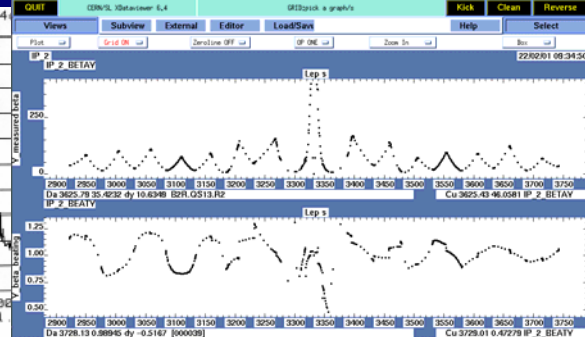
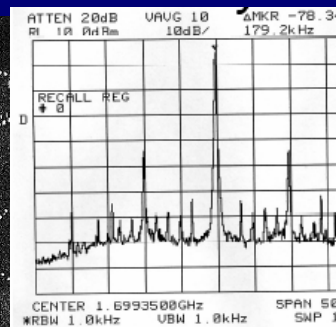
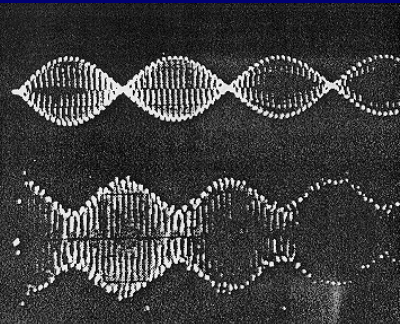
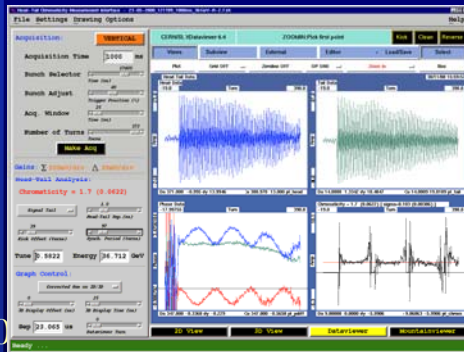
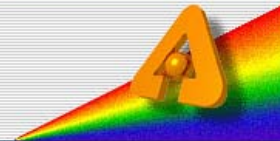


MEASUREMENTS

- All measurements to be recorded together with measurement parameters
- Standard facilities for display, browsing and analysis
- Archiving, references etc.
- Access for post-mortem, post-run analysis, web access etc.
- Standard data format
- Interface to analysis tools





SDDS and SDDS-Compliant Programs: A Modular System for Accelerator Design, Simulation, Control, and Analysis

Operations Analysis Group

Accelerator Operations Division—Argonne National Laboratory

[Introduction](#)
[System Analysis and Control](#)
[Presentations](#)
[Papers](#)

[Documentation](#)
[Accelerator Simulation](#)
[Obtaining Software](#)

Introduction

At the [Advanced Photon Source](#), we use a unique framework for accelerator simulation, control, and analysis. Our method is based on the use of a common self-describing file protocol and a toolkit of commandline programs that manipulate these files. In addition, we use [Tcl/Tk](#) scripts to orchestrate the tools and create graphical user interfaces.

The heart of this method is SDDS, which stands for "Self-Describing Data Sets." It is the name of our self-describing file protocol as well as the name of the toolkit we have created to manipulate these files. SDDS software offers capabilities comparable to commercial analysis packages, but it is free, unlicensed, and open source. We support several platforms and languages, including C/C++, Fortran, Java, Tcl/Tk, and Python. As a result, SDDS is in use at several accelerator laboratories around the world.

