IN13B-2162: **Introducing Das3** *Mission Support from the Bench to Orbit*

C. W. Piker[1], K. L. Steele[1], C. A. Gonzalez[2], L. J. Granroth[1], D. M. Miles[1] 1. The University of Iowa, 2. Southwest Research Institute

SUMMARY

A new version of the data analysis software authored at the University of Iowa has been providing telemetry storage and conversion for **TRACERS** mission development since early 2022. The system is a successor to das2, and is thus named ***das3.***

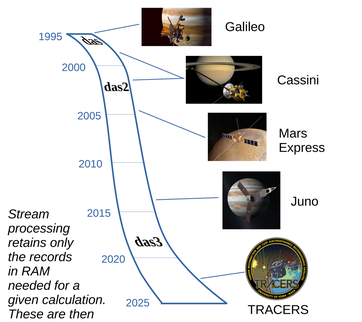
Though TRACERS development is nearing completion, das3 will continue to provide quicklook visualization and statically calibrated science data products throughout the operational phase. With it's focus on stream processing, and no-code raw packet parsing, das3 is ideal for early mission development phases when real-time or near real time processing is crucial for hardware validation. Since das3 is able to convert raw instrument packets into statically calibrated ISTP CDF files in geophysical coordinates, it will continue to provide mission infrastructure.

In addition to supporting the TRACERS mission, das3 utilities have been used for rapid conversion of PDS3 **Juno/Waves** science into PDS4 archive files during the second extended mission.

Here we introduce major components of das3. A companion poster in session IN13B, *"DASOC: A Display Application for the Science Operations"* details a new das3 client built upon modern web technologies.

Das3 development is not complete. Release candidate one is planned for summer of 2025. Das3 is open source software. Development is coordinated at github.com/das-developers

Stream processing over the years



Stream processing retains only the records in RAM needed for a given calculation. These are then flushed as soon as the calculation completes. Using a common stream format allows multiple independent components to be strung together into POSIX command pipelines. U. Iowa's stream stream format has changed slowly over the years.

dasCatalog Finding and defining data sources

- Each file in the dasCatalog is a JSON object
- Replaces das2 dataset lists and dataset descriptor responses
- There are two basic types, **Catalog** and **Source**
- A **Catalog** file provides a listing of sub items and set of absolute URLs at which each sub item is located
- A **Source** file describes a single data stream end-point and provides a set of absolute URLs at which it is located. Source files also define:
  - The **User Interface** controls that should be provided
  - The **Protocol**, including HTTP GET params, for speaking to the server
  - Available **output MIME** types and sub-type formatting options
  - Named **example** control settings that will provide data
- dasFlex servers automatically build query web-forms from Catalog files

