

Contest Archive Format

From ICPC-Contest Control Standard

Jump to:[navigation](#), [search](#)

This page describes a draft archive format for a contest. It is developed in parallel with the [Contest API](#). It might very well end up being very similar to the CDP which has been suggested to use for this, but which has a slightly different purpose...

Contents

- [1 Introduction](#)
- [2 Structure](#)
 - [2.1 Metadata](#)
 - [2.1.1 Example](#)
 - [2.2 Config](#)
 - [2.3 Problems](#)
 - [2.4 Registration](#)
 - [2.5 Activity](#)
 - [2.6 Results](#)
 - [2.7 Events](#)
- [3 Contest API archive formats](#)
 - [3.1 Design principles](#)
 - [3.1.1 Examples of Multiple File References](#)
 - [3.2 Config](#)
 - [3.2.1 Example file listing](#)
 - [3.3 Registration](#)
 - [3.3.1 Example file listing](#)
 - [3.4 Activity](#)
 - [3.4.1 Example file listing](#)
 - [3.5 Results](#)
 - [3.5.1 Example file listing](#)
 - [3.6 Events](#)
 - [3.6.1 Example file listing](#)
- [4 Multiple Archives and Secondary Systems](#)
 - [4.1 Shadow CCS](#)
 - [4.1.1 Example file listing](#)
 - [4.2 Contest Data Server](#)
 - [4.2.1 Example file listing](#)

Introduction

There are several reasons that contest information will be stored on disk, including:

- As configuration used to initialize a CCS
- As an archive of what happened in a contest
- As an archive for replaying a contest, either for testing contest tools or for teams to compete against live data

- As a base for offline analysis

This standard lays out the relative location and format of each type of contest-related information when reading or writing to disk. The top level structure is inspired by the [Contest API](#) structure, and a dump of the output from that API is one way to store the data. That said, the archive format allows for any format whose documentation has been registered with the ICPC. The sections below lists all currently known (and thus accepted) formats.

Structure

The Contest Archive consists of a single directory (possibly ZIP compressed) with a metadata file (**archive.yaml**) and at most a single entry for each of the types of data (**config**, **problems**, **registration**, **activity**, **results**, **events**). An entry is either a directory with the same name as the type of data, or a file with same base name as the type of data and any file extension.

Metadata

Archive metadata is stored in a YAML file called **archive.yaml** with the following keys:

Key	Description
created-by	Name of system creating this archive
archive	ID of format used for content types not listed in archive.yaml
config	ID of format used for config
problems	ID of format used for problems
registration	ID of format used for registration
activity	ID of format used for activity
results	ID of format used for results
events	ID of format used for events

Only **created-by** is always required, but if any of content type is not listed then **archive** must be specified.

Example

```
# Contest Archive
---
created-by: Kattis
archive:    contest-api
problems:   kattis
events:     missing
```

Config

Configuration data for the contest.

The following formats may be used:

ID	Description	Specification
contest-api	Contest API dump	https://clics.ecs.baylor.edu/index.php/Contest_Archive_Format#Config_2

Problems

Problems used at the contest.

The following formats may be used:

ID	Description	Specification
kattis	Kattis problem package format	http://www.problemarchive.org/
icpc	ICPC problem package format (subset of the Kattis format)	https://clics.ecs.baylor.edu/index.php/Problem_format

Registration

Registration data for the contest.

The following formats may be used:

ID	Description	Specification
contest-api	Contest API dump	https://clics.ecs.baylor.edu/index.php/Contest_Archive_Format#Registration_2

Activity

Activity in the form of submissions, judgements, runs and clarifications from the contest.

The following formats may be used:

ID	Description	Specification
contest-api	Contest API dump	https://clics.ecs.baylor.edu/index.php/Contest_Archive_Format#Activity_2

Results

Final results from the contest.

The following formats may be used:

ID	Description	Specification
contest-api	Contest API dump	https://clics.ecs.baylor.edu/index.php/Contest_Archive_Format#Results_2

Events

List of events from the contest.

The following formats may be used:

ID	Description	Specification
contest-api	Contest API dump	https://clics.ecs.baylor.edu/index.php/Contest_Archive_Format#Events_2
missing	No events data	

Contest API archive formats

NB!: This section should not be in this document, and is only included here while WIP. It should either be its own document, or be added to the [Contest API](#) document. Every subsection defines the on disk format **contest-api** for a type of data.

Design principles

Endpoints are stored in a single `<endpoint>.json` file containing the full list of objects. This file is identical to the API call to `/<endpoint>`. If there are file references in the JSON file these are stored in a subfolder per object, using the object ID as the folder name and the element name of the file reference as the base name of the file.

```
<endpoint>.json
<endpoint>/<id>/<referenced files>
```

Empty directories should be omitted. i.e. if there are no files in `<endpoint>/<id>`, the folder should not exist.

The file extension for each file reference must match the mime type in the REST endpoint using the following mapping:

Mime type	File extension
image/png	.png
image/jpeg	.jpg
application/zip	.zip

When there are multiple file references with the same file extension (e.g. multiple sizes of a logo), a specifier is added between the base name and file extension. Use of the regular file name that does not have the specifier is optional in this case, but if used it must be the largest or most important file, e.g. the source image that the other/smaller images were generated from. For images, the specifier must be the string "`<width>x<height>`". For other file types use any appropriate specifier.

