‘2016-06-12’

The manual is converted to Texinfo format.

‘2016-03-31’

Sisyphus: Update status of stone in Medusa 9.2 using inventory API. Add ability to read and write NFC tag.

‘2014-03-31’

Sisyphus: Refactor codes to be consistent with Medusa 9.0.

‘2013-03-31’

Sisyphus: Add capability to print barcode to label printer SATO Petit lapin. See Section 9.3 [Barcode printer (Petit lapin)], page 33.

5 Stone gazer

5.1 Server Medusa

See `medusa`. See [rails project – medusa], page 3. Medusa is the core program in the Fudo toolkit.

5.2 Labor Sisyphus for Android

See `sisyphus-en`. See [titanium project – sisyphus], page 3.

There are several ways to update status of a stone. The most fundamental way is to use web browser. More casual and fast approach is to utilize smart phone.

The goal of this project is let smart phone be a Medusa client utilizing its Camera to take photograph and scan barcode, and NFC capability. As of February 25 (2018), we only support Android. The screen of **Sisyphus for Android** is shown in [Figure 2], page 14.

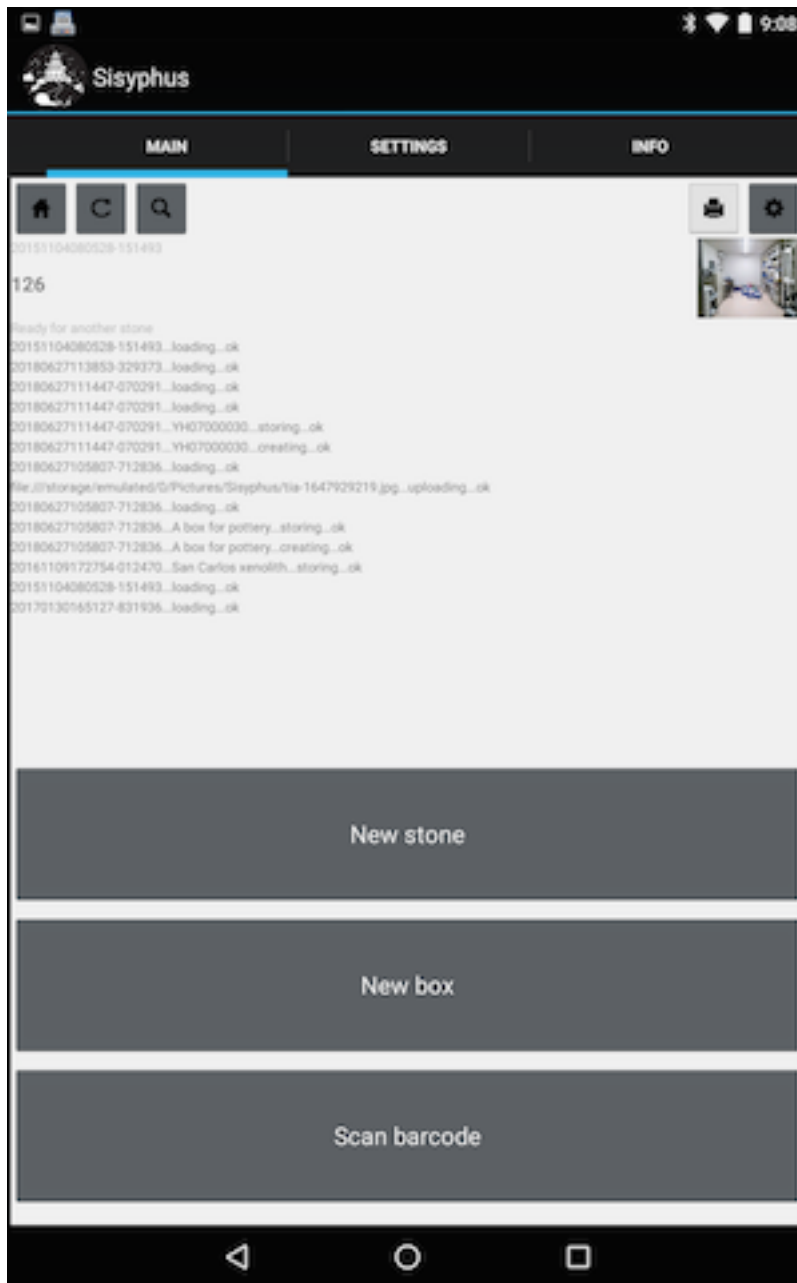


Figure 2: Screen of Sisyphus

Sisyphus for Android creates a new stone, a new box, and a link between a stone and a box. To create a new stone or a new box, tap 'New stone' button, or 'New box' button, respectively. To make a links, (1) load pre-existing box by tapping 'reload' button and scan barcode tag equipped with the box, and (2) load pre-existing stone by tapping 'Scan barcode' button and scan barcode tag equipped with the stone. When there is the link already, the operation updates inventory status of the stone.

Sisyphus for Android can print QR-code label. To do so, setup label printer first. See Section 9.2 [Barcode printer], page 31. Section 9.3 [Barcode printer (Petit lapin)], page 33. To

configure **Sisyphus for Android**, tap ‘Settings’ tab. When you want to take photo first on creation of a new stone, check appropriate option.

Sisyphus for Android can scan not only barcode tag but also NFC tag. Also you can let **Sisyphus for Android** log into **Medusa** using NFC ID card.

As of October 7, 2020, **Sisyphus for Android** can only talk ‘https:’.

5.3 Labor Sisyphus for MS-Windows

See [gem package – sisyphus-for-medusa], page 4.

The most basic action for **Medusa** is to create a record for a stone. This package offers Web App that lets a user create a stone and have a label at the same time.

To create a record in **Medusa** and have a label at the same time, you use Web App **sisyphus** that runs on MS-Windows. Functionality that the Web App provides is subset of those by Android App **Sisyphus**. See Section 5.2 [Labor Sisyphus for Android], page 13. If the Android App is working great, this tool is not necessary for you.

Setup label printer first. See Section 9.2 [Barcode printer], page 31. Then install gem package -- **sisyphus-for-medusa** as shown below.

```
CMD> gem source -a https://dream.misasa.okayama-u.ac.jp/rubygems/
CMD> gem install sisyphus-for-medusa
CMD> gem update sisyphus-for-medusa
CMD> which sisyphus
/usr/local/bin/sisyphus
CMD> sisyphus
```

A program named **sisyphus.bat** will be installed to somewhere appropriate. The program is Web App that runs on <http://localhost:8890/>. Check where it is by **which sisyphus** and launch it. Access to it by web browser. On creation of a record, a label would be created automatically.

Create a shortcut of **sisyphus.bat** and place it in `~/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/Startup/`. Also create a bookmark to <http://localhost:8890/> and keep it in Bookmarks Toolbar on web browser.

5.4 Record operator Orochi

See [gem package – orochi-for-medusa], page 4.

Utilities that talk to **Medusa** is named **orochi-for-medusa**. They run with Ruby on MS-Windows.

```
$ gem source -a https://dream.misasa.okayama-u.ac.jp/rubygems/
$ gem install orochi-for-medusa
$ gem update orochi-for-medusa
```

The program reads a configuration file `~/.orochirc`. The file should look like below.

```
---
uri: https://dream.misasa.okayama-u.ac.jp/demo
user: admin
password: admin
```

Commands shown below allow you to manipulate records in **Medusa** as if GNU tools do on files and directories.

```
orochi-help [Command]
List all orochi commands
```

orochi-cd	[Command]
Change the orochi working box	
orochi-ditto	[Command]
Clone box recursively	
orochi-download	[Command]
Download full datasets for poly-families	
orochi-find	[Command]
Search Medusa by keyword	
orochi-label	[Command]
See Section 9.2 [Barcode printer], page 31. Create barcode label with Medusa-ID and stone-name. It takes stone-ID as an argument and obtains stone-name from Medusa. Then it tosses them to an external program <code>tepra</code> . This is useful when you do not know name of stone but ID.	
<pre>CMD> orochi-label 20110119154409-142363 ⇒ tepra print "20110119154409-142363,Heaven"</pre>	
When failed to resolve, it creates a label without name.	
<pre>CMD> orochi-label 19991231-999999 ⇒ tepra print "19991231-999999,19991231-999999"</pre>	
orochi-ls	[Command]
List box contents	
orochi-mkstone	[Command]
Create a stone (or box), if it does not already exist	
orochi-mv	[Command]
Store a stone to a box	
orochi-name	[Command]
Query stone-ID to Medusa and returns its name	
orochi-open	[Command]
Open record by default browser	
orochi-place	[Command]
Search Medusa and return latitude and longitude	
orochi-pwd	[Command]
Print full hierarchy of orochi working box	
orochi-refresh-tile	[Command]
Refresh tiles in layers on a surface	
orochi-rename	[Command]
Rename record or change attribute	
orochi-rm	[Command]
Remove a specified record	
orochi-stone-in-box	[Command]
Transform stone to box	

oroichi-uniq	[Command]
Repeat only one stone in family	
oroichi-upload	[Command]
Upload any files to web service by Medusa	
oroichi-url	[Command]
Show record in standard output by curl and w3m	