Catalog Node ID

Low-level Server HTTP Parameters

How to subset in coordinates

How to select output formats

```
{
  "label": "BDC-ROI",
  "title": "MAG FM-1: Magnetic Field, 128 Hz Level 1 (x222, x22f, x232)",
  "type": "HttpStreamSrc",
  "version": "0.7",

  "protocol": {
    "method": "GET",
    "conversation": "HTTP/1.1",
    "baseURLs": [
      "https://physics.uiowa.edu/stream/source/preflight/l1/mag/fm-1/bdc_roi/flex"
    ],
    "authorization": {"required": false },
    "httpParams": {
      "read.time.min":  {"required": true, "type": "isotime" },
      "read.time.max":  {"required": true, "type": "isotime" },
      "bin.time.max":   {"required": false,"type": "real", "units": "s" },
      "read.sqid":      {"type": "enum", "required": false, "enum": {"x222","x22f","x232"}},
      "format.type":    {"required": false,"type": "enum","enum": ["das","cdf"]},
      "format.cdf.nocomp":{"required": false, "type": "boolean"},
    }
  },

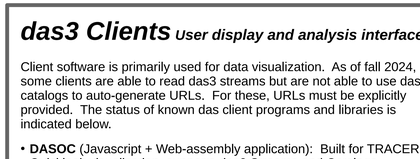
  "interface": {
    "coords": {
      "time": {
        "label": "Time",
        "presets": {
          "min": {
            "label": "Minimum","value": null, "title": "Minimum time value to stream",
            "set": {"param": "read.time.min","required": true}
          },
          "max": {
            "label": "Maximum","value": null,"title": "Maximum time value to stream",
            "set": {"param": "read.time.max","required": true}
          },
          "units": {"value": "UTC"},
          "res": {
            "label": "Resolution","value": null,"units": "s",
            "title": "The maximum time bin width for bin-reduced data in seconds",
            "set": {"param": "bin.time.max", "required": false}
          }
        },
        "validRange": ["2023-08-18T21:15", "2024-10-11T16:09"]
      }
    },

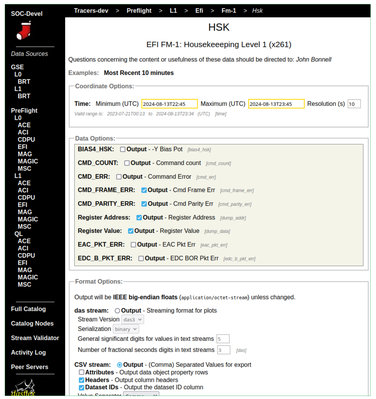
    "formats": {
      "cdf": {
        "label":"CDF file","title":"NASA Common Data Format","mimeTypes": ["application/x-cdf"],
        "prams": {
          "enabled": {
            "type": "boolean","cerGroup": "format", "value": false,
            "set": {"value": true,"param": "format.type","pval": "cdf"}
          },
          "nocompress": {
            "value": false, "type": "boolean","label": "No-Compression",
            "title": "Do not compress variables, needed for PDS compliant output.",
            "set": { "value": true, "param": "format.cdf.nocomp","pval": "1"}
          }
        }
      },
      "das": {
        "label":"das_stream","title": "Streaming format for plots",
```

Example TRACERS/MAG data source definition

dasFlex Web Service providing data over HTTP

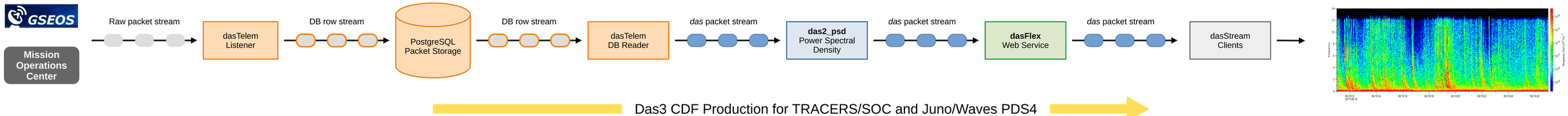
- Is a backwards compatible to das2py-server
- Server core is format agnostic, each installed component indicates supported MIME types
- It's supplied with das2 and das3 components, but others can be installed.
- Can convert das2 or das3 pipeline output **CSV** or **CDF** via dasStream utilities.
- Reads HTTP GET parameters and solves for a command pipeline, then runs it to *continuously* provide data until the command terminates.





Data query form auto-generated from a TRACERS/EFI catalog entry

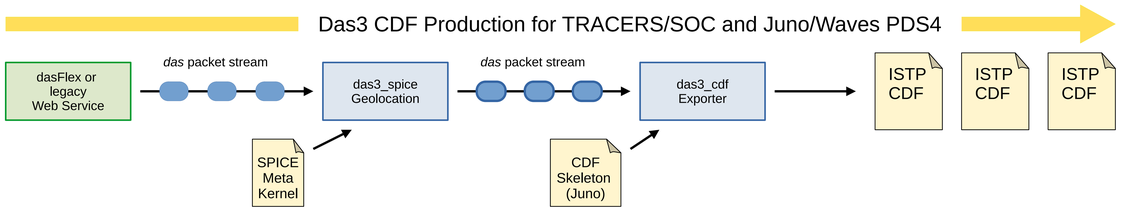
Das3 On-Demand Processing for TRACERS
real-time capable with DASOC



Any source of TRACERS CCSDS packets can be used to supply dasTelem, which then feeds the rest of the system. During instrument level testing data are often provided via GSEOS streams. On-orbit data will be provided via sFTP transfer from the Mission Operations Center (MOC) at Millennium Space Systems.