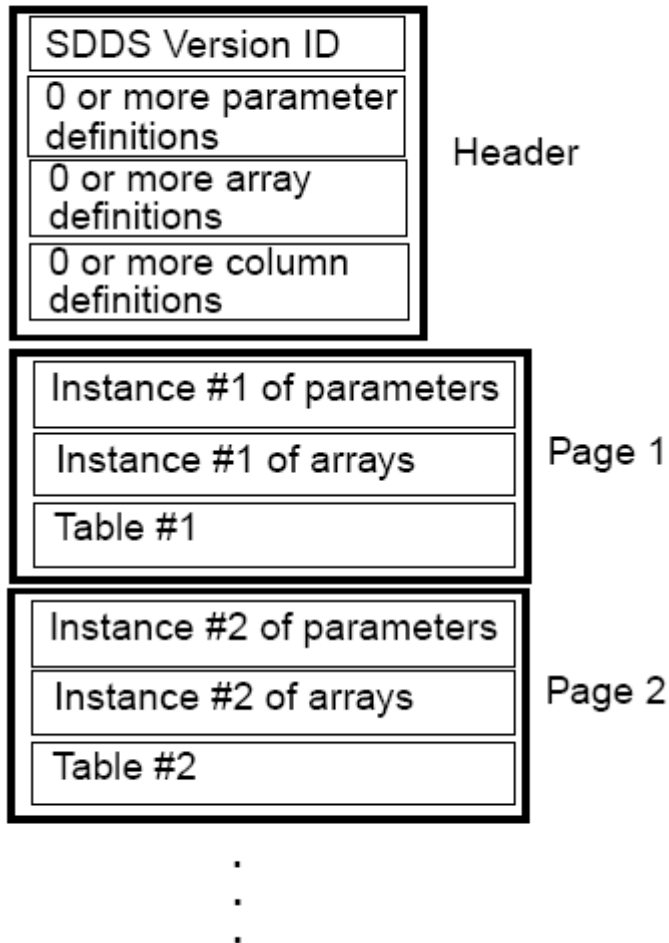
What is Self-Describing Data?

- **Self-describing (SD) data is identified and accessed by name only.**
- **SD data files include meta-data about data,**
 - e.g., units and data type.
- **Advantages:**
 - genuinely generic programs possible
 - data elements may be added to files without “breaking” existing programs
 - data tends to be self-documenting
 - source of data (measurement, simulation) is irrelevant

What is SDDS?

- **SDDS stands for “Self Describing Data Sets.”**
- **SDDS is just a standardized way to store and access data, i.e., a “file protocol.”**
- **SDDS also refers to a group of ~85 programs that use this file protocol.**
- **These programs are the “tools” in the SDDS Toolkit.**

SDDS Data Model



An SDDS file consists of a file header describing a structure composed of an arbitrary number of parameters and arrays, and a data table of arbitrary rows and columns.

Toolkit

- **Data display**

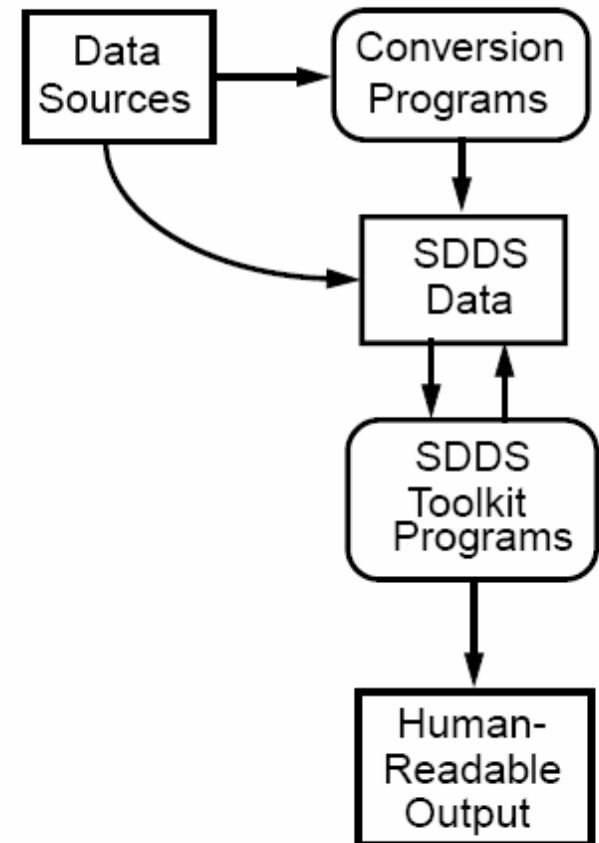
- plotting (2 programs)
- printing data as formatted text
- summarizing data set contents

