



DDN Collect Tool User Guide

Preface

DDN Collect is a new support utility that makes diagnostic data collection simpler and faster on SFA storage platforms. A single Python line command (or menu option) directs the tool to aggregate system status information, logs, and diagnostics into a single file transfer bundle for upload to DDN Support.

Audience

All operators of DDN SFA products.

About this Guide

This guide describes the DDN Collect Tool that is used to collect and send SFA system diagnostic data to DDN support. The latest version of this software and guide is available for download at:

- <https://www.ddn.com/download/ddn-collect/>

The software is also available directly through wget:

```
wget -o ddn_collect.zip https://www.ddn.com/?wpdmdl=49161
```

Related Documentation

The following documents are sources of information for this product:

- SFA OS User Guide
- SFA OS API Reference Guide

The latest version of the documentation is available on the DDN Support Portal at:

<https://community.ddn.com/login>

1. Introduction

The DDN Collect Tool is an easy new way to collect and send SFA diagnostics to support. It is easy to install and run, and is supported on most platforms. Compatible with all existing versions of SFA OS, it allows for a simple and straightforward way to gather the necessary files without error. Please contact DDN support regarding questions about usage of this tool.

2. Glossary

The following terms are used throughout this document.

API

Application Programming Interface. The DDN SFA OS API client is included in this tool. The API client allows for programmatic access to SFA system objects and logs.

DIAGNOSTICS

DIAGNOSTICS. Sometimes referenced as DIAGS. DIAGS are a collection of files containing developer logs, kernel logs, and system crash core files. Developer logs contain state of SFA OS firmware including memory contents, including a very high resolution log file called logdisk for SFA developer analysis. Kernel logs contain syslog, dmesg, log files from /var/log, and other files. During unexpected firmware events (e.g. null pointer access) core files saved in /tmp/corefiles for each RP (for 14KX, 2 core files) as janus_core-n.m “n” is the RP number, starting with 0; “m” is the corefile number, 1 for most recent, 2 for 2nd most recent crashes. Only last 4 instances kept.

CLUI

Command-Line User Interface. The CLUI is the interface used when you ssh into a controller.

SFA Logging

References to SFA Logging are regarding the tool creating entries in the SFA system logs stating that data collection has occurred. An entry is made at the start of collection time, and another entry upon successful completion.

3. Quick Start

Follow these simple steps to quickly make use of the tool:

1. **Extract:** `tar -zxvf ddncollect-1.3.5.tar.gz`
2. **Install:** `cd ddncollect-1.5.1; ./install.sh`
3. **Configure:** `python3 ddncollect.py configure`
(or edit `ddntools.conf` manually)
4. **Execute:** `python3 ddncollect.py collect`
5. **Upload:** `python3 ddncollect.py upload`
(only necessary if `auto_upload` not set)

4. Download

Please download the latest version from the following location. Improvements are continually being made to the tool and it is important to have the most up-to-date version.

Link to download location: <https://community.ddn.com/login>

5. System Requirements

The API client package is distributed alongside the collector tool. It requires the following packages to be installed on the target system:

- Python 2.7 –or– Python 3.4 or higher

- RedHat/CentOS Package List:

Install EPEL repository support: `yum -y install epel-release`

Common dependencies: `yum -y install curl`

Python 2.x: `yum install python-setuptools.noarch pexpect.noarch python2-six.noarch m2crypto.x86_64 python2-pyyaml.noarch`

Python 3.x: `yum install python36-ptyprocess.noarch python36-pexpect.noarch python36-six.noarch python36-PyYAML.x86_64`

- Debian/Ubuntu Package List (18.04 or higher recommended):

Common dependencies: `sudo apt install curl`

Python 2.x: `sudo apt install python-ptyprocess python-pexpect python-pkg-resources python-six python-yaml python-m2crypto`

Python 3.x: `sudo apt install python3-ptyprocess python3-pexpect python3-pkg-resources python3-six python3-yaml`

6. Installation

The tool is designed to support the various systems that exist in the field. It is currently supported on RedHat/CentOS and Debian/Ubuntu. Either Python version 2 or 3 may be used; Python 3 is recommended. If Python version 2 is used, it must be at least 2.7. To install, simply execute the install script found in the tool's home directory (e.g. `./install.sh`).