With iRuby and `medusa_rest_client`, direct edition of records is possible. First access to a record of concern using web browser. You see corresponding internal ID such as '1950'. Then access again with appending '.json' to the URL. You see name of properties. Launch iRuby and edit the record as shown below.

```
$ irb
irb(main):001:0> require "medusa_rest_client"
=> true
irb(main):002:0> include MedusaRestClient
=> Object
irb(main):003:0> obj = Specimen.find(1950)
=> #<MedusaRestClient::Specimen:0x007ff7d24d19e0 @attributes={"id"=>1950, "name"=>"AD03", "
irb(main):006:0> obj.age_min
=> 9999
irb(main):007:0> obj.age_min = 1060
=> 1060
irb(main):008:0> obj.save
=> true
irb(main):009:0>
```

5.5 Document uploader Pmlatex

See [gem package – pmlatex], page 4. These commands scan the report, upload the report, and correlate samples to the report.

pmlatex -help	[Command]
Give you an idea about a work flow.	
pmlatex add	[Command]
Create a new bib in Medusa and insert the bib-ID	
pmlatex commit	[Command]
Upload and replace pdf file in Medusa	
pmlatex update	[Command]
Update title and author of a bib in Medusa	
pmlatex scan	[Command]
Scan a document and correlate stones with the bib	

To store a report to Medusa requires several steps. A bibliography record should be created. A report in PDF file should be uploaded and linked to the bibliography record. Stones referred in the report should be correlated to the bibliography record. A suite of programs placed by 'gem package – pmlatex' helps the steps in a following way.

```
$ ls
report.tex  report.pdf
$ pmlatex add report
```

```

⇒ create a new bib in Medusa and insert the bib-ID
$ pdflatex report
⇒ compile to have a pdf file
$ reducepdf report.pdf
⇒ reduce size of a pdf file
$ pmlatex commit report
⇒ upload the pdf file to Medusa as a bib
...
$ pmlatex update report
⇒ update title and author of a bib in Medusa
$ pmlatex commit report
⇒ replace pdf in Medusa

```

It is convenient to have a local bib file with all stones and all bibles stored in Medusa such as `~/texmf/bibtex/bib/ref_dream.bib`. As of June 17, 2016, you cannot download `ref_dream.bib` from Medusa yet.

5.6 Surface with images

On a plain ‘surface’ of a sample, a coordinate should be defined first. We refer to the coordinate as device-zero (or world) coordinate.

An image is obtained by microscopy such as secondary electron microscopy. The image should be accompanied with an Affine matrix to project the image to the device coordinate. Essentially the Affine matrix contains information of x-and-y coordinate, magnification, and angle of rotation.

An image projected in the device coordinate can be re-projected into the device-zero coordinate using an Affine matrix of a device-zero coordinate relative to a device coordinate. Essentially, the Affine matrix contains information of x-and-y shift and angle of rotation.

Since each device coordinate is fragile and not important, Medusa stores an image with a combined Affine matrix which is a product of an Affine matrix of an image to device coordinate and an Affine matrix of device-zero coordinate relative to device coordinate. We refer to an image with the combined Affine matrix as a Image. As inferred already, a Image can be projected into device-zero coordinate. The surface is visualized by overlays of Images that are projected into the device-zero coordinate.

Medusa creates a ‘mosaic’ that consists of multiple layers of images that is projected into the device-zero coordination. Dimension of a mosaic is determined to include all Image.

To improve performance on visualization, Medusa prepares a mosaic in five resolutions, that are 256x256, 512x512, 1024x1024, 2048x2048, 4096x4096, 8192x8192.

As of September 21, 2017, a surface of `mnt-CBK1b-i <stone: 20160820170853-707954>` <https://dream.misasa.okayama-u.ac.jp/?query=20160820170853-707954> is visualized by Images at <https://archive.misasa.okayama-u.ac.jp/R/surfaces/?id=20160820170853-707954>.

6 Stone measurer Imoko

See [raspberry package – DREAM-PI], page 5.

There are several ways to update status of a stone. The most fundamental way is to use web browser. More robust and fast approach is to write a client that runs on PC utilizing barcode reader and NFC reader. Connection between PC and barcode reader is via Bluetooth while that between PC and NFC reader is USB. The connection between PC and the devices are fragile and it is not easy to maintain reliable connection, with paying day-to-day effort.

Thus we decided to setup a dedicated computer (Raspberry Pi). The relationship among device is shown below. As of May 8, 2017, interface to NFC is not yet implemented.

```
browser <http> web interface <PubNub> Computer
Balance <USB><RS232C> Computer <Bluetooth> Barcode reader
```

Imoko is web interface to update quantity of stones using DREAM-PI. Typical operation, after setup, is shown below.

- Open Imoko (<https://database.misasa.okayama-u.ac.jp/io/>).
- Click Start to reboot the DREAM-PI to ensure reliable connection.
- Scan connection code to hear beep twice. The code is shown in Imoko. If barcode reader cannot scan it, find the connection code printed on label.
- Scan stone-ID on label of the stone. Confirm that current stone by updated.
- Put a stone on balance and wait until the read gets settled down.
- Push ‘Measure’ button on Imoko.html to update the quantity of a stone in a record on Medusa.

Imoko can create barcode label via printer. To specify the printer, configure DREAM-PI.

7 Data format CASTEML

7.1 CASTEML Introduction

CASTEML is a CASTEM language that describes chemical property (or ‘abundance’) of specimen (or ‘stone’). Chemical property (or ‘abundance’) is obtained by session (or ‘analysis’). CASTEML can describe multiple stones as well. CASTEML with mono stone and multiple stones are referred as mono CASTEML and multi CASTEML, respectively.

Until 2019, CASTEM accepts only single session to describe abundances of a stone, even they were obtained by a few sessions. Thus abundances described by several sessions were re-described in a virtual session.

From September, 2019, CASTEML accepts a few sessions to describe a stone. To have a CASTEML with several sessions, create a table that is linked to a bibliography in Medusa, and download the table using `casteml download`. Most of tables in Medusa describes chemical properties of multiple stones that were determined by several sessions.

7.2 CASTEML commands

CASTEML is processed by a series of programs. Some download and upload datasets from or to Medusa, and others join, split, and convert CASTEML files. See [gem package – casteml], page 4.

On subsection, primitive commands and links for genetic operations are summarized.

‘Primitive commands’

<code>casteml -help</code>	[Command]
Return list of sub commands.	
<code>casteml sub-command -help</code>	[Command]
Show help for sub-command.	
<code>\$ casteml download --help</code>	
Usage: <code>casteml download [options] ID0 [ID1 ...] [options]</code>	
...	
<code>casteml download</code>	[Command]
Download CASTEML file from Medusa.	
<code>casteml join</code>	[Command]
Join several CASTEML files to a single multi-CASTEML file.	
<code>casteml split</code>	[Command]
Split one multi-CASTEML file into multiple CASTEML files.	
<code>casteml upload</code>	[Command]
Upload a CASTEML file to Medusa.	
<code>casteml convert</code>	[Command]
Convert (pml csv tsv isorg) to (pml csv tsv isorg org dataframe tex pdf).	
<code>casteml spots</code>	[Command]
Export spots info in a CASTEML file to a L ^A T _E X file.	
<code>casteml download</code>	[Command]
Download CASTEML file from Medusa.	

Note that `casteml join`, `casteml download`, and `casteml convert` do not care about duplicates of records. We recommend to convert to CSV file and manipulate them using external tool such as `uniq`.

‘More genetic operations’

To upload spot analyses with image by `casteml upload`, see See Section “Publish spots information” in `stone-operation-en`.