- **Data processing**

- equation evaluation
- data filtering and outlier removal
- statistics, histograms, and correlations
- fitting and smoothing
- matrix operations (e.g., SVD)
- cross-referencing, sorting, and collation
- FFTs and digital filtering

Not actually using this stuff

SDDS Toolkit Paradigm



Some Commissioning Activities Performed Using SDDS and Scripts

- orbit/trajectory response matrices
- orbit correction algorithm development* (SR)
- dispersion and chromaticity* (SR, PAR)
- automated first-turn steering* (SR)
- beta-functions* (SR)
- dynamic aperture* (SR)
- tune shift with amplitude (SR)
- phase-space tracking (SR)
- tune shift with current (SR)
- BPM intensity dependence* (SR)
- integer tune (SR, booster)
- energy aperture for stored beam* (SR)
- physical aperture search* (SR)
- insertion device effect on orbit and tune* (SR)
- x-ray BPM “pollution” tests (SR)
- septum leakage field using beam (SR, PAR)

- lifetime vs scraper position, bump height* (SR)
- beta-function and dispersion correction* (SR)
- linear coupling reduction (SR)
- kicker bump matching (SR)
- BPM-to-quad offset measurements* (SR)
- search for sources of beam motion (SR)
- power supply ramp correction* (booster)
- automated injection steering (booster)
- longitudinal injection acceptance (PAR)
- bunch length and damping time (PAR)
- rf voltage calibration using beam (PAR)
- kicker waveform shape using beam (PAR)
- automated beam-excited HOM search (PAR)
- energy gain vs input power, frequency, and temperature (linac)
- klystron gain measurements (linac)
- search for cause of energy oscillations (linac)
- emittance measurements and beta-function matching* (linac)

All Classes

Packages

[cern.acccsoft.sdds.core](#)
[cern.acccsoft.sdds.core.format](#)
[cern.acccsoft.sdds.core.support](#)

All Classes

[AbstractConverter](#)
[CSVConverter](#)
[Converter](#)
[ConverterApp](#)
[Metadata](#)
[MultiDimArray](#)
[NumberTokenizer](#)
[SDDSAcqFile](#)
[SDDSArray](#)
[SDDSColumn](#)
[SDDSCConverter](#)
[SDDSDecoder](#)
[SDDSDirectory](#)
[SDDSElement](#)
[SDDSFile](#)
[SDDSFileAppender](#)
[SDDSIException](#)
[SDDSPParameter](#)
[SDDSParseException](#)
[SDDSToCSVConverter](#)
[SDDSToTSVConverter](#)
[SDDSTypes](#)
[SDDSUtills](#)
[Time](#)
[TxiImageToSDDSCConverter](#)

Overview Package **Class** Use Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

cern.acccsoft.sdds.core

Class SDDSArray

java.lang.Object

└─ [cern.acccsoft.sdds.core.SDDSElement](#)

└─ **cern.acccsoft.sdds.core.SDDSArray**

public class **SDDSArray**

extends [SDDSElement](#)

The class represents an array stored in a SDDS file.

Note that after array is created - its type cannot be changed e.g. if you create an array of type SDDSTypes.DOUBLE you can use only [setValues\(double\[\]\)](#) and [#setValues\(double\[\], int\[\]\)](#).