7. Execution

Execution Examples

Overview

```
python ddncollect.py [command]
```

commands:

help	Display this help
menu	Run the tool using a menu-based system
install	Install the tool
configure	Configure ddntools.conf
test	Test password-less connections to controllers
settings	View settings in ddntools.conf
collect	Collect data according to the settings
select	Select a bundle for upload
upload	Send selected bundle to DDN

Configure

This example configures `ddntools.conf` through the prompting mechanism enabled through the 'configure' command option

```
# python ddncollect.py configure
Configuration file ddntools.conf does not exist. Creating one at .
```

```
Configuring options for ddncollect tool. Please answer prompts. Options will be stored in ddntools.conf
```

```
SFA Admin Contact Name: : Joe Admin
SFA Admin Phone: : 512-555-1212
SFA Admin Email: : myemail@ddn.com
Company Name: : MyCompany
Case Number: : 123456
Collect CLUI Output: [True]:
Collect Diagnostics: [True]:
Core Files [1|2|3|all|lite|report]: [1]: 1
Directory to Store Data: [/tmp/ddncollect]:
Suppress Writing to SFA Logs: [False]:
Verbose Output?: [False]:
Auto Upload?: [True]:
```

```
Controller 0 IP [10.10.10.1]: 10.36.13.80
Controller 1 IP [10.10.10.2]: 10.36.13.81
Controller Username [user]:
```

Finished collecting configuration. Next recommended action: python ddncollect.py collect

Collect

This example collects DIAGS, CLUI, and API data and automatically uploads the data to DDN.

```
# python ddncollect.py collect
SFA system data collection has begun.

Password is not cached. You will be prompted for it.

Password for user@10.36.13.80:
CLUI output will be collected

DIAGS will be collected

    Core files setting: 1

Suppress not specified. Entries notifying start and stop of data extraction will be written to SFA system
log.

Auto upload selected. Data will be sent to DDN automatically after collection.

SFA system data collection via API has begun. Please wait...

Gathering Diagnostics from controller 0. Please wait...

/bin/tar: tmp/corefiles/logdisk: file changed as we read it

Gathering Diagnostics from controller 1. Please wait...

/bin/tar: tmp/corefiles/logdisk: file changed as we read it

Gathering CLUI output from controller 0.

Gathering CLUI output from controller 1.

Gathering bundles into single file...

Successfully extracted and bundled data to /tmp/ddncollect.

Auto Upload option is selected. Bundle will now be uploaded.

Sending data to DDN...
```

```
Command Executed: curl -T /tmp/ddncollect/2020-04-06-17-32-MyCompany-SFA7700XE-NU3N1302R-NU3N1302X-123456.tar -u anonymous:myemail@ddn.com ftps://ftp.ddn.com/upload/
```

```
% Total      % Received % Xferd  Average Speed   Time    Time       Time   Current
                               Dload  Upload  Total  Spent    Left     Speed

100  427M    0      0  100  427M      0  1153k  0:06:19  0:06:19  --:--:--  976k
```

Data successfully uploaded to DDN!

Test Connections

This example shows how to use the tool to test connections to the SFA controllers. Use this to determine if SSH keys are configured correctly. The test first tests password-based connection to both controllers, then SSH key-based authentication is checked.

```
# python ddncollect.py test
Testing password-based connections to controllers.
```

```
Continue? [Y]/N
Password for user@10.36.13.80:
```

Testing connection to user@10.36.13.80 using cached password:

```
show controller local
```

```
*****
*      Controller(s)      *
*****
```

Comm	Firmware Version	Up Time	Encl	Inter-Ctrlr
Idx Name	Mastership Locality	D: H: M: S RP	ID	Idx ULA
State ServicesUp	Release	Version Type		
0	es7k01-c0 PRIMARY LOCAL	0236:04:24:29 1	0001ff0a11d80000	0 00000001ff0a11e6 Up
DG+Msg+MAD 3.1.3.0 41240 Product				

Total Controllers: 1

Mon Apr 6 16:29:37 2020

Testing connection to user@10.36.13.81 using cached password:

```
show controller local
```

```
*****
*      Controller(s)      *
*****
```

Comm	Firmware Version	Up Time	Encl	Inter-Ctrlr
Idx Name	Mastership Locality	D: H: M: S RP	ID	Idx ULA
State ServicesUp	Release	Version Type		
1	es7k01-c1 SECONDARY LOCAL	0236:04:25:44 1	0001ff0a11e60000	3 00000001ff0a11e6 Up
DG+Msg+MAD 3.1.3.0 41240 Product				

Total Controllers: 1

Mon Apr 6 16:29:38 2020

Testing completed. If connections failed or you were prompted for a password, please correct the configuration and try again.