To download and revise CASTEML files, see See `stone-operation-ja`.

Small tools that deal CASTEML files are shown as below. See Section 11.3 [Emacs script], page 39. See Section 11.4 [Matlab script], page 41. See Section 11.5 [Ruby/Perl/Shell script], page 43.

7.3 CASTEML polymorph

Datasets described by CASTEML can be described in different formats (or polymorphs), which are listed in [Table 7], page 21. The ‘R:pmlame’ is defined as a two-dimensional dataframe with row of stone and column of chem under R environment. The polymorphs can be obtained from CASTEML. The relationship between CASTEML and polymorphs as shown in [Figure 1], page 21. Brief descriptions of convert commands are shown in [Table 8], page 21.

Row and column of ‘CSV:dflame’ correspond to chem and stone, respectively. This system is used in Janouek et al. [1]. See section 3.6 ‘Spiderplots’ in it.

- [1] Janouek, V., Moyer, J. F., Martin, H., Erban, V., and Farrow, C. (2016). Geochemical modelling of igneous processes-principles and recipes in R language. Springer, Berlin, doi:10.1007/978-3-662-46792-3, 10, 978-3. <https://dream.misasa.okayama-u.ac.jp/?q=20160405094928-467961>

Table 7: Row and column of polymorphs.

	row	column
R:pmlame	stone	chem
CSV:tblame	stone	chem
ortbl:tblame	stone	chem
CSV:dflame	chem	stone

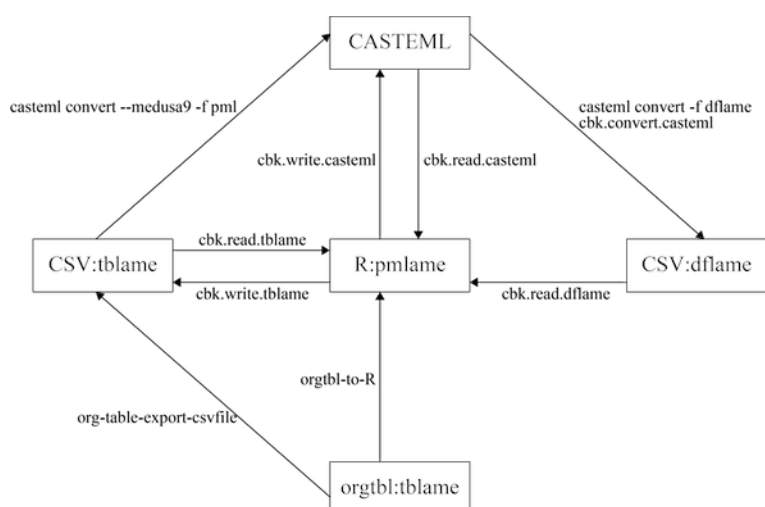


Figure 1: Relationship between CASTEML and polymorphs of CASTEML, and commands for conversion.

Table 8: Convert commands between CASTEML and polymorphs.

	from	to	remark
casteml convert -medusa9 -f pml	CSV:tblame	CASTEML	See Chapter 7 [Data format CASTEML], page 20.
casteml convert -f dflame	CASTEML	CSV:dflame	See Chapter 7 [Data format CASTEML], page 20.
cbk.convert.casteml	CASTEML	CSV:dflame	See [r package chelyabinsk], page 5.
cbk.write.casteml	R:pmlame	CASTEML	See [r package chelyabinsk], page 5.
cbk.read.casteml	CASTEML	R:pmlame	See [r package chelyabinsk], page 5.
cbk.read.tblame	CSV:tblame	R:pmlame	See [r package chelyabinsk], page 5.
cbk.write.tblame	R:pmlame	CSV:tblame	See [r package chelyabinsk], page 5.
cbk.read.dflame	CSV:dflame	R:pmlame	See [r package chelyabinsk], page 5.
orgtbl-to-R	orgtbl:tblame	R:pmlame	See Section 11.3 [Emacs script], page 39.
org-table-export-csvfile	orgtbl:tblame	CSV:tblame	See Section 11.3 [Emacs script], page 39.
castbl-to-latex	orgtbl:tblame	CASTEML	See Section 11.3 [Emacs script], page 39.
pml	clipboard	CASTEML	See Section 11.5 [Ruby/Perl/Shell script], page 43.

8 Data format IONML

See Chapter 10 [R-library Chelyabinsk], page 35.

The IONML is xml file that contains one acquisition, that is with ion signals of multiple elements with certain cycles, using mass spectrometry.

The IONML can be in form of pmlame which is R object referred as ‘ion-type pmlame’ and ‘IONCSV’ that consists of columns of time and ion intensities. The first column of each line should be number of ‘cycle’. Colname of the IONCSV should be ‘time’ and name of element followed by atomic weight (‘Si29’ instead of ‘29Si’).

Table 9: Structure of polymorphs of IONML.

	row	column
R:pmlame	cycle	intensity
CSV:IONCSV	cycle	intensity

8.1 Create IONML

See Chapter 10 [R-library Chelyabinsk], page 35.

To create IONML, convert raw data file originated from a mass spectrometer using appropriate script shown below.

`ionml-exp.rb` [RubyScript]

Convert text file `*.exp` from Thermo Triton to ionml.

`ionml-icp.rb` [RubyScript]

Convert text file `*.xls` from Agilent 7500cs to ionml.

`ionml-vg5400.rb` [RubyScript]

Convert text file `*.nob` from VG 5400 to ionml.

`ionml-ims1270.rb` [RubyScript]

Convert text file `asc_*.ais` from Cameca ims-1270 to ionml.

`ionml-ims5f.rb` [RubyScript]

Convert text file `*.asc` from Cameca ims-5f to ionml.

`ionml-ims1280.rb` [RubyScript]

Convert text file `*.asc` from Cameca ims-1280 to ionml.

`ionml-laicpqms.R` [Rscript]

Convert text file `*.csv` from Thermo iCAP-Q to ionml. See Chapter 10 [R-library Chelyabinsk], page 35.

`ionml.convert.ioncsv` [Rfunc]

Convert ioncsv file `*.ion` from Thermo iCAP-Q to ionml. See Chapter 10 [R-library Chelyabinsk], page 35.

`ionml-laicpqms.plx` [PerlScript]

Convert text file `*.csv` from Thermo iCAP-Q to ionml.

Makefile that creates several ionml files at once is shown below as an example.

```
# -*- mode: makefile -*-
SRCS:=$(wildcard *.asc)
TARGETS:=$(SRCS:%.asc=%.ionml)
```

```

CC:=ruby
CFLAGS:=-S
SCRIPT:~/xtreeml/bin/ionml-ims1280.rb

.PHONY: all clean
all: $(TARGETS)

%.ionml: %.asc $(SCRIPT) Makefile
$(CC) $(CFLAGS) $(SCRIPT) $< > $
clean:
rm -f $(TARGETS)

```

8.2 Read IONML

See Chapter 10 [R-library Chelyabinsk], page 35. See Section 11.4 [Matlab script], page 41.

To read IONML, use Matlab script or R script as shown below.

`cbk:read.ionml` [Rfunc]
 Return raw signal of the whole period. See [r package – chelyabinsk], page 5.

`ionml.read.laicpqms` [Rfunc]
 Return ionic ratio online with statistical information. See [r package – chelyabinsk], page 5.

`ionml.read.session` [Rfunc]
 Return average and standard error of ionic ratio. See [r package – chelyabinsk], page 5.

`readionml.m` [MatlabScript]
 Read IONML and return raw signal of the whole period. See Section 11.4 [Matlab script], page 41.