Version:

\$Id: SDDSArray.java,v 1.7 2005/10/21 09:20:37 gkruk Exp \$

Author:

Grzegorz Kruk

Field Summary

Fields inherited from class [cern.acccsoft.sdds.core.SDDSElement](#)

[ARRAY TAG](#), [COLUMN TAG](#), [DESCRIPTION](#), [DIMENSIONS](#), [END TAG](#), [FORMAT STRING](#), [NAME](#), [PARAMETER TAG](#), [SYMBOL](#), [TYPE](#), [UNITS](#)

Constructor Summary

[SDDSArray](#)(java.lang.String name, int type)

Creates a new SDDSArray.

Greg Kruk

Shot by Shot

SDDS Browser

File Options

SDDS root directory: Z:\2004_11_07

Choose directory...

Parameters

BTVI_Ti8.81204/getImage			
	Time	Cycle Sta...	Selector
BCTFI_TT40/BCTFI.Shared.Actions.acquisit	10:02:50...	28299	
BLMI_Ti8/BLMI.ClassGlobalCommactionLis	10:03:19...	28300	
BPMI_Ti8/BPMI.Shared.Actions.dabCrateA	10:04:08...	28301	
BPMI_Ti8/DWN/BPMI.Shared.Actions.crateE	10:04:17...	28302	
BPMI_Ti8/UP/BPMI.Shared.Actions.crateBu	10:04:46...	28303	
BTVI_LSS4.41831/getImage	10:05:14...	28304	
BTVI_LSS4.41831/getProfiles	10:05:43...	28305	
BTVI_LSS4.41895/getImage	10:06:12...	28306	
BTVI_LSS4.41895/getProfiles	10:06:41...	28307	
BTVI_Ti8.81204/getImage	10:07:10...	28308	
BTVI_Ti8.81204/getProfiles	10:07:38...	28309	
BTVI_Ti8.81306/getImage	10:08:07...	28310	
BTVI_Ti8.81306/getProfiles	10:08:36...	28311	
BTVI_Ti8.81306/getProfiles	10:09:05...	28312	
BTVI_Ti8.84304/getImage	10:09:34...	28313	
BTVI_Ti8.84304/getProfiles	10:10:02...	28314	
BTVI_Ti8.84404/getImage	10:10:31...	28315	
BTVI_Ti8.84404/getProfiles	10:11:00...	28316	
BTVI_Ti8.84404/getProfiles	10:11:29...	28317	
BTVI_Ti8.84604/getImage	10:11:58...	28318	
BTVI_Ti8.84604/getProfiles	10:12:26...	28319	
BTVI_Ti8.87437/getImage	10:12:55...	28320	
BTVI_Ti8.87437/getProfiles	10:13:24...	28321	
BTVI_Ti8.87604/getImage	10:13:53...	28322	
BTVI_Ti8.87604/getProfiles	10:14:22...	28323	
BTVI_Ti8.87750/getImage	10:14:50...	28324	
BTVI_Ti8.87750/getProfiles	10:15:19...	28325	
BTVI_TT40.400105/getImage	10:15:48...	28326	
BTVI_TT40.400105/getProfiles	10:16:17...	28327	
BTVI_TT40.400222/getImage	10:16:46...	28328	
BTVI_TT40.400222/getProfiles	10:17:14...	28329	
BTVI_TT40.400222/getProfiles	10:17:43...	28330	
	10:18:12	28331	

SDDS Default View

BTVI_Ti8.81204 @ Cycle sdds.10_17_43_000

Update 10:17:43 0

Name	Type and Value	Axis
descriptor	(String[:2]) -> BTVI, Ti8.81204, SPS.USE...	
image	(short[:]:100536) -> 47, 39, 41, 37, 48, 4...	Z
positionX	(float[:]:354) -> 23.312801, 23.184, 23.05...	X
positionY	(float[:]:284) -> -19.3766, -19.237202, -1...	Y
acqError	(String[:1]) -> ""	
acqType	(long[:1]) -> 0	
camera	(short[:1]) -> 1	
cameraOffset	(short[:1]) -> 0	
cameraType	(short[:1]) -> 0	