Select Bundle

Sometimes, it is necessary to upload a different bundle other than the one that was just created. Use 'select' to select a different bundle to upload. Note that the tool only looks in the configured output directory for bundles to select.

```
# python ddncollect.py select
Bundle(s) currently stored in the specified directory (/tmp/ddncollect):

[1]. 2020-03-25-23-18-Walt-Sample-Company-SFA7700XE-sysinfo.tar.gz
[2]. 2020-03-27-15-09-DDN-SFA7700XE-NU3N1302R-NU3N1302X-000201.tar
* [3]. 2020-04-06-17-32-MyCompany-SFA7700XE-NU3N1302R-NU3N1302X-123456.tar
[4]. 2020-03-27-15-53-DDN-SFA7700XE-NU3N1302R-NU3N1302X-000201.tar
[5]. 2020-03-30-15-19-DDN-c0-diag-pre_service.xz
[6]. 2020-03-30-15-24-DDN-SFA7700XE-NU3N1302R-NU3N1302X-000301.tar
[7]. 2020-03-27-14-47-DDN-SFA7700XE-NU3N1302R-NU3N1302X-000201.tar
[8]. 2020-03-30-15-19-DDN-SFA7700XE-sysinfo.tar.xz
[9]. 2020-03-30-15-19-DDN-c1-diag-pre_service.xz

[M]ain menu

* Bundle currently selected for upload

Enter a number to change selected bundle or 'Q' to quit.
>> 4

Selected bundle is now: 2020-03-27-15-53-DDN-SFA7700XE-NU3N1302R-NU3N1302X-000201.tar
```

Menu-based Usage

The ddncollect tool provides an easy-to-use interface that guides the user through configuration, collection, and upload through a menuing system. This method is a good choice if you are unfamiliar with the tool.

Menu-based example

The menuing system provides a handy interface for guiding you through using the tool. The main menu is presented here. The status of the tool is shown in bold text at the top, along with the next recommended action. As the user steps through the process, the status changes and the next step is highlighted as well. The actions that are performed when selected are the same ones as using the tool via command-line, so they are not repeated here.

```
# python ddncollect.py menu

DDN Support Data Collection Tool

STATUS: Tool Installed.      Recommended Action:  2. [C]onfigure the Data Collection Tool

Please select from the following choices. Enter either the number or letter:

1. [I]ninstall this tool
* 2. [C]onfigure the Data Collection Tool settings in ddntools.conf
3. [P]assword caching (Status: Not entered)
4. [T]est connections to controllers
5. [V]iew configuration settings
6. [E]xecute data collection according to settings in ddntools.conf
7. [S]elect support bundle to upload (Selected: 2020-03-27-15-53-DDN-SFA7700XE-NU3N1302R-NU3N1302X-000201.tar)
8. [U]pload selected bundle
```



```
0. [Q]uit
>>
```

A Note About Passwords

When data is collected, the program attempts to access the controller multiple times in various ways. Each time an access attempt is made, authentication must occur. Depending upon the configurations selected, the runtime may vary from less than one minute up to multiple minutes (if DIAGS are collected). To prevent the user from having to monitor the running program and entering a password for each attempt, the 'user' password is cached in the program memory and subsequently entered automatically for each access attempt. *The cached password is never written to stored disk nor will it appear in any logs or cli commands.*

If DIAGS are collected, password caching is not possible. However, it is possible to not be required to enter them through the use of SSH keys. If you would like to not have the 'diag' user password prompted for input during the execution of the program, you must configure SSH key usage on the controllers or alternatively, configure for SSH multiplexing.