TRACERS test-set output in Autoplot.
Since dasFlex provides both das3 catalogs and das2 DSDF interfaces, existing client programs are supported

Das3 CDF Production for TRACERS/SOC and Juno/Waves PDS4

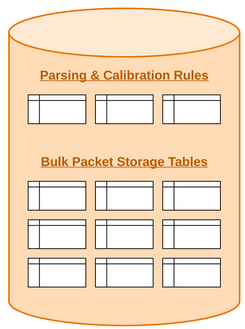


dasTelem Stores, Parses and Calibrates raw instrument packets into dasStreams

Das servers have traditionally relied upon bespoke data readers to produce the initial intrinsic resolution streams, and these are still supported. For TRACERS, a new system has been created, dasTelem which uses parsing rules stored in PostgreSQL tables to crack binary instrument packets into engineering values, and then calibrates these to physical units in defined coordinate frames.

For TRACERS, all Mission Level 0 telemetry packets are handled by dasTelem, which provides them as on-demand as das2 and das3 streams for Quicklook plotting and low level CDF file creation. Packets can also be provided as-is, without conversion or calibration.

- Provides bulk telemetry storage and data readers for ***dasFlex***
- Removes the need to write custom data reader programs for early mission data.
- Written in D and PL/pgSQL. Relies on PostgreSQL for packet storage, parsing configuration and calibration data.
- Provides an interface for custom packet decompression plugins.
- Real-time events handled via PostgreSQL notifications
- Parsing and calibration schemas are independent. Parsing tables can be wiped and recreated **without** affecting bulk packet tables.
- Not yet optimized. Initial performance is 2.3 to 16.2 MB/sec, depending on packet complexity.



PostgreSQL Packet Telemetry Database

dasStream Core Format

Interchangeable stream processors require a common communication format. DasStream **Version 3** allows the full *das2C* data model to be serialized *and validated*. High-rank arrays are supported along with and text and binary serializations Headers are are defined via XSD schemas. Stream processors commonly in use include:

- **das2_psd** converts waveforms to power spectral density at selectable with and overlap
- **das2_bin_avg** reduces stream volume, replaces intrinsic resolution data with time averages
- **das2_hapi** flow-through conversion of das streams to Heliophysics API streams
- **das3_spice** injects Geolocation and performs frame transforms via NAIF SPICE kernels
- **das3_csv** flow-through conversion of das streams to Comma Separated Values streams
- **das3_cdf** writes das streams to CDF files. Optionally uses CDF skeletons.

Not all das2 stream processors currently support version 3 streams. Updates are in progress in conjunction with mission needs.