Two examples to process LA-ICP-QMS acquisition is shown below. Issue R commands or write an R script.

```

$ ls
ref_gl_tahiti@10um@x80@1.csv
$ Rscript ionml-laicpqms.R --input "ref_gl_tahiti@10um@x80@1.csv"
$ ls
ref_gl_tahiti@10um@x80@1.csv ref_gl_tahiti@10um@x80@1.xml
R> library(chelyabinsk)
R> pmlame0 <- cbk.read.ionml("ref_gl_tahiti@10um@x80@1.xml",representative_time=TRUE)
R> time <- pmlame0["time"]
R> intlame <- pmlame0[colnames(pmlame0)!="time"]
R> chem <- colnames(intlame)
R> col <- rainbow(length(chem))
R> matplot(time,intlame,type="l",col=col,lty=1,xlab="time",ylab="intensity")
R> legend("topleft",chem,col=col,lty=1,ncol=2,cex=0.5)

#!/usr/bin/env Rscript
library(chelyabinsk)
rplotfile <- 'Rplots.pdf'; pdf(rplotfile)
pmlame0 <- cbk.read.ionml("ref_gl_tahiti@10um@x80@1.xml",representative_time=TRUE)
time <- pmlame0["time"]
intlame <- pmlame0[colnames(pmlame0)!="time"]
chem <- colnames(intlame)
col <- rainbow(length(chem))

```

```
matplot(time,intlame,type="l",col=col,lty=1,xlab="time",ylab="intensity")  
legend("topleft",chem,col=col,lty=1,ncol=2,cex=0.5)  
dev.off()
```

9 Barcode operation

9.1 Barcode reader

9.1.1 Setup code device

‘USB - Bluetooth adapter’

When a computer is not with Bluetooth interface, you have to install USB-Bluetooth adapter. A device with USB such as those by SANWA SUPPLY doesn't work. Those without CSR woke such for those by IO-DATA.

‘Connect to PC’

Make the PC to the waiting state for adding a Bluetooth device, and make the barcode reader in discovery mode.

‘Enter PIN code’

PC will asked to enter PIN code, so enter 12345678. If the barcode reader cannot connect to your PC, read PIN Reference code 12345678 and Submit PIN with the code reader.


‘Create a quick connect barcode’

To let barcode reader know Bluetooth address of the device, you have to prepare a device-connection code. Find 6-byte Bluetooth-address of the computer. ‘More Bluetooth setting’ -> ‘hardware’ tab -> ‘Devices’ -> Choose ‘Generic Bluetooth Radio’ -> ‘Properties’ -> ‘Advanced’ tab -> ‘Address:’(e.g., 00:1b:dc:06:ac:d6) Visit vendor's web (<https://ww2.codecorp.com/bdaddr.php>) and enter 12-digit numbers that you got such as 001bdc06acd6. Print the code and hang it around the computer.

‘Connect to COM port’

Add a COM port to PC that runs MS-Windows by following steps. ‘More Bluetooth setting’ -> ‘Option’ tab -> Check all checkboxes -> ‘COM Ports’ tab -> ‘Add...’ button After a while the driver will be installed and the new number will be assigned to the COM port.

Setup-codes for barcode reader is shown in a table below. There are more options. For example, you can change time duration before sleep. See manual for detail.

code	remark
 M052_01	

(a)

Clear All CodeXML Rules



M418_02

— Reset RS232 Factory Defaults

(b)



M261_01

QR Code On

(c)



M170_04

(d)

Set Suffix to CR+LF



M746_01

(e)

Enable Auto Save Last Bluetooth Address



M121_01

(f)

5 min instead of 90 s before sleep



M587_02

(g)

notify on out of range



(h)

Save Settings

9.1.2 Setup AC-2D601W device

‘USB - Bluetooth adapter’

When a computer is not with Bluetooth interface, you have to install USB-Bluetooth adapter. A device with USB such as those by SANWA SUPPLY doesn't work. Those without CSR woke such for those by IO-DATA.

‘Connect to COM port’

Add a COM port to PC that runs MS-Windows by following steps. ‘More Bluetooth setting’ -> ‘Option’ tab -> Check all checkboxes -> ‘COM Ports’ tab -> ‘Add...’ button After a while the driver will be installed and the new number will be assigned to the COM port.

‘Activate COM port’

After describing the COM port in C:\Users\dream\.orochirc and start tepra-duplicate.bat. Or launch Tera Term to specify the COM port added to serial port, and click OK.

‘Initialize AC-2D601W’

Scan barcode below with large button on AC-2D601W.



‘Disable cable and memory mode’

Scan the barcode below to lock mode so that you don't accidentally change settings by pressing button.



‘Register Bluetooth MAC address’

Find Bluetooth MAC address to connect to and register it with AC-2D601W.



‘Connect to PC’

Make the PC to the waiting state for adding a Bluetooth device, and make the barcode reader in pairing mode.

**‘Disable device power saving’**

If you do not make this setting, pairing may be lost. Turning off Bluetooth power save via the following routes. Device Manager -> Bluetooth Device -> Properties -> Power Management -> Uncheck "Allow the computer to turn off this device to save power".

For USB Bluetooth, cancel the USB power save in the same way.

‘To scan QRcode’

Unplug the USB charging cable before scanning.

9.1.3 Connect to COM port

Add a COM port to PC that runs MS-Windows by following steps. Go ‘Control Panel’ ‘Bluetooth Devices’ ‘Bluetooth Settings’. Check ‘Allow Bluetooth devices to find this PC’ for initial connection. Then switch to ‘COM Ports’ tab and click ‘add’. Create a port with ‘incoming’ property. Typically ‘Serial Port Name’ of the new port is COM4. Clear all ports to avoid confusion when necessary. Also remove all ‘Code Reader’ related icons from ‘Devices and Printers’ in Control Panel when necessary.

Note that communication between software and scanner consists of two steps, that are scannerCOM4 and COM4software. The former is via Bluetooth and latter has nothing to do with Bluetooth. In other word, software does not know if the input is from Bluetooth.

You can check Bluetooth connection by Tera Term (<http://sourceforge.jp/projects/ttssh2/>). Launch it and try to monitor input from the serial port. If the software recognizes the port dedicated for Bluetooth connection, scan the device-connection code. If asked, pair the device by typing the passcode. The scanner is supposed to beep once. If it beeps multiple times (like four), try to scan the code again. Scan a barcode on your stone. You will see input from the scanner at console of ‘Tera Term’.

The port should be described in a configuration file. Orochi applications refer to `/.orochirc` as of January 27, 2020. Device number on Cygwin is one less than COM number on MS-Windows. The port COM4 should be expressed as serial_port: `/dev/ttyS3` in the file.

9.2 Barcode printer

As of January 24 (2020), we support Tepra by Japanese vendor King Jim. Install device driver and program SPC 10 provided by King Jim. Before going into next steps, make sure you can print something using the software SPC 10. A botched setup will likely make you miserable for quite some time. When necessary, change system locale to Japanese during installation.

9.2.1 Setup Tepra server

See [gem package – tepra], page 5. Print label with QR-code using Ruby gem named `tepra`. Install the gem as shown below.

```
CMD> gem source -a https://dream.misasa.okayama-u.ac.jp/rubygems/
```

```
CMD> gem install tepra
CMD> gem update tepra
```

Have a configuration file `~/.teprarc`. An example is shown below.

```
---
:printer: KING JIM SR3900P
:port: 8889
:timeout: 5
```

The program will be installed on somewhere appropriate such as `c:/Ruby27-x64/bin/`. Check where it is. When `tepra` is launched with option `print` and an argument `input.csv`, it creates barcode labels. Issue following commands for test.

```
CMD> which tepra
/c/Ruby27-x64/bin/tepra
CMD> tepra print "1234567890,Allende"
```

Most of the time, failure is because the program cannot find printer. This often happens when printer is connected on Wi-Fi instead of USB. Identify name of the printer on SPC 10 such as KING JIM SR5900P-NW. Put it to a configuration file `~/.teprarc`. A line should look like `:printer: KING JIM SR5900P-NW`.

Following commands allow you to print QR-code to King Jim's Tepra.

`tepra -help` [Command]
Show instruction for setup.

`tepra` [Command]
Launch server to accept query via REST interface.

`tepra print` [Command]
print QR-code with number and name.

The program `tepra` works as label-print server. When `tepra` is launched without argument, it runs as web server (`http://localhost:8889/`). Access `http://localhost:8889/` by Firefox and try to print label. To make this PC Tepra server for Android, give PC a static IP address. External URL will look like `http://192.168.11.103:8889/`. Print the URL with barcode and put the label onto PC (and the label printer), so that Android App can be configured quickly. Create a shortcut of `tepra.bat` and place it in `~/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/Startup/`.

9.2.2 Configure Android

See Section 5.2 [Labor Sisyphus for Android], page 13. The Android App `Sisyphus` prints label with QR-code via the Tepra server.

On the Android App, tap `Configuration` tab. Push `Label` subsection and set URL by scan of the QR code. Now printer icon on main tab will do the job.