1. Configuration Options

Configuration options are stored within the ddntools.conf file for ease of use and reference. Before collection of data, these options must be set. The options are grouped together into sections regarding user contact information, data collection options, SFA controller, SSH keys, and bundle. Each section is described below. You may create your own ddntools.conf either through use of the configure option or directly using the ddntools.conf.template.

ddntools.conf.template

```
#
# Contact Information
#
[Contact]
admin_name =
admin_phone =
admin_email =
company_name =
case_number =

#
# Data Collection Options
#
[Options]
collect_diags = True
collect_clui = True
core_files = 1
```

```

directory = /tmp/ddncollect
suppress_sfa_logging = True
verbose = True
auto_upload = True

#
# Controller Config
#
[Controller]
c0_ip = 10.10.10.1
c1_ip = 10.10.10.2
user_name = user

#
# Bundle to be uploaded by tool to DDN. Set and used by tool. DO NOT UPDATE.
#
[Bundle]
selected_bundle =

#
# Installation status. Set and used by tool. DO NOT UPDATE.
#
[Install]
install_status = Installed

```

[Contact]

This section allows you to enter the name of the person DDN support should contact regarding SFA operation guidance. Case Number should be the SR number that is produced when the support ticket was opened.

OPTION	ACCEPTED VALUES	DESCRIPTION
admin_name	Alphanumeric text	Name of primary contact of SFA system
admin_phone	Alphanumeric text	Phone number where primary contact can be reached
admin_email	Any valid email address	Email address where primary contact can be reached
company_name	Alphanumeric text	Name of your company
case_number	6-digit Support Case Number	Support Case SR number

[Options]

This section allows you to determine which data to collect from the SFA system. There are three ways data is extracted from the system: via API, via CLUI, and via the DIAGS command. API data extraction is always performed. Optionally, CLUI and DIAGS can be collected as well. Please collect what you are directed to by DDN support.

1.

OPTION	ACCEPTED VALUES	DESCRIPTION
collect_diags	True or False	Diagnostic files including vm crash cores
collect_clui	True or False	CLUI output of various commands
core_files	1,2,3,all,lite,report	Number of core files to collect when collecting diags
directory	Any valid directory	Where to place the collected data
suppress_sfa_logging	True or False	If True, informational entries are written to system log
verbose	True or False	If True, details about operations are displayed
auto_upload	True or False	If True, upload to DDN is performed automatically

[Controller]

Enter the IP addresses of the SFA controllers from which data is to be collected. At least one IP is required (c0_ip). The second controller IP is optional but recommended. If both controllers are accessible, please set both IPs. NOTICE: It does not matter which of these is the PRIMARY controller.

OPTION	ACCEPTED VALUES	DESCRIPTION
co_ip	IP address or hostname	Controller address or hostname
c1_ip	IP address or hostname	Controller address or hostname
user_name	'user' or the configured username	The account name of the controller

[Bundle]

The bundle section is automatically populated by the tool. The tool populates this option with the file path of the produced bundle, which is not known in advance. If you do need to use the tool to upload a different file, you may choose another one through: `python ddncollect.py select`.

2. What is Collected

Depending on the options selected, various amounts of data are collected from the system. There are three main types of data – API extracted data, CLUI output, and DIAGNOSTIC data. API extracted data is collected every time, while CLUI and DIAG data are optional. The API extracted data consists of system configuration, logs, and the contact information provided by the user of the tool. The contact information is stored in the BundleInfo.json exclusively. This file also contains important details about the SFA system such as the model, controller serial numbers, and license information.

Each collection is bundled separately as it is collected, then bundled and compressed together as a final step. If DIAGS and CLUI are collected from both controllers, a maximum of five file will exist in the final bundle. For example:

```
# tar -tvf 2020-04-08-19-07-DDN-SFA7700XE-NU3N1302R-NU3N1302X-123456.tar
-rw-r--r-- root/root      67747 2020-04-08 19:07 ./2020-04-08-19-07-DDN-SFA7700XE-sysinfo.tar.gz
-rw-r--r-- root/root     2761329 2020-04-08 19:08 ./2020-04-08-19-07-DDN-c0-text-pre_service.txt
-rw-r--r-- root/root      53052 2020-04-08 19:07 ./2020-04-08-19-07-DDN-c0-diag-pre_service.xz
-rw-r--r-- root/root     2761326 2020-04-08 19:08 ./2020-04-08-19-07-DDN-c1-text-pre_service.txt
-rw-r--r-- root/root      53012 2020-04-08 19:08 ./2020-04-08-19-07-DDN-c1-diag-pre_service.xz
```

The final bundle naming format includes your company name, SFA platform, controller serial numbers, and case number. Its format is as follows: YYYY-MM-DD-HH-mm-CompanyName-SystemPlatform-Controller0Serial-Controller1Serial-CaseNumber.tar

3. System API Data

Data collected via API is system health status, configuration details, and the system logs. The data is stored in JSON files and bundled into a single file named YYYY-MM-DD-HH-mm-CompanyName-SystemPlatform-sysinfo.tar. This data is always collected and there is no option to not collect it.