Active keys : [X] -> x axis, [Y] -> y axis, [Z] -> z axis (image), [D] -> display line, [H] -> display histogram, [SPACE] -> clear, [T] -> time/numbers on x axis

Data for Cycle: -

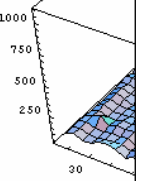
Point # 72717 X 10.81920051574707 Y -17.56439971923828 Z 52.0

Generic Browser

Mathematica/Java

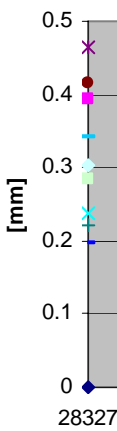
- `sddsdir = "Z:\\SDDS_LOGGING\\2004_11_07\\" <> BTVName <> "@getImage"`
- `SDDSDir = JavaNew["cern.accsoft.sdds.core.support.SDDSDirectory",sddsdir]`
- `si1 = SDDSDir@getFile[fileName];`
- `image = si1 @getArray["image"]@toArray[];`
- `xlist =si1 @getArray["positionX"]@toArray[];`
- `ylist = si1 @getArray["positionY"]@toArray[];`
- `sizeX = si1 @getParameterValue["sizeX"];`
- `sizeY = si1 @getParameterValue["sizeY"];`

Process 'n' loop

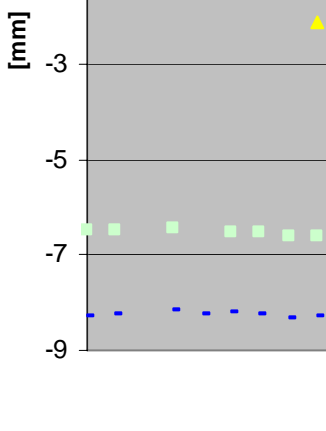
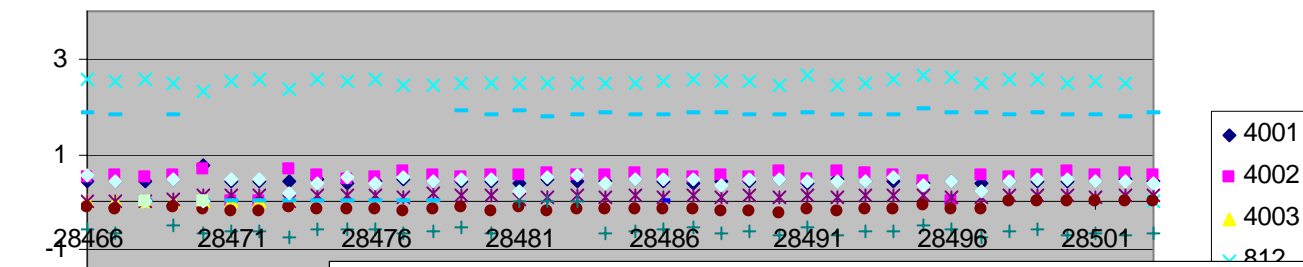