9.2.3 Configure web browser

During record browsing of Medusa using web browser Firefox on MS-Windows, to click 'label' icon initiates downloading CSV file with extension `.label`. To print label from the web browser, the CSV file with extension `.label` should be deflected to `tepra print`. As of March 22 (2015), we setup web browser Firefox in following manner.

1. Create a file with extension `.BAT` that contains single line `tepra print %1`.
2. Convert the file with extension `.BAT` into a file with extension `.EXE` by third party's utility such as `Bat_To_Exe_Converter`.
3. Correlate file with extension `.label` and the `EXE` file on Firefox's Options.

To create a label from command line with stone-ID known, call `orochi-label`. See Section 5.4 [Record operator Orochi], page 15.

To duplicate a label using barcode reader, call `perl -S tepra-duplicate`. See Section 11.5 [Ruby/Perl/Shell script], page 43.

To relocate a stone using barcode reader, call `ruby -S stone`. See Section 11.5 [Ruby/Perl/Shell script], page 43.

9.3 Barcode printer (Petit lapin)

From both Sisyphus for android and WordPad for MS-Windows, one can have label of QR barcode by label printer SATO PT-208e.

9.3.1 Setup SATO PT-208e

The label printer is supposed to have static IP address. Configure the device and network. Fudo's SATO PT-208e has MAC address `00:80:92:4e:a6:ad` and `172.24.1.10` is assigned by odnet. It is recommended to set `'auto power-off time'` to 480 minutes instead of default 5 minutes. Use PT200e to configure the parameter. Port 1024 is used for configuration communication.

The label ribbon is `'item number 565035121, P20xW25 (E)'`.

9.3.2 Configure Android for Petit lapin

See Section 5.2 [Labor Sisyphus for Android], page 13.

To print to SATO PT-208e from the Android App, you need SmaPri Driver for Android, which is printer server that runs on the Android device. You can download demo version from SATO's web site (<http://www.satoworldwide.com/software/smapi-for-android.aspx>). On SmaPri Driver that runs on the Android device, type in IP address of PT-208e. Also, set printer disconnection timer to `'Every label print'` to allow connections from other Androids.

For example on SmaPri Driver for Android, specify `http://172.24.1.10:1024`. On Sisyphus for Android, specify `http://localhost:8080`.

9.3.3 Configure PC for Petit lapin

You can print QR code from WordPad under MS-Windows Vista. First install Printer driver for MS-Windows from SATO's web site (http://www.sato.co.jp/download/manual/printerdriver_utility/pt408e412e-pt200e208e208m208j.html). The printer name can be something like `'SATO PT208e'`.

Then configure the driver to have a paper setup for the label ribbon. Dimension of the paper is 20 mm length, 25 mm width, and 3 mm gap. The paper name can be something like `'Standard Label P20 x W25 mm'`. Also setup a barcode font to substitute letters by QR code. On `tab`, create a new by `'`. The font name can be something like `'SATO DREAM QR2'`. Select `to be QR 2`

On WordPad, choose printer `'SATO PT208e'` and the paper setup for 20 x 25 mm. With appropriate margin (left and top to 0.2 inch, right and bottom to 0.05 inch), list stone IDs delimited by new page `'^L'`. Hi-light all text, and set font and size to `'SATO DREAM QR2'` and **2 point**, then choose print from menu.

A file in rich text format that was successfully printed on February 12, 2016 is shown in Listing below.

```
CMD> type stray-code.rtf
{\rtf1\ansi\ansicpg932\deff0\deflang1033\deflangfe1041{\fonttbl{\f0\fswiss\fprq2\fchar
{*}\generator Msftedit 5.41.21.2509;}\viewkind4\uc1\pard\f0\fs12 DREAM\par
\fs10 20160213000000-000001\fs12\par
\lang1041\fs6 20160213000000-000001\page
```

```
\lang1033\f0\fs12 DREAM\par
\fs10 20160213000000-000002\fs12\par
\lang1041\f1\fs6 20160213000000-000002\page
\lang1033\f0\fs12 DREAM\par
\fs10 20160213000000-000003\fs12\par
\lang1041\f1\fs6 20160213000000-000003\page
}
CMD> write /pt stray-code.rtf "SATO PT208e"
CMD> ...
```

10 R-library Chelyabinsk

See Chapter 7 [Data format CASTEML], page 20. See Chapter 8 [Data format IONML], page 23. See [r package – chelyabinsk], page 5. See manual (<https://dream.misasa.okayama-u.ac.jp/documentation/chelyabinsk-manual.pdf>).

This R library ‘Chelyabinsk’ allows you process geochemical datasets, namely CASTEML and IONML.

Geochem format for R PMLAME

The core data structure used in this library is a ‘R:pmlame’, which is a polymorph of CASTEML (See Chapter 7 [Data format CASTEML], page 20). An example of ‘R:pmlame’ is expressed by a table with rows-and-columns as shown below. See Chapter 8 [Data format IONML], page 23.

	SiO2	Li	Sr
ref-gl-tahiti	0.59	2.08e-05	1.07e-04
ref-cpx-klb1	0.52	1.37e-06	3.61e-05

Application – mass balancing

Create three CSV files, which are elemental abundances of phases, elemental abundances of whole-rock (WR), and modal abundances and densities of phases and WR. Then issue the following command. These csv files should be in the structure of ‘tblame’. The densities of typical phases are summarized in <https://dream.misasa.okayama-u.ac.jp/?query=20160823144438-068846>.

```
$ cat insitu.csv
,H,H_error,Li,Li_error,Na,Na_error,P,P_error,S,S_error,Cl,Cl_error,K,K_error,Ca,Ca_err
ol,3.3568E-05,1.1189E-05,1.7000E-06,7.0000E-07,7.4186E-05,7.4186E-05,NA,NA,NA,NA,NA,NA
opx,3.8044E-05,5.5947E-06,1.0000E-07,1.0000E-07,7.4186E-05,7.4186E-05,NA,NA,NA,NA,NA,NA
di,5.5947E-05,2.2379E-05,3.0000E-07,4.0000E-07,3.7093E-03,5.1930E-03,NA,NA,NA,NA,NA,NA
ab,1.4546E-04,1.2308E-04,4.0000E-07,3.0000E-07,6.7602E-02,4.4687E-03,NA,NA,NA,NA,NA,NA
K-fel,NA,NA,2.0000E-06,1.0000E-06,1.8992E-02,1.0683E-02,NA,NA,NA,NA,NA,NA,9.8788E-02,2
apa,NA,NA,1.0000E-07,6.0000E-08,4.4319E-03,1.4991E-03,1.8366E-01,1.3317E-03,1.4100E-04
mer,NA,NA,1.1000E-07,9.0000E-08,2.2241E-02,1.0334E-03,2.0525E-01,1.5088E-03,7.8513E-05
chr,NA,NA,NA,NA,1.4837E-04,1.4837E-04,NA,NA,NA,NA,NA,NA,8.3015E-05,8.3015E-05,2.1441E-0
il,NA,NA,NA,NA,1.4837E-04,1.4837E-04,NA,NA,NA,NA,NA,NA,8.3015E-05,8.3015E-05,5.0028E-04
gl,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA
tro,NA,NA,NA,NA,NA,NA,NA,NA,3.5000E-01,3.0000E-03,NA,NA,NA,NA,NA,NA,6.4700E-01,2.0000E
tae,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,4.4480E-01,3.2000E-03,5.5520E-01,2
kam,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,9.4630E-01,7.0000E-04,5.3700E-02,1

$ cat wholerock.csv
,H,Li,Be,B,C,N,F,Na,Mg,Al,Cr,P,S,Cl,K,Ca,Ti,Si,Mn,Fe,Fe,Ni,Ni,Rb,Sr,Y,Zr,Nb,Cs,Ba,La,C
SRCmeas,1.0070E-04,1.7900E-06,NA,NA,4.0000E-04,NA,1.0000E-01,7.2338E-03,1.5558E-01,1.2

$ cat mode.csv
,mode,density
WR,1,3.57
ol,0.4596,3.5
opx,0.3143,3.6
di,0.0612,3.4
ab,0.0913,2.66
K-fel,0.0075,2.6
apa,0.0020,3.1
```

```
mer,0.0027,3.1
chr,0.0096,4.8
il,0.0010,4.8
gl,0,3
tro,0.0371,4.9
tae,0.0051,8.3
kam,0.0089,7.9
```

```
$ Rscript -e "chelyabinsk::cbk.plot.balance(\"insitu.csv\", \"wholerock.csv\", \"mode.csv\""
```

```
R> library(chelyabinsk)
R> pdf("Rplots.pdf")
R> cbk.plot.balance("insitu.csv", "wholerock.csv", "mode.csv")
R> dev.off()
```

```
$ echo Makefile
```

```
# -*- mode: makefile -*-
```

```
INSITU:=insitu
```

```
WR:=wholerock
```

```
MODE:=mode
```

```
PLOT:=Rplots.pdf
```

```
all: $(PLOT)
```

```
$(PLOT): $(INSITU).csv $(WR).csv $(MODE).csv Makefile
```

```
Rscript -e "chelyabinsk::cbk.plot.balance(\"$(INSITU).csv\", \"$(WR).csv\", \"$(MODE).csv\")"
open $(PLOT)
```

```
clean:
```

```
rm -f $(PLOT)
```

11 Okadai utilities

These commands allow you to operate analytical machine, image manipulation, and documentation.