The following tables show potentially sensitive data included in the files.

IP Addresses

These are the IP addresses collected via the SFA API and the file(s) they are found in:

FILE	FIELD	DESCRIPTION
SFAController.json	NTPSyncAddress	IP Address of the NTP server
SFADiscoveredInitiator.json	IPAddress	IP Address of the Initiator
SFAEnclosure.json	BmcIPAddress	BMC IP Address
	BmcIPAddressAlt	BMC Alternate IP Address
SFAHostChannel.json	IPAddress	Host Channel IP Address

FILE	FIELD	DESCRIPTION
SFAICLChannel.json	EthernetIPAddress	ICL Channel IP Address
SFASNMPTrapAgent.json	IPAddress	SNMP Trap Agent IP Address
SFAUserInterface.json	SyslogIPAddress	Syslog IP Address
	SNMPTrapIPAddress	SNMP Trap IP Address
	EmailIPAddress	Email server IP Address
	EmailBackupIPAddress	Email server Backup IP Address
	IPv4Addresses	Controller IPv4 Address
	IPv6Addresses	Controller IPv6 Address
	ControllerIPAddress	Controller Alternate IP Address
	ControllerIPGateway	Controller Gateway IP Address
	ControllerIPGatewayAlt	Controller Gateway Alternate IP Address

Email Addresses

These are the Email Addresses collected via the SFA API and the file(s) they are found in:

FILE	FIELD	DESCRIPTION
SFAUserInterface.json	EmailToAddress	Email To Address
	EmailFromAddress	Email From Address

Hostnames

These are the Hostnames collected via the SFA API and the file(s) they are found in:

FILE	FIELD	DESCRIPTION
SFAStorageSystem.json	Name	System hostname (not canonical)
SFAController.json	Name	Controller hostname (not canonical)
	FWBuildHost	Firmware Build hostname (not canonical)

Username

These are the Usernames collected via the SFA API and the file(s) they are found in:

FILE	FIELD	DESCRIPTION
SFAUserAccount.json	Name	Controller user account username
	GroupName	Controller user account group
	UserID	Controller account UID
	GroupID	Controller account GID

BundleInfo.json

This is an example of what is stored in the BundleInfo.json file:

```
[
  {
    "Customer": "DDN",
    "ContactPhone": "555-555-5555",
    "uuid": "60001ff0a0af600000000000030000000",
    "DateAndTime": "2020-03-26T10:36:49.922987-06:00",
    "Controller0Serial": "NT7F12061-R",
    "LicenseNeeded": true,
    "ContactEmail": "support@ddn.com",
    "Platform": "SFA7700X",
    "Controller1Serial": "null",
    "ContactName": "Mr. S. Fusion Administrator",
    "TimeZone": "America/Denver",
    "Licenses": [],
    "Description": "1-SS7700X(head only) with 0 missing enclosures (AUTOMATIC Selection).",
    "CollectorVersion": "1.0",
    "SFAAPIVersion": "11.7.0",
    "CaseNumber": "00508"
  }
]
```