Examples of Multiple File References

```
<endpoint>/<id>/banner.800x100.png # Must be a 800 x 100 px PNG
<endpoint>/<id>/banner.80x10.png   # Must be a 80 x 10 px PNG
```

or:

```
<endpoint>/<id>/logo.png           # Is probably the source image for the
other logo versions. Must be larger than the others.
<endpoint>/<id>/logo.56x56.png     # Must be a 56 x 56 px PNG
<endpoint>/<id>/logo.160x160.png   # Must be a 160 x 160 px PNG
```

Config

A directory, **config**, containing:

- a JSON file (`contest.json`) for the `/contests/<id>` endpoint
- a JSON file (`judgement-types.json`) for the `/contests/<id>/judgement-types` endpoint
- a JSON file (`languages.json`) for the `/contests/<id>/languages` endpoint

- system.yaml defined in the [CCSR](#).
- the contest banner(s), from the banner element of /contests/<id>, if available.
- the contest logo(s), from the logo element of /contests/<id>, if available.

Example file listing

```
config/contest.json
config/judgement-types.json
config/languages.json
config/system.yaml
config/banner.800x100.png
config/logo.png
config/logo.512x512.png
```

Registration

A directory, **registration**, containing:

- a JSON file (groups.json) for the /groups endpoint, if available
- a JSON file (organizations.json) for the /organizatons endpoint, if available
- a directory for organizations containing:
 - a directory for each organisation using the organisation ID as directory name, containing:
 - a file for each file reference available in /organizatons/<id>
- a JSON file (teams.json) for the /teams endpoint
- a directory for teams containing:
 - a directory for each team using the team ID as directory name, containing:
 - a file for each file reference available in /teams/<id>
- a JSON file (team-members.json) for the /team-members endpoint, if available
- a directory for team-members containing:
 - a directory for each team member using the team member ID as directory name, containing:
 - a file for each file reference available in /team-members/<id>

Example file listing

```
registration/groups.json
registration/organizations.json
registration/organizations/<id>/logo.56x56.png
registration/organizations/<id>/logo.160x160.png
registration/organizations/<id>/logo.512x512.png
registration/teams.json
registration/teams/<id>/photo.png
registration/teams/<id>/photo.jpg
registration/teams/<id>/backup.zip
registration/team-members.json
registration/team-members/<id>/photo.jpg
```

Activity

A directory, **activity**, containing:

- a JSON file (submissions.json) for the /submissions endpoint
- a directory for submissions containing:

- a directory for each submission using the submission ID as directory name, containing:
 - a file for each endpoint available under /submissions/<id>
- a JSON file (judgements.json) for the /judgements endpoint for the primary system
- a JSON file (runs.json) for the /runs endpoint on the primary system
- a JSON file (clarifications.json) for the /clarifications endpoint

Example file listing

```
activity/submissions.json
activity/submissions/<id>/files.zip
activity/submissions/<id>/reaction.mp4
activity/judgements.json
activity/runs.json
activity/clarifications.json
```

Results

A directory, **results**, containing:

- A JSON file (awards.json) for the /awards endpoint, if available
- A JSON file (scoreboard.json) for the /scoreboard endpoint

Example file listing

```
results/awards.json
results/scoreboard.json
```

Events

A directory, **events**, containing:

- a JSON file (event-feed.json) for the /event-feed endpoint on the primary system

Example file listing

```
events/event-feed.json
```

Multiple Archives and Secondary Systems

In more complicated contest configurations there may be a secondary CCS that is used to shadow the primary CCS to verify judgements or results, or a Contest Data Server used to serve additional contest data. In these cases it is expected that each system creates its own archive **as if it was the primary system** and then after the contest an automated tool is used to validate that the results are consistent and merge (flatten) the archives into one master archive.

In order to avoid conflicts and reduce size, identical files from a secondary system are ignored, and more important files are copied into the master archive with an extra `.specifier` before the file extension. This allows post-contest tools that are unaware of the additional systems to read the files as specified in the previous sections, while tools that need to understand the differences in the secondary system(s) can easily check for the existence of these additional files.

Shadow CCS

In the case of a shadow CCS verifying the results of a primary CCS, the specifier `.shadow` must be used. If there are multiple shadows, the specifier must be `.shadow1`, `.shadow2`, etc.

By definition the submissions must match and the master archive will only have one copy. Before merging, any differences in results or awards must be verified or explained. The remaining files that will get merged into the master archive are:

- a JSON file (`judgements.shadow.json`) for the `/judgements` endpoint on shadow system(s)
- a JSON file (`runs.shadow.json`) for the `/runs` endpoint on shadow system(s)
- a JSON file (`event-feed.shadow.json`) for the `/event-feed` endpoint on a shadow system(s)

Example file listing

```
activity/judgements.shadow.json
activity/runs.shadow.json
events/event-feed.shadow.json
```

Contest Data Server

A contest data server may generate additional non-critical files into the contest archive, such as reactions. Since the CDS does not generate new activity, by definition the submissions, judgements, and runs must match the primary CCS. Results and awards must be verified to match before merging. This leaves the event-feed, which is merged using the `.cds` specifier.

Example file listing

```
events/event-feed.cds.json
```

Retrieved from "https://clics.ecs.baylor.edu/index.php?title=Contest_Archive_Format&oldid=3036"

Navigation menu

Views

- [Page](#)
- [Discussion](#)
- [View source](#)
- [History](#)
- [PDF Export](#)

Personal tools

- [Log in](#)

Navigation

- [Main page](#)

- [Recent changes](#)
- [Random page](#)
- [Help](#)

Search

Tools

- [What links here](#)
- [Related changes](#)
- [Special pages](#)
- [Permanent link](#)
- [Page information](#)



- This page was last edited on 28 January 2019, at 20:47.
- Content is available under [Creative Commons Attribution-ShareAlike](#) unless otherwise noted.
- [Privacy policy](#)
- [About ICPC-Contest Control Standard](#)
- [Disclaimers](#)