11.1 Vstool

See [gem package – vstool], page 8. See Section 11.4 [Matlab script], page 41. `vs_attach_image.m` is the core program and others assist it.

`vs-attach-image` *option imagefile* [*imagefile2*,...] [Command]

Create imageometry and copy IMAGEFILE obtained under a live session, to VisualStage 2007. Create only imageometry with OPTION `--dry-run`. Image format of PNG, JPG, BMP, and TIF are supported.

To upload image with imageometry to web service by Medusa, have images with PNG or JPG format and call [Command `oroichi-upload`], page 17.

Image format supported are summarized in See `stone-operation-ja`.

[MatlabScript `vs_attach_image.m`], page 42.

Typical work flow to upload mosaic to VisualStage and Echo using `vs-attach-image` is shown below.

```
cmd> dir
20200214-1808/ 20200214-1808.jpg 20200214-1808.mapxml 20200214-1808.semstg 2020
cmd> vs-attach-image 20200214-1808.jpg
cmd> oroichi-upload 20200214-1808.jpg --stone_id=20200213095448-577600 --surface_id=2
...
cmd> cd 20200214-1808/
cmd> vs-attach-image --dry-run *.jpg
cmd> oroichi-upload *.jpg --surface_id=20200213115535-834552 --layer=BSE

cmd> dir
cnso-om-mcs4a@2763.jpg
cmd> vs-attach-image --dry-run cnso-om-mcs4a@2763.jpg
matlab>> spots
cmd> dir
cnso-om-mcs4a@2763.jpg cnso-om-mcs4a@2763.geo cnso-om-mcs4a@2763.tex cnso-om-mcs4
cmd> casteml-upload cnso-om-mcs4a@2763.pml
```

Restart a process `TileWorker` (<https://gitlab.misasa.okayama-u.ac.jp/yyachi/deploy-stone>) when necessary (obsolete).

`loop-vs-attach-image` [Command]

Automatically attach images obtained by SEM to VisualStage 2007. This command is background daemon that calls `vs-attach-image`.

`vs-add-file` [Command]

No description is available.

`vs-attach-image-1269` *imagefile* [Command]

Upload IMAGEFILE of view from the top on Cemece's sample holder to VisualStage 2007.

`vs-attach-image-1270` *imagefile* [Command]

Upload IMAGEFILE of view from the top on Cemece's sample holder to VisualStage 2007.

`vs-get-affine` [Command]

Return Affine matrix under current session of VisualStage 2007.

11.2 Image

These commands save your time on image manipulation.

spots0 [*options*] *imagefile* [RubyScript]

Generate a cosmetic L^AT_EX file from IMAGEFILE.

image-scalebar [*-grid=1 -width=25.4*] *imagefile* [Command]

Put scalebar on IMAGEFILE.

To draw a scale bar on an image, call this command. A PML^AT_EX file to show the image with scale bar is created. When PML^AT_EX file already exists, it does nothing.

When an imagefile is accompanied with imajeometry *.txt which is created by JEOL JSM-7001F, the command does not prompt you to type in magnification. The magnification is relative to 12 cm.

See [gem package – scalebar], page 7.

imageX [*options*] *imagefile* [*imagefile2...*] [PerlScript]

Create a L^AT_EX catalog file for IMAGEFILE.

spots-warp [*options*] *stagelist.txt* [Command]

Project coordinate described in STAGELIST.TXT to other space. See [gem package – multi_stage], page 7.

projection-map *stagelist.txt -affine-file stageometry.geo* [*options*] [Command]

Generate imageometry based on project coordinates described in STAGELIST.TXT and STAGEOMETRY.GEO exported from StageVIEW. See [gem package – multi_stage], page 7.

projection-device *imagefile -magnification M -stageposition X,Y,Z* [Command]
-scan-rotation R

Generate project coordinates from image files, magnification, center coordinates, and rotation. See [gem package – multi_stage], page 7.

image-warp [*options*] *imagefile* [Command]

Project IMAGEFILE into VS space based on Affine matrix *affine_xy2vs* stored in imageometry file and export sub-area of the VS space as image file. See [python package – image_mosaic], page 6.

image-get-affine [*options*] *imagefile* [Command]

Return *affine_xy2vs* (also *affine_ij2vs* and *anchors_xy*) estimated from *anchors* and *anchors_ij* in imageometry file. This command also reads *anchors_ij* via GUI and *anchors* via stdin. This command is subset of **image-warp**, without image manipulation. See [python package – image_mosaic], page 6.

warp_image [*options*] *imagefile* [Command]

Transform IMAGEFILE using Affine matrix *affine_ij2ij* and export image. Affine matrix can be specified by (1) 3x3 matrix, (2) center of rotation in original image, rotation angle, and magnification as similar to **haffine_from_params**, and (3) coordinates where the 4 corners of the original image are projected. The area to be exported can be specified by width and height via arguments. Without width and height specified, those of the original image would be applied. This program also imposes the original image on wall image. In this case, the area to be exported is set by width and height of the wall image. See [python package – image_mosaic], page 6.

blend-image *imagefile0 imagefile1 x y width height alpha beta* [Command]

Impose image to wall image with alpha blend techniques. Location to impose the image is set by ij coordinates via arguments. See [python package – image_mosaic], page 6.

11.3 Emacs script

See [emacs package – fudo], page 9.

Some of Fudo software can be invoked from GNU Emacs. Install emacs package -- fudo. Evaluate following symbolic expressions.

```
(add-to-list 'package-archives '("dream" . "https://dream.misasa.okayama-u.ac.jp/docume
(package-install 'fudo)
```

These commands allow you edit text efficiently.

fudo-eshell [Elisp]

Call `eshell` and suggest commands for `spotDoc` operation.

fudo-browse-spotDoc &optional spotDoc-URL [Elisp]

View `spotDoc` by web browser, sftp client, and Dired.

fudo-for-echo regexp [Elisp]

List stones in Echo with `regexp` matched. With selection of stone, you can insert reference to L^AT_EX document, plot related datasets, or call `oroichi` or `casteml`. You need the newest copy of database as local file in `~/ .emacs.d/auto-stone-list`.

fudo-barcode-scan rubyscript [Elisp]

Launch `rubyscript` that accepts input from barcode reader interactively. Make sure if the script runs on its own.

fudo-solar-abundance [Elisp]

Return solar elemental composition with atomic weight and isotope abundances.

fudo-isocircle isorg-file option [Elisp]

Create a LaTeX file with isocircles from ISORG-FILE. This command calls `casteml spots` and processes ISORG-FILE.

fudo-texify-casteml stones &optional texfile [Elisp]

Create TEXFILE of STONES or from `casteml`. This function calls external command `chelyabinsk::cbk.texify`. See Chapter 10 [R-library Chelyabinsk], page 35. When called interactively, this prompts to specify existing `casteml` file.

fudo-mv-1270-chain start end [Elisp]

Synthesize a shell script to rename files by chain analysis using `ims-1270`.

fudo-automosaic-demo list-area list-element [Elisp]

Demonstrate how to create a shell script that invokes `automosaic`. This command internally calls `fudo-automosaic`.

orgtbl-to-R TABLE PARAMS [Elisp]

Convert an orgtbl-mode TABLE to R statements to have multiple variables. Have an orgtbl in R script as shown below.

```
comment.datain <- "
#+TBLNAME: tblame0
#+ORGTBL: SEND datain:tblame0 orgtbl-to-R :no-escape t :dataframe t
| rownames | stone | phase | d180 | d170 | En | Fs | Wo |
|-----+-----+-----+-----+-----+-----+-----+-----|
| ol_chd1  | chd1  | ol    | 2    | 1    | 98   | 0    | 2    |
| ol_chd2  | chd1  | ol    | 3    | 2    | 90   | 10   | 0    |
| cpx_chd1 | chd1  | cpx   | 1    | 0.5  | 5    | 0    | 45   |
| cpx_chd2 | chd2  | cpx   | 5    | 3    | 2    | 0    | 48   |
```

```
| opx_chd3 | chd8 | opx | -1 | -0.5 | 99 | 0 | 1 |
"
```

```
## BEGIN RECEIVE ORGTBL datain:tblame0
## END RECEIVE ORGTBL datain:tblame0
```

`orgtbl-to-matlab` *TABLE PARAMS* [Elisp]
Convert an orgtbl-mode TABLE to Matlab statements to have multiple variables.