Sample 2020-02-25-11-12-My-Company-SFA7700XE-sysinfo.tar

```
# 11
total 4676
-rw-r--r--. 1 root root      623 Feb 25 13:10 BundleInfo.json
-rw-r--r--. 1 root root 3584272 Feb 25 13:10 logmessages0.txt
```

```

-rw-r--r--. 1 root root 77397 Feb 25 13:10 logmessages1.txt
-rw-r--r--. 1 root root 2 Feb 25 13:10 SFAAllowedIPRange.json
-rw-r--r--. 1 root root 2 Feb 25 13:10 SFAAuthentication.json
-rw-r--r--. 1 root root 4103 Feb 25 13:10 SFAClientIOC.json
-rw-r--r--. 1 root root 2 Feb 25 13:10 SFAConfigurationFile.json
-rw-r--r--. 1 root root 9848 Feb 25 13:10 SFAConnector.json
-rw-r--r--. 1 root root 3293 Feb 25 13:10 SFAController.json
-rw-r--r--. 1 root root 2 Feb 25 13:10 SFADiscoveredInitiator.json
-rw-r--r--. 1 root root 5762 Feb 25 13:10 SFADiskChannel.json
-rw-r--r--. 1 root root 272837 Feb 25 13:10 SFADiskDrive.json
-rw-r--r--. 1 root root 251802 Feb 25 13:10 SFADiskDriveStatistics.json
-rw-r--r--. 1 root root 88814 Feb 25 13:10 SFADiskSlot.json
-rw-r--r--. 1 root root 716 Feb 25 13:10 SFAEnclosureConfigurations.json
-rw-r--r--. 1 root root 7891 Feb 25 13:10 SFAEnclosure.json
-rw-r--r--. 1 root root 7834 Feb 25 13:10 SFAExpander.json
-rw-r--r--. 1 root root 7020 Feb 25 13:10 SFAFan.json
-rw-r--r--. 1 root root 2 Feb 25 13:10 SFAHostChannel.json
-rw-r--r--. 1 root root 810 Feb 25 13:10 SFAHost.json
-rw-r--r--. 1 root root 6884 Feb 25 13:10 SFAICLChannel.json
-rw-r--r--. 1 root root 2368 Feb 25 13:10 SFAICLIOC.json
-rw-r--r--. 1 root root 738 Feb 25 13:10 SFAImage.json
-rw-r--r--. 1 root root 2 Feb 25 13:10 SFAInitiator.json
-rw-r--r--. 1 root root 3946 Feb 25 13:10 SFAInternalDiskDrive.json
-rw-r--r--. 1 root root 3518 Feb 25 13:10 SFAIOC.json
-rw-r--r--. 1 root root 2 Feb 25 13:10 SFAJob.json
-rw-r--r--. 1 root root 5486 Feb 25 13:10 SFAPowerSupply.json
-rw-r--r--. 1 root root 16370 Feb 25 13:10 SFAPresentation.json
-rw-r--r--. 1 root root 960 Feb 25 13:10 SFARaidProcessor.json
-rw-r--r--. 1 root root 2510 Feb 25 13:10 SFASEP.json
-rw-r--r--. 1 root root 823 Feb 25 13:10 SFASNMPAgent.json
-rw-r--r--. 1 root root 435 Feb 25 13:10 SFASNMPTrapAgent.json
-rw-r--r--. 1 root root 2 Feb 25 13:10 SFASNMPTrapReceiver.json
-rw-r--r--. 1 root root 2 Feb 25 13:10 SFASNMPUser.json
-rw-r--r--. 1 root root 2 Feb 25 13:10 SFASparePool.json
-rw-r--r--. 1 root root 2 Feb 25 13:10 SFASStack.json
-rw-r--r--. 1 root root 23356 Feb 25 13:10 SFAStoragePool.json
-rw-r--r--. 1 root root 1580 Feb 25 13:10 SFAStorageSystem.json
-rw-r--r--. 1 root root 2 Feb 25 13:10 SFA syslogAgent.json
-rw-r--r--. 1 root root 19223 Feb 25 13:10 SFATemperatureSensor.json
-rw-r--r--. 1 root root 314 Feb 25 13:10 SFAUnassignedPool.json
-rw-r--r--. 1 root root 2358 Feb 25 13:10 SFAUPS.json
-rw-r--r--. 1 root root 1229 Feb 25 13:10 SFAUserAccount.json
-rw-r--r--. 1 root root 357 Feb 25 13:10 SFAUserGroup.json
-rw-r--r--. 1 root root 1379 Feb 25 13:10 SFAUserInterface.json
-rw-r--r--. 1 root root 29088 Feb 25 13:10 SFAVirtualDisk.json
-rw-r--r--. 1 root root 157540 Feb 25 13:10 SFAVirtualMachine.json
-rw-r--r--. 1 root root 902 Feb 25 13:10 SFAVirtualProcessor.json

```

```
-rw-r--r--. 1 root root 48049 Feb 25 13:10 SFAVoltageSensor.json
-rw-r--r--. 1 root root 32 Feb 25 13:10 UUID
```