calculated entry point beta functions

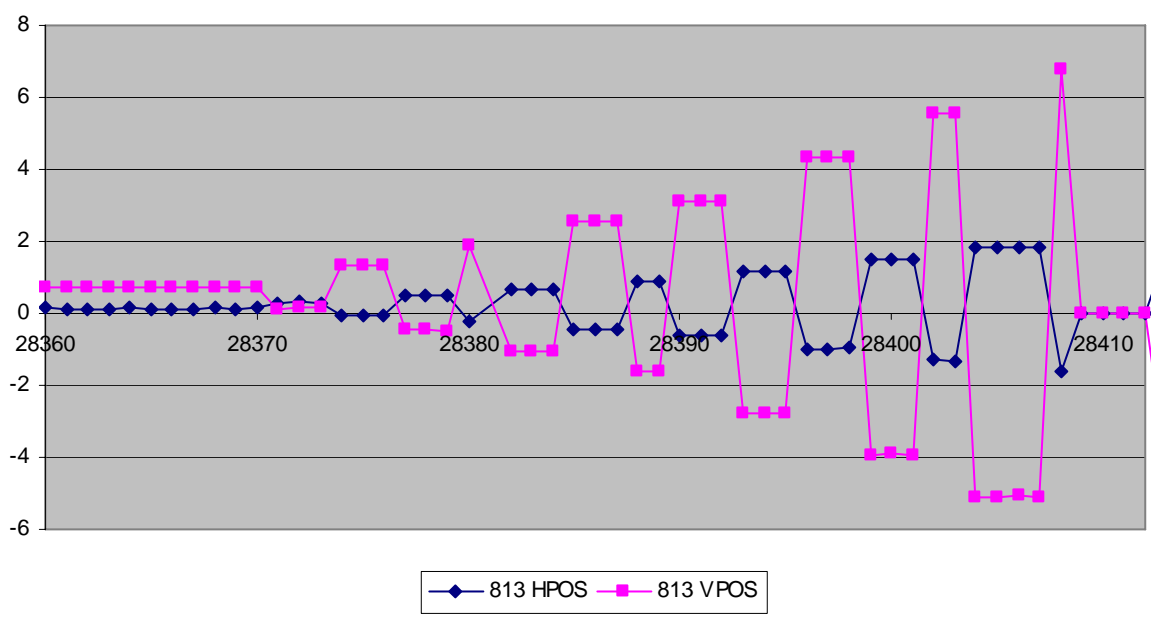
140



Horizontal Position



Position shift during dispersion meas.



XPOC

SDDS Browser

File Options

SDDS root directory: Z:\lbdstall_mkb\2007_07_01_full Choose directory...

Parameters

BTVDD_TD62/getImage

Time	Cycle Sta...	Selector
00:00:00.0...	10155	
00:00:00.0...	10156	
00:00:00.0...	10157	
00:00:00.0...	10158	
00:00:00.0...	10159	
00:00:00.0...	10160	
00:00:00.0...	10161	
00:00:00.0...	10162	
00:00:00.0...	10163	
00:00:00.0...	10164	
00:00:00.0...	10165	
00:00:00.0...	10166	
00:00:00.0...	10167	
00:00:00.0...	10168	
00:00:00.0...	10169	
00:00:00.0...	10170	
00:00:00.0...	10171	
00:00:00.0...	10172	
00:00:00.0...	10173	
00:00:00.0...	10174	
00:00:00.0...	10175	
00:00:00.0...	10176	