`org-table-export-csvfile` *TABLE OUTFILE* [Elisp]
Write an orgtbl-mode TABLE into csvfile. This is wrapper to `org-table-export`. See Section “Built-in table editor” in `org`.

`castbl-to-latex` *TABLE PARAMS* [Elisp]
Convert an orgtbl-mode TABLE referred as ‘castable’ to a table in L^AT_EX. At the same time export the TABLE to CASTEML using `casteml convert`. See Section “Data format CASTEML” in `toolkit`. Have an org-table in L^AT_EX file as shown below.

Some programs refer to major elements as oxide (e. g., FeO) instead of as metal (e.g., Fe). When an element stays in a sample in oxide form, it is safe to store concentration both in oxide and metal.

```
% BEGIN RECEIVE ORGTBL 2019-02/20190214-0851
% END RECEIVE ORGTBL 2019-02/20190214-0851
\iffalse
#+ORGTBL: SEND 2019-02/20190214-0851 castbl-to-latex :no-escape t
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| session | | | ISZ9 | | | ISZ10 |
| stone-ID | | | 20100310092602184.stokeshi | | 20100310092602096.stokeshi |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| SiO2 | cg/g | | 65.31 | | | 74.07 |
| FeO | cg/g | | 3.96 | | | 0.61 |
| Fe | cg/g | | | | | |
| P2O5 | cg/g | | 0.19 | | | 0.05 |
| P | cg/g | | | | | |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Li | ug/g | | | | | 65.8 |
| B | ug/g | | | | | |
| Pb | ug/g | | | | | 34.093 |
| U | ug/g | | | | | 3.585 |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| d180 | | | | | | |
| d180_error | | | | | | |
| Sr87zSr86 | | | | | | |
| Sr87zSr86_error | | | | | | |
| Pb206zPb204 | | | | | | |
| Pb206zPb204_error | | | | | | |
|-----+-----+-----+-----+-----+-----+-----+-----+-----+
\fi
```

When you toss the table in Emacs, a CSV file is generated and it will be further converted into a CASTEML file by `castbl-to-latex` with commands shown below.

```
CMD> casteml convert -f pml 20190214-0844.csv > 20190214-0844.pml
```

11.4 Matlab script

See [matlab script – VisualSpots], page 9.

These commands allow you visualize coordinates in image and read datasets from mass spectrometry.

`oxideweight.m` [MatlabScript]

Return molecular weight of oxides per cation.

```
>> oxideweight('FeO')
ans = 71.8464
>> doc oxideweight
```

`isotable.m` [MatlabScript]

Return atomic weight and isotope abundance.

```
>> isotable('Li7')
ans =
    6.940
>> [m R nicknames] = isotable('Li7')
m =
    6.9400
R =
    0.9241
nicknames =
Li
>> isotable('Si30')
ans =
    60.0842
```

`isotope_constants.m` [MatlabScript]

Return atomic weight and isotope abundance.

```
>> [mass, ratio] = isotope_constants;
>> mass.Li7
    7.0160
>> ratio.Li7
    0.9241
```

`isotope.m` [MatlabScript]

Define element mass, isotope mass and isotope abundances on workspace.

```
>> isotope
>> whos
  Name      Size      Bytes  Class
  H         1x1       3704  struct
  Li        1x1       3704  struct
  O         1x1       5184  struct
>> O.mass
ans =
  data: 15.9994
  unit: 'amu'
  label: 'atomic weight'
```

`atomic_ratio.m` [MatlabScript]

Read a CASTEML file and return molar number of atom relative to Si-28.

See online help by following command.

```
>> doc atomic_ratio
```

Procedure to estimate atomic ratios (F19/Si30, Cl35/Si30, and Cl37/Si30) of reference glass ref-gl-myk1-1 (<http://dream.misasa.okayama-u.ac.jp/?query=20150518154615-539156>) is shown below as an example.

```
$ cd ref_data/
$ casteml download 20150518154615-539156 > ref-gl-myk1-1.pml
>> atomic_ratio('ref-gl-myk1-1',{ '19F', '35Cl', '37Cl'}, 'ref', '30Si')
```

spots.m [MatlabScript]

Create relative-spot table and L^AT_EX file from image interactively.

This script creates L^AT_EX file of an image with spots information in it. It also creates a CASTEML file with spot coordinates embedded. With the CASTEML file, you can upload analyses with image and spots.

See online help by following command.

```
>> doc spots
```

vs_attach_image.m [MatlabScript]

Transform an image to VS space and upload the image to VisualStage 2007.

This Matlab script converts an image to fit into space of VisualStage 2007. You will click three anchors of image and move stage to the three corresponding location by VisualStage 2007. This creates the converted image, and imageometry (with Affine matrix). After image creation, this attaches the image to VisualStage 2007.

See online help by following command.

```
>> doc vs_attach_image
```

See [python package – vs2007], page 8, that is referred by this script. [vs-attach-image], page 37.

demo_calib.m [MatlabScript]

Estimate *ionic yield* from *atomic_ratio* and *ionic_ratio* of reference materials. This command obtain obtain *atomic_ratio* from local casteml files. If the command cannot not find the files, this automatically downloads them from Medusa. There are two scenes you want to estimate *ionic yield*, that are one with LA-ICP-QMS and one with ims-5F.

1. LA-ICP-QMS: The *ionic_ratio* is load from `spreadsheet/demo_calib.m`. Needless to say, you have to edit the script.
2. ims-5f: The *ionic_ratio* is load from from local CDF files. Revise `autoload/demo_calib.m` for minor optimization.

readionml.m [MatlabScript]

Read IONML and return raw signal of the whole period. See Section 8.2 [Read IONML], page 24.

See online help by following command.

```
>> doc readionml
```

read_casteml.m [MatlabScript]

Read elemental info from CASTEML file. To refer to element info, this will download CASTEML file.

11.5 Ruby/Perl/Shell script

See [ruby/perl/shell script – xtreaml], page 9.

These scripts allow you concentrate more on science.

tepra-duplicate [RubyScript]

See Chapter 9 [Barcode operation], page 26. Duplicate QR-code using barcode reader and label printer. It tosses letters obtained by barcode reader to external program `orochoi-label`. This is useful when you want to place stone's name and ID on notebook.

Install gem package – `rubyserial` by `gem install rubyserial` in advance.

See also online-help of the program.

```
ruby -S tepra-duplicate.rb --help
```

spots-vs2007to2022 [RubyScript]

Convert spots exported in VisualStage2007 for import in VisualStage2022. See also online-help of the program.

```
spots-vs2007to2022 vs2007-export.txt > SIF-vs-import.csv
```

stone [RubyScript]

See Chapter 9 [Barcode operation], page 26. Relocate a stone. This program is supposed to be used with barcode reader. See also online-help of the program.

```
ruby -S stone.rb --help
```

stones [RubyScript]

See Chapter 9 [Barcode operation], page 26. Do inventory count. This program is supposed to be used with barcode reader. See also online-help of the program.

```
ruby -S stones.rb --help
```

sutex *LaTeX-document* [RubyScript]

Issue series of programs to push the document to Echo and spotDoc. Following programs are called sequentially.