4. CLUI Data

Textual output of SFA CLUI commands is collected if this option is selected. It is stored in a bundle named YYYY-MM-DD-HH-MM-CompanyName-controllerIndex-text-pre_service.txt. The command list is as follows:

```
show subsystem summary short
show subsystem summary
show subsystem summary all
show subsystem fault
show subsystem fault all
show sas_connector
show sas_connector all
show ioc fault
show ioc
show ioc phy
show ioc all
show physical_disk phy_errors
app show discovered *
app show host *
app show initiator *
app show presentation *
app show presentation * all
app show processor *
app show channel *
app show channel errors
app show stack * all
app show image *
app show ioc * all
app show subsystem summary
app show subsystem summary all
ui show api
ui show account
ui show account all
ui show cli
ui show email
ui show gui
ui show network
ui show network all
ui show ntp
ui show snmp_agent
ui show snmp_trap_agent
ui show stats_server
ui show syslog
```


Sample CLUI file: 2020-02-25-11-12-My-Company-c0-text-pre_service.txt

```
# head -25 2020-02-25-11-12-My-Company-c0-text-pre_service.txt
Keyword belongs to a mutually exclusive group that has already been specified,
please choose from:
Mutually Exclusive Optional:
Optional keywords:
    ALL_ATTRIBUTES    Show all attributes for the specified object
```

Tue Feb 25 11:12:58 2020

```
*****
*      Subsystem Summary      *
*****
```

```
*****
*      Subsystem      *
*****
```

Name	Time	Locate	Fast	Verify	Licenses	Dwell Time	Timeout	Policy	UUID
es7k01	Tue Feb 25 11:12:58 2020		ON	DISABLED		120 seconds			
60001ff0a11e600000000000030000000	SFA7700XE								

```
*****
```

5. Diagnostic Data

If the DIAG option is selected, a collection of files containing developer logs, kernel logs, and system crash core files is produced. Developer logs contain state of SFA OS firmware including memory contents, including a very high resolution log file called logdisk for SFA developer analysis. Kernel logs contain syslog, dmesg, log files from /var/log, and other files. During unexpected firmware events (e.g. null pointer access) core files saved in /tmp/corefiles for each RP (for 14KX, 2 core files) as janus_core-n.m “n” is the RP number, starting with 0; “m” is the corefile number, 1 for most recent, 2 for 2nd most recent crashes. Only last 4 instances kept.

It is stored in a bundle named YYYY-MM-DD-HH-MM-CompanyName-controllerIndex-diag-pre_service.xz.

Sample DIAG file: 2020-02-25-11-12-My-Company-c0-diag-pre_service.xz

The file tree of the contents of this bundle is shown here:

```
[root@vm78-centos7 data]# tar -Jxvf 2020-02-25-11-12-My-Company-c0-diag-  
pre_service.xz  
tmp/audit  
ddn/janus_version.txt  
boot/grub/  
boot/grub/interfaces.4  
boot/grub/lost+found/  
boot/grub/interfaces.1  
boot/grub/interfaces.2  
boot/grub/config/  
boot/grub/config/snmpd.conf  
boot/grub/config/gui.conf  
boot/grub/config/timezone.conf  
boot/grub/config/snmp_trap.conf  
boot/grub/config/etc/  
boot/grub/config/etc/shadow  
boot/grub/config/smtp.conf  
boot/grub/config/syslog.conf  
boot/grub/config/stats.conf  
boot/grub/config/roles.conf  
boot/grub/config/snmp/  
boot/grub/config/snmp/mibs/  
boot/grub/config/snmp/mibs/SFA-TRAP-MIB.txt  
boot/grub/config/snmp/mibs/.index  
boot/grub/config/snmp/mibs/SFA-INFO.txt  
boot/grub/config/cpPersStoreFile.db  
boot/grub/config/api.conf  
boot/grub/interfaces.tmp  
boot/grub/menu.lst  
boot/grub/interfaces  
boot/grub/stage1  
boot/grub/interfaces.3  
boot/grub/stage2  
boot/grub/default  
boot/grub/menu.lst.template  
boot/grub/e2fs_stage1_5  
boot/grub/uuid.txt  
tmp/diag_health_monitoring_report.txt  
tmp/ipmitool_sel_list.txt  
tmp/ipmitool_sel_list.raw  
tmp/systemd_journal.txt.gz  
tmp/dmesg.current  
var/log/auth.log  
var/log/auth.log.1  
var/log/auth.log.2.gz  
var/log/auth.log.3.gz  
var/log/auth.log.4.gz
```

