DOCUMENT NUMBER: **SUM7056227**

DOCUMENT REVISION: **A**

DATE: **21 June 2024**

CAGE CODE: **A0810**

GRIFO-E/600

GRIFO-E RIS SOFTWARE USER MANUAL

Document Authority

Owner: Alberto Chessa **SW Project Engineering Lead** Nerviano / EO and Radar Sensors	…… Alberto Chessa
Verification: Alberto Chessa **SW Project Engineering Lead** Nerviano / EO and Radar Sensors	…… Alberto Chessa
Approval: Elisa Terreni **SW Product Assurance Engineer** Quality Airb.Sensors & Missions Systems	…… Elisa Terreni
Authorization: Carlo Sala **COMBAT IPT Program Manager** Airborne Radar Programmes	…… Carlo Sala

<Point of Contact>

Alberto Chessa **SW Project Engineering Lead** Nerviano / EO and Radar Sensors e-mail : alberto.chessa@leonardocompany.com Tel. : +39 03311753417	**Leonardo S.p.a.** Site: **Nerviano C.so Europa snc.** **20014 Nerviano (Milan) Italy**

REVISION RECORD

Rev.	Modification N.	Date	Description	Author/s
A	N/A	**21 June 2024**	First issue	A. Chessa

TABLE OF CONTETS

INDEX OF FIGURES

INDEX OF TABLES

1 SCOPE

1.1 Identification

This Software User Manual (SUM) is applicable to the software Configuration Item identified as:

IBC	Title
S1004081	GRIFO-E RIS SW CSCI

1.2 System overview

The GRIFO-E Radar Interface Simulator (RIS) is a system designed to simulate the GRIFO-E Radar interfaces for preliminary integration with a control Missions Computer in accordance with GRIFO-E Data Exchange Control Document (DECD).

The RIS:

- Interface with a Mission Computer, via a MIL-STD-1553B bus
- Simulate the data exchange and behavior of the GRIFO-E Radar
- Simulate the Video output of the GRIFO-E Radar, via Gigabit Ethernet interface
- Simulate targets and BIT failures in such a way to permit the stimulation and test of the data specified in the Radar DECD

It is composed of a WINTEL Personal Computer (RIS-PC) and RIS software (RIS-SW) that interface to:

- The Mission Computer, via an Ethernet RJ-45 built-in port
- The user, via the built-in display and keyboard

1.3 Document overview

This purpose of this SUM is to supply all the required information to use the GRIFO-E RIS SW:

- Software installation and configuration
- Available features
- Man Machine interface user manual
- Processing reference guide

The following naming conventions are used:

- Radar: the GRIFO-E Radar
- RIS: always means the bundle of the GRIFO-E Radar Interface Simulator software running on the WINTEL PC, equipped with MIL-STD-1553 device, configured as released by Leonardo.
- RIS-PC: refers the specific characteristic of the hosting WINTEL PC
- RIS-SW: refers to the software only

This document uses the following typographic conventions:

- **calibri bold** font is used to highlight Microsoft Windows console command
- calibri font is used to highlight Microsoft Windows file system file name and directory path
- **[square bracket consolas bold]** is used to highlight Microsoft Windows console and MMI control key sequence, such as **[CTRL+C]** to indicate to press together CTRL key and C key
- **courrier-new bold** font is used to highlight Man-Machine-Interface (MMI) user command, typed on the MMI input interface (video terminal keyboard)
- courrier-new plain font is used to highlight MMI reply, displayed on the MMI output interface (video terminal window)

Potential dangerous operation or required user cautions are prefixed by "**WARNING:**".

For the understanding of this document it is required a well knowledge of the Radar functionalities, Radar external interface and Radar Data Exchange Control document (DECD).

For a complete list of used acronyms see §6.1.

2 REFERENCED DOCUMENT

2.1 Applicable document

Ref.	Code	Title
A1.	ECS1004081-01 rev. A	GRIFO-E RIS Executable Code

Table 2-1: Applicable documents

2.2 Reference document

Ref.	Code	Title
[ESC]	ISO/IEC 6429	Information technology – Control functions for coded character sets
[ASCII]	ISO/IEC 646	Information technology – ISO 7-bit coded character set for information interchange

Table 2-2: Referenced documents

3 SOFTWARE SUMMARY

3.1 Software application

This software is intended for simulation of the GRIFO-E Radar in term of:

- GRIFO-E MIL-STD-1553 Remote Terminal
- MFD and SAR image serialization over Ethernet
- Simulated Radar Graphic Display
- Simulated Radar modes
- Simulated targets, limited to the characteristic required to permit to simulate the Radar modes and Radar display presentation
- Simulated BIT failure

3.2 Software inventory

The SW is released as a single file:

File	Contents
GRIFO-E-RIS-Installer.exe	The installer executable

Table 3-1: Software inventory

The installation executable automatically creates the following files and folders:

File	Contents
C:\Leonardo\GRIFO-E-RIS\RIS	Root application folder
C:\Leonardo\GRIFO-E-RIS\RIS\bin\GRIFO-E-RIS-Launcher.exe	RIS-SW GUI launcher
C:\Leonardo\GRIFO-E-RIS\RIS\bin\GRIFO-E-RIS.exe	RIS-SW executable
C:\Leonardo\GRIFO-E-RIS\RIS\Config	Configuration folder, where the SW will store internal configuration data files (see 5.4)
C:\Leonardo\GRIFO-E-RIS\RIS\REC	Data Recorder folder, where the SW will store the log file and internal data recorded upon user request
C:\Leonardo\GRIFO-E-RIS\RIS\user	A user folder, where the user can store command script files
C:\Leonardo\GRIFO-E-RIS\RIS\example	A folder containing some example script files

Table 3-2: Software installed files

3.3 Software environment

The RIS requires a WINTEL Personal Computer with the following characteristics:

CPU	At least Intel i5,11Th Gen or equivalent
Display resolution	Minimum SVGA (800x600), XGA (1024x768) or higher recommended
RAM	8 Gigabytes
HD Free space	At