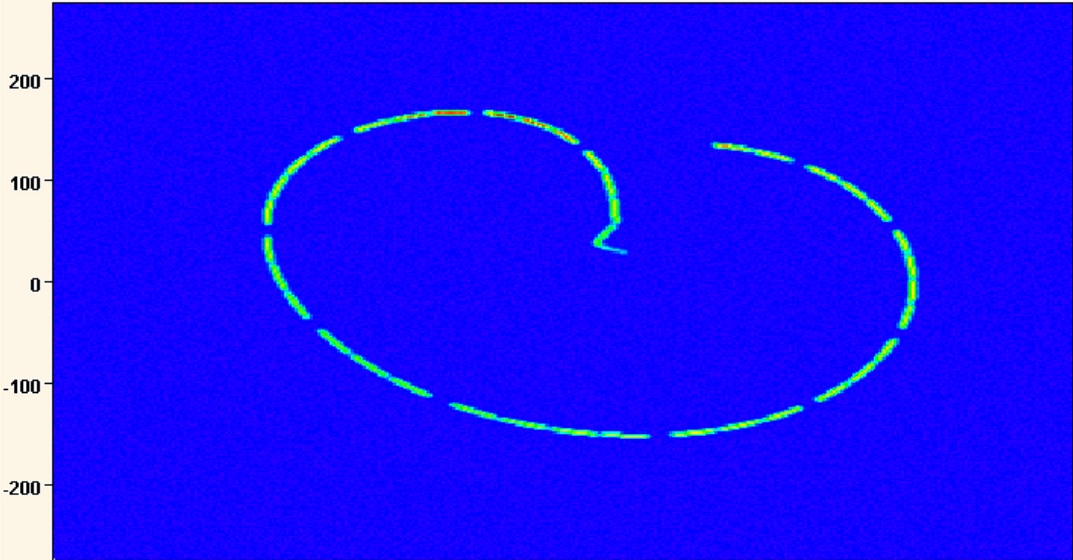
SDDS Default View

BTVDD_TD62 @ Cycle sdds.00_00_00_006 Update 00:00:00 6

Name	Type and Value	Axis
image	{short[][]:120000} -> 142, 116, 135, 143, ...	Z
positionX	{float[:]:400} -> -273.63, -272.25, -270.88, ...	X
positionY	{float[:]:300} -> -273.17, -271.33, -269.5, ...	Y
SLPenergy	{float:1} -> 7000.0	
acqError	{String:1} -> ""	
acqType	{long:1} -> 0	
camera	{short:1} -> 1	
cameraOffset	{short:1} -> 0	
cameraType	{short:1} -> 0	

Active keys : [X] -> x axis, [Y] -> y axis, [Z] -> z axis (image), [D] -> display line, [H] -> display histogram, [SPACE] -> clear, [T] -> time/numbers on x axis

Data for Cycle: -



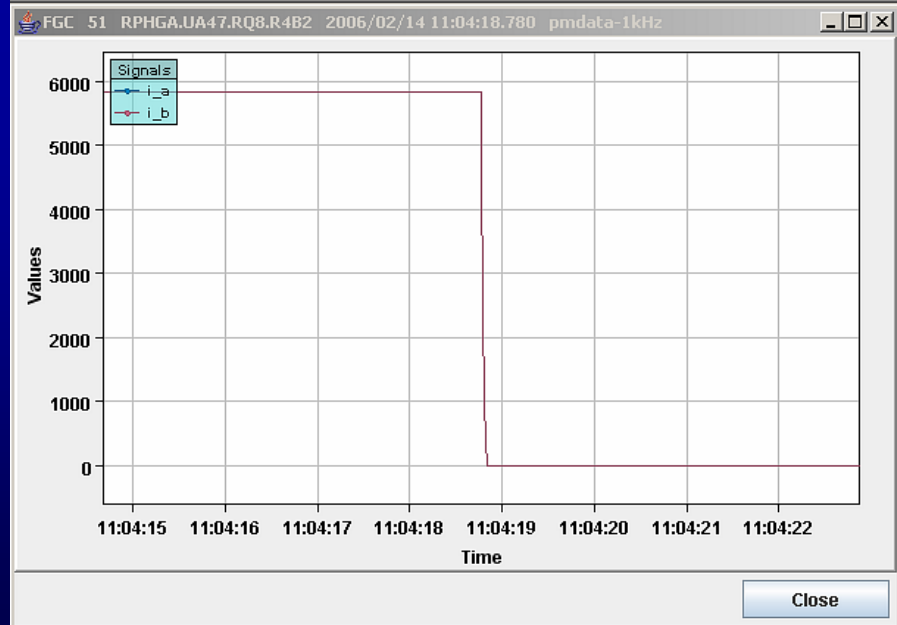
Point # 56099 X -17.8799991607666 Y 275.0 Z 122.0