1. `pmlatex add`
2. `pmlatex scan`
3. `latexmk`
4. `latexmk -c`
5. `reducepdf`
6. `pmlatex update`
7. `pmlatex commit`
8. `icp commit`

Check if the document was uploaded correctly. See spotDoc (<http://itokawa.misasa.okayama-u.ac.jp/?page=documentations>).

icp *spotDoc-URL* [PerlScript]

Copy and restore a document to/from spotDoc.

```
(let ((default-directory (sesame-junk-dir)))
  (select-frame (make-frame-command)) (eshell)
  (insert "pwd") (eshell-send-input)
  (insert "# Hint: perl -S icp spotDoc-URL") (eshell-send-input)
  (sesame-browse-url "http://itokawa.misasa.okayama-u.ac.jp/?page=documentations"))
$ pwd
```

```
~/deleteme.d/2016-06/
$ perl -S icp http://itokawa.misasa.okayama-u.ac.jp/?page=documentations/Intern2016/
...
$ cd TS-Dhofar-018/; vi report.tex
```

- lsims** *path-do-analysis-dir* [PerlScript]
List directory under HOSTPATH on HOST.
- TimeBokan** *dir1 dir2* [PerlScript]
Copy local dir to remote dir in fashion of Apple's Time Machine.
- cips2cdf** *infile* [PerlScript]
Convert Cameca's datafile to NASA's CDF.
- vs-warp-spots-1270** [PerlScript]
Convert device's coordinate to world's.
- dust-imposer-tex** [PerlScript]
Draw ellipses on L^AT_EX picture environment.
- qrcode** [PerlScript]
Download QR-code image from a remote server.
- mergepdf** *infile ... infileN outfile* [ShellScript]
Merge pdf files using Ghostscript.
- reducepdf** *infile* [ShellScript]
Reduce size of a PDF file using Ghostscript.
- stone-diff** [ShellScript]
Compare Medusa record by record using *auto-stone-list*.
- pml** [ShellScript]
Export TSV in clipboard to CASTEML file on Desktop. This is for Mac and MS-Windows.
Export TSV-format dataset in clipboard to standard CASTEML file. Download template for Excel from <https://dream.misasa.okayama-u.ac.jp/documentation/templates/template.xlsx>, fill your data, copy the data region to clipboard, then launch *pml.bat*. This command scans data in clipboard and exports the dataset to ~/Desktop.
- OpenTemplateAndPrint** [AppleScript]
Print barcode label to P-Touch 2430PC by Brother.

12 Machine gazer

The machine gazer consists of server and client. See [rails project – machine_time], page 6, for server.

See [gem package – godigo], page 7, for client. These commands allow you to keep track of ‘machine’ status. Have a configuration file `~/.godigo` as shown below.

```
---
:uri_machine: https://database.misasa.okayama-u.ac.jp/machine/
:machine: MACHINE-1
```

`godigo-session start` [Command]
Start ‘machine’ on machine-server to log status

`godigo-session stop` [Command]
Stop ‘machine’ on machine-server to log status and issue `sync`

`godigo-session sync` [Command]
Synchronize local directory to remote directory specified in a configuration file `~/.godigo`

13 Command index

A

AppleScript OpenTemplateAndPrint..... 44

B

blend-image..... 38

C

casteml --help..... 20
 casteml convert..... 20
 casteml download..... 20
 casteml join..... 20
 casteml plot..... 20
 casteml split..... 20
 casteml spots..... 20
 casteml upload..... 20
 Command casteml sub-command --help..... 20

E

Elisp castbl-to-latex..... 40
 Elisp fudo-automosaic-demo..... 39
 Elisp fudo-barcode-scan..... 39
 Elisp fudo-browse-spotDoc..... 39
 Elisp fudo-eshell..... 39
 Elisp fudo-for-echo..... 39
 Elisp fudo-isocircle..... 39
 Elisp fudo-mv-1270-chain..... 39
 Elisp fudo-solar-abundance..... 39
 Elisp fudo-texify-casteml..... 39
 Elisp org-table-export-csvfile..... 40
 Elisp orgtbl-to-matlab..... 40

G

godigo-session start..... 45
 godigo-session stop..... 45
 godigo-session sync..... 45

I

image-get-affine..... 38
 image-scalebar..... 38
 image-warp..... 38

L

loop-vs-attach-image..... 37

M

MatlabScript atomic_ratio.m..... 41
 MatlabScript demo_calib.m..... 42
 MatlabScript isotable.m..... 41
 MatlabScript isotope.m..... 41
 MatlabScript isotope_constants.m..... 41
 MatlabScript oxideweight.m..... 41
 MatlabScript read_casteml.m..... 42
 MatlabScript readionml.m..... 24, 42
 MatlabScript spots.m..... 42
 MatlabScript vs_attach_image.m..... 42

O

orgtbl-to-R..... 39
 orocho-cd..... 16
 orocho-ditto..... 16
 orocho-download..... 16
 orocho-find..... 16
 orocho-help..... 15
 orocho-label..... 16
 orocho-ls..... 16
 orocho-mkstone..... 16
 orocho-mv..... 16
 orocho-name..... 16
 orocho-open..... 16
 orocho-place..... 16
 orocho-pwd..... 16
 orocho-refresh-tile..... 16
 orocho-rename..... 16
 orocho-rm..... 16
 orocho-stone-in-box..... 16
 orocho-uniq..... 17
 orocho-upload..... 17
 orocho-url..... 17

P

PerlScript cips2cdf..... 44
 PerlScript dust-imposer-tex..... 44
 PerlScript icp..... 43
 PerlScript imagex..... 38
 PerlScript ionml-laicpqms.plx..... 23
 PerlScript lsims..... 44
 PerlScript qrcode..... 44
 PerlScript TimeBokan..... 44
 PerlScript vs-warp-spots-1270..... 44
 pmlatex --help..... 17
 pmlatex add..... 17
 pmlatex commit..... 17
 pmlatex scan..... 17
 pmlatex update..... 17
 projection-device..... 38
 projection-map..... 38

R

Rfunc cbk:read.ionml	24
Rfunc ionml.convert.ioncsv	23
Rfunc ionml.read.laicpqms.....	24
Rfunc ionml.read.session.....	24
Rscript ionml-laicpqms.R.....	23
RubyScript ionml-exp.rb.....	23
RubyScript ionml-icp.rb.....	23
RubyScript ionml-ims1270.rb.....	23
RubyScript ionml-ims1280.rb.....	23
RubyScript ionml-ims5f.rb.....	23
RubyScript ionml-vg5400.rb	23
RubyScript spots-vs2007to2022.....	43
RubyScript spots0.....	38
RubyScript stone.....	43
RubyScript stones.....	43
RubyScript sutex.....	43
RubyScript tepra-duplicate	43

S

ShellScript mergepdf	44
ShellScript pml.....	44
ShellScript reducepdf	44
ShellScript stone-diff.....	44
spots-warp.....	38

T

tepra.....	32
tepra --help.....	32
tepra print.....	32

V

vs-add-file	37
vs-attach-image.....	37
vs-attach-image-1269	37
vs-attach-image-1270	37
vs-get-affine.....	37

W

warp_image.....	38
-----------------	----

14 Project index

E

emacs package – fudo 9

G

gem package – actsasmappable 3
 gem package – casteml 4
 gem package – godigo 7
 gem package – jxa1_api 7
 gem package – machine_time_client 8
 gem package – medusa_rest_client 4
 gem package – multi_stage 7
 gem package – opencvtool 7
 gem package – orochi-for-medusa 4
 gem package – pmlatex 4
 gem package – scalebar 7
 gem package – sisyphus-for-medusa 4
 gem package – tepra 5
 gem package – visual_stage 8
 gem package – vstool 8
 gem package – withrecursive 3

I

ID Query 10

J

Jenkins for Fudo 10

M

matlab script – VisualSpots 9

P

Project manager for Fudo development 10
 python package – image_mosaic 6
 python package – jxmap 9
 python package – vs2007 8, 9

R

r package – chelyabinsk 5
 r package – MedusaRClient 5
 r package – VisualAnalysis 5
 rails project – epma_archiver 6
 rails project – jxa 7
 rails project – machine_time 6
 rails project – medusa 3
 rake project – mosaic-sem 9
 raspberry package – DREAM-PI 5
 Repository for packages – local 3
 Repository for packages – public 3
 ruby/perl/shell script – xtreaml 9

T

titanium project – sisyphus 3

W

web app – EPMA archiver 10
 web app – EPMA archiver for JXA-8800 10
 web app – Medusa for demo 10
 web app – Medusa for Echo 10
 web app – Session for machines 10
 www – dream 6
 www – toolkit 6

Z

Zabbix for DREAM 10