Title

RS Log Interface Control Document

Prepared (Author)

        Signature :   **Date :**

        Name / Function   **Pierre CUQ**
Reference System Engineer

Approved

        Signature :   **Date :**

        Name / Function   **Jean-Luc PASCAL**
Reference System Quality Manager

Pierre CUQ
Reference System Engineer

Released

        Signature :   **Date :**

        Name / Function   **Benjamin BRARD**

Reference System project manager

Document type	Nb WBS	Keywords
ICD		
CMMI-SVC	CM	

Export Control Information

This document contains EU or/and US Export Controlled technology (data) :
YES ☐   NO ☒

If **YES** :

1/ European/French regulation controlled content

☐ Technology contained in this document is controlled by the European Union in accordance with dual-use regulation 428/2009 under Export Control Classification Number [xExx]. *(1)*

☐ Technology contained in this document is controlled by Export Control regulations of French Munitions List under Export Control Classification Number [MLXX or AMAXX]. *(1)*

2/ US Regulation controlled content

☐ Technology contained in this document is controlled under Export Control Classification Number [xExxx] by the U.S. Department of Commerce - Export Administration Regulations (EAR). *(1)*

☐ Technology contained in this document is controlled by the U.S. Department of State - Directorate of Defense Trade Controls - International Traffic in Arms Regulations (ITAR). *(1)*

(1) See applicable export control license/authorization/exception in Delivery Dispatch Note.

SUMMARY

This ICD explains how an application must format its logs, so all information for Reference System platform monitoring can be stored and computed.

CHANGE LOG

Issue/Revision	Date	Change Requests	Observations
01	30 Aug 2021	N/A	Derived from S1PDGS cloud system ICD S1PD-ICD-ADST-1000585122 issue 2.0 ● Add facultative field "mission"
02 draft A	28/06/2022		● Delete Kubernetes section ● Increase message maximum size to 10 Kbytes
02	05/09/2022		**Remove draft**. Release for RS system v1.1.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. GENERAL INFORMATION

1.1 PURPOSE

The LOG ICD explains how an application must format its logs, so all information for application monitoring can be stored and computed.

1.2 GENERAL

The application **must write all logs** to the standard outputs **stdout** and **stderr**.
Each message is in JSON format to facilitate integration and analysis.

2. LOG FORMAT

Log are structured in following sections:

Header

Message

Application custom fields

1. HEADER

key	description	mandatory	value sample	format
type	message type	YES	"LOG"	string "LOG"
timestamp	message date and hours (UTC)	YES	"2019-01-21T05:24:40.000000Z"	<YYYY>-<MM>-<DD>T<hh>:<mm>:<ss>.<microseconds>Z
mission	mission identifier	NO	"S1", "S2", "S3"	string "S1" | "S2" | "S3"
level	message level	YES	"ERROR"	string "INFO" | "WARNING" | "ERROR" | "DEBUG" | "FATAL
line	line where the log is generated	NO	"1234"	integer
file	file causing the log	NO	"TestLogOutPut.java"	string maximum size of 256
thread	thread causing the log	NO	"main"	string maximum size of 256

2. MESSAGE

key	description	mandatory	value sample	format
content	message itself	YES	"free text..."	string maximum size of 10 Kbytes UTF-8 character format

3. APPLICATION CUSTOM FIELDS

The application can add custom fields in an additional section.
These fields are nested on a tag named "**custom**" using JSON format.
This additional section is facultative. It respects the following format :

```
"custom": {
    "key1" : "value1",
    "key2" : "value2",
        ...
    "keyN" : "valueN"
}
```

■ Key

"**key**" is formatted following these rules:

snake_notation	Each key is a string field. 256 characters maximum length. It uses a snake_notation : all word lowercase written and separated with underscore character (_) and no space.
single key suffix	The suffix provides information about key value type: - numeric type: "_long" | "_integer" | "_short" | "_byte" | "_double" | "_float" - other type: "_boolean" | "_string" | "_date" | "_object"
array key suffix	The suffix provides information about the key value type for **all the array elements.** The array key suffix is identical to the single key suffix with the extra '**s**' at the end. - array of numeric type : "_long**s**" | "_integer**s**" | "_short**s**" | "_byte**s**" | "_double**s**" | "_float**s**" - array of other type: "_boolean**s**" | "_string**s**" | "_date**s**" | "_object**s**"

■ Value

One single "**value**" can have one of the following types:

numeric value	long, integer, short, byte, double, float
boolean	true | false
string	UTF-8 character format / maximum size of 256
date	UTC only <YYYY>-<MM>-<DD>T<hh>:<mm>:<ss>.<microseconds>Z
object	{"key1" : "value1", "key2" : "value2"}, ... "keyN" : "valueN"} Each "key" embedded on an object complies to the same rules as the single key.

An array has got format : **"key" : [value1", "value2", "value3"]** where all values has got the same type.

For example: "my_key_string**s**" : ["a1", "abcd", "Test123"]

Array type must be avoided as much as possible because It prevents introspection from monitoring tools.

When the array only contains one element, it must be replaced by a "single" value.

■   **Example**

```
{
"custom": {
    "my_param0_float": "12.34",
    "my_filename_strings": [ "temp1.txt", "tmp.csv","tmp.xml"],
    "my_param1_object": {

     "item1_long": "123",
     "item2_boolean": "false
     },
    "sample_date": "2019-01-21T05:24:40.000000Z"
    }
}
```