Post Mortem

Simple PM Browser

SYSTEMS	CLASSES	Sources	Events	Data files	Arrays
BLM	40	RPHF.UA47.RD4.R4	2006/02/14 14:18:27.760	pmdata-100Hz	timestamp_sec
FGC	51	RPHF.UA83.RD1.L8	2006/02/14 14:33:42.440	pmdata-1kHz	timestamp_usec
PM		RPHF.UA83.RD2.L8	2006/02/14 14:36:37.440	pmdata-50Hz	i_ref
QPS		RPHF.UA83.RQX.L8	2006/02/15 12:44:22.220	pmdata-events	i_meas
		RPHGA.UA47.RQ10.R4B1	2006/02/15 14:44:01.900		v_ref
		RPHGA.UA47.RQ10.R4B2	2006/02/15 14:51:50.580		v_meas
		RPHGA.UA47.RQ7.R4B1	2006/02/16 11:13:59.040		
		RPHGA.UA47.RQ7.R4B2	2006/02/16 11:29:44.280		
		RPHGA.UA47.RQ8.R4B1	2006/02/16 11:50:18.120		
		RPHGA.UA47.RQ8.R4B2	2006/03/07 11:51:02.660		
		RPHGA.UA47.RQ9.R4B1			
		RPHGA.UA47.RQ9.R4B2			
		RPHGA.UA83.RQ10.L8B1			
		RPHGA.UA83.RQ10.L8B2			
		RPHGA.UA83.RQ7.L8B1			
		RPHGA.UA83.RQ7.L8B2			
		RPHGA.UA83.RQ8.L8B1			
		RPHGA.UA83.RQ8.L8B2			
		RPHGA.UA83.RQ9.L8B1			
		RPHGA.UA83.RQ9.L8B2			
		RPHGB.UA83.RQ5.L8B1			

Buttons: Refresh, Refresh, Refresh, Refresh, Refresh, Show in table, Show in chart



FGC 51 RPHGA.UA47.RQ9.R4B1 2006/02/16 11:13:59.040

timestamp...	timestamp...	i_ref	i_meas	v_ref	v_meas
11:13:38	550000	99.99999	100.00065	0.046878	0.047895
11:13:38	560000	99.99999	100.00743	0.046878	0.051579
11:13:38	570000	99.99999	100.00319	0.046878	0.051579
11:13:38	580000	99.99999	99.99811	0.046878	0.040527
11:13:38	590000	99.99999	99.995575	0.046878	0.040527
11:13:38	600000	99.99999	99.99726	0.046878	0.039299
11:13:38	610000	99.99999	100.0150...	0.046878	0.039299
11:13:38	620000	99.99999	100.0099...	0.046878	0.043802
11:13:38	630000	99.99999	99.99133	0.046878	0.043802
11:13:38	640000	99.99999	99.99219	0.046878	0.041755
11:13:38	650000	99.99999	100.00319	0.046878	0.041755
11:13:38	660000	99.99999	100.00065	0.046878	0.047486
11:13:38	670000	99.99999	99.98202	0.046878	0.047486
11:13:38	680000	99.99999	99.965935	0.046879	0.047076
11:13:38	690000	99.99999	99.98033	0.046879	0.047076
11:13:38	700000	99.99999	100.00234	0.046879	0.047895
11:13:38	710000	99.99999	99.98711	0.046879	0.047895
11:13:38	720000	99.99999	99.96424	0.04688	0.041755
11:13:38	730000	99.99999	99.96932	0.04688	0.041755
11:13:38	740000	99.99999	99.965935	0.046881	0.043392
11:13:38	750000	99.99999	99.96255	0.046881	0.043392
11:13:38	760000	99.99999	99.97770	0.046882	0.042672

Close

SDDS - Conclusions

- **Fast, Simple**
- **ASCII or Binary**

- **Interfaces**
 - C
 - C++
 - Java (and thus Mathematica)

- **Converters**
 - CVS, TVS, Matrix
 - Others easily developed

- **Already in use:**
 - Shot by shot
 - XPOC
 - Post Mortem

Acknowledgements: Greg Kruk