var/log/boot.log
var/log/boot.log.0
var/log/boot.log.10.gz
var/log/boot.log.11.gz
var/log/boot.log.12
var/log/boot.log.1.gz
var/log/boot.log.2.gz
var/log/boot.log.3.gz
var/log/boot.log.4.gz
var/log/boot.log.5.gz
var/log/boot.log.6.gz
var/log/boot.log.7.gz
var/log/boot.log.8.gz
var/log/boot.log.9.gz
var/log/janus_config.log
var/log/janus_config.log.1
var/log/janus_config.log.2
var/log/janus_config.log.3
var/log/janus_config.log.4
var/log/janus_config.log.5
var/log/janus_config.log.6
var/log/janus_config.log.7
var/log/jerry.log
var/log/janus/
var/log/janus/eventlog
var/log/syslog
var/log/syslog.1
var/log/syslog.2.gz
var/log/syslog.3.gz
var/log/syslog.4.gz
var/log/syslog.5.gz
var/log/syslog.6.gz
var/log/syslog.7.gz
tmp/corefiles/compressed-persist-data-file
tmp/corefiles/cpld.log
tmp/corefiles/cpld.log.1
tmp/corefiles/cpld.log.10
tmp/corefiles/cpld.log.2
tmp/corefiles/cpld.log.3
tmp/corefiles/cpld.log.4
tmp/corefiles/cpld.log.5
tmp/corefiles/cpld.log.6
tmp/corefiles/cpld.log.7
tmp/corefiles/cpld.log.8
tmp/corefiles/cpld.log.9
tmp/corefiles/cpPersStoreFile.db1442031319 (2015-09-11 20:15:19)
tmp/corefiles/cpPersStoreFile.db1467393207 (2016-07-01 10:13:27)

```
tmp/corefiles/dmesg.txt.1
tmp/corefiles/persistent-data-on-quorum
tmp/corefiles/Stack-0x10000000.log
tmp/corefiles/Stack-0x10000000.log.1
tmp/corefiles/Stack-0x10000000.log.10
tmp/corefiles/Stack-0x10000000.log.2
tmp/corefiles/Stack-0x10000000.log.3
tmp/corefiles/Stack-0x10000000.log.4
tmp/corefiles/Stack-0x10000000.log.5
tmp/corefiles/Stack-0x10000000.log.6
tmp/corefiles/Stack-0x10000000.log.7
tmp/corefiles/Stack-0x10000000.log.8
tmp/corefiles/Stack-0x10000000.log.9
tmp/corefiles/Stack-0x10000001.log
tmp/corefiles/Stack-0x10000001.log.1
tmp/corefiles/Stack-0x10000001.log.2
tmp/corefiles/Stack-0x10000001.log.3
tmp/corefiles/Stack-0x10000001.log.4
tmp/corefiles/Stack-0x10000001.log.5
tmp/corefiles/Stack-0x10000001.log.6
tmp/corefiles/Stack-0x10000001.log.7
tmp/corefiles/Stack-0x10000001.log.8
tmp/corefiles/Stack-0x10000001.log.9
tmp/corefiles/testfile
```

6. What is Not Collected

Only system data and logs are collected. Absolutely no data coming from file systems or the data stored on the SFA systems is gathered. This includes file system structure and format, including directory paths, mount points, or file names.

7. Uninstall

To uninstall the tool, simply delete all the files stored in the installation directory. Optionally, you may also delete any prerequisite packages that were required for the install. No additional files are installed or changed outside of the installation directory.

8. Support

If you have questions or require assistance, contact DDN Support:

Web

DDN Community Support Portal
Portal Assistance

<https://community.ddn.com/login>
webportal.support@ddn.com

Telephone

DDN Support Worldwide Directory

<https://www.ddn.com/support/global-services-overview/>

Email

Support Email

support@ddn.com

Bulletins & Notices

Support Bulletins

End-of-Life Notices

Release Notes

Subscription Requests

<http://www.ddn.com/support/technical-support-bulletins>

<http://www.ddn.com/support/end-of-life-notices>

<https://community.ddn.com/login>

support-tsb@ddn.com