Stream Header

Coordinate Definitions

Data Definition

Data Packets

```
|Sx|!50|
<stream type="das-basic-stream" version="3.0">
  <properties>
    <p name="instrument_host">MEX</p>
    <p name="instrument">MARSIS</p>
  </properties>
</stream>
|Hx|!3455|
<dataset name="ais" rank="3" index="*;160;80" plot="cartesian3d" >

  <coord physDim="time" axis="x">
    <properties><p name="label">SCET</p></properties>
    <scalar use="center" semantic="datetime" index="*;-;-" units="UTC">
      <packet numItems="1" itemBytes="24" encoding="utf8" />
    </scalar>
  </coord>

  <coord physDim="altitude" axis="y">
    <properties><p name="label">Alt/2, Apparent altitude</p></properties>
    <scalar use="reference" semantic="real" storage="float" index="*;-;-" units="km">
      <packet numItems="1" itemBytes="12" encoding="utf8" />
    </scalar>
    <scalar use="offset" semantic="real" storage="float" index="-;-;80" units="km">
      <sequence minVals="0.0000e+00" interval="-1.3705e+01" />
    </scalar>
  </coord>

  <coord physDim="frequency" axis="z">
    <properties><p name="label">Frequency (MHz)</p></properties>
    <scalar use="center" semantic="real" storage="float" index="-;160;-" units="MHz">
      <values>
        1.0340e-01 1.2850e-01 1.3120e-01 1.4230e-01 1.5300e-01 1.7520e-01 1.8590e-01
        2.0760e-01 2.1880e-01 2.2990e-01 2.4050e-01 2.5170e-01 2.7340e-01 2.8460e-01
        3.0630e-01 3.1700e-01 3.2810e-01 3.3920e-01 3.4990e-01 3.6100e-01 3.7170e-01
        ...
      </values>
    </scalar>
  </coord>

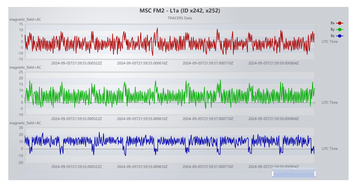
  <data physDim="E_spec_dens" name="sounder_return">
    <properties>
      <p name="label">Spectral Density (V1a2!nm!a-2!nHz!a-1!n)</p>
    </properties>
    <scalar use="center" semantic="real" index="*;160;80" units="V**2 m**-2 Hz**-1" >
      <packet numItems="12800" itemBytes="12" encoding="utf8" fill="-1.0e+31"/>
    </scalar>
  </data>
</dataset>
|Pd|1|153636|2012-12-21T15:06:40.596   1.5404e+03  1.7882e-14  1.1964e-14  4.3845e-15  1.9897e-15
5.8294e-16  5.5629e-15  3.6921e-16  5.1378e-16  5.3454e-15  2.7233e-15  1.1994e-15  8.0604e-16
4.7716e-15  3.9284e-15  2.8748e-15  2.0690e-14  2.5603e-16  2.7896e-14  4.0383e-14  4.3642e-14
4.0472e-15  1.7855e-16  1.2781e-15  1.8675e-15
```

Example Mars Expresss das3 Stream, UTF-8 serialization variant

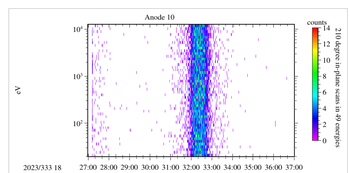
das3 Clients User display and analysis interfaces

Client software is primarily used for data visualization. As of fall 2024, some clients are able to read das3 streams but are not able to use das3 catalogs to auto-generate URLs. For these, URLs must be explicitly provided. The status of known das client programs and libraries is indicated below.

- **DASOC** (Javascript + Web-assembly application): Built for TRACERS Quicklook visualization, supports das3 Streams and Catalogs.
- **SDDAS** (C++ Application): SwRI has added das3 stream support to their visualization tools. Catalog support is in progress.
- **das2C** (C library and utilities): Supports das v2 and v3 Streams. Adding the catalog interface is in progress.
- **das2py** (Python library and stream validator): Supports the das2 API as well as das3 Streams and Catalogs.
- **das2dlm** (IDL library for SPEDAS): Supports das v2 and v3 streams, but not automatic URL generation via das3 Catalogs.
- **das2java** (Autoplot & MIDL core library): Does not yet support das3. Upgrades are planned for late spring 2025.
- **das2mat** (MATLAB library): *Future work*. Development is planned for fall 2025 or earlier.



Real-time streaming TRACERS/MAG to a browser with DASOC



SDDAS display of TRACERS/ACI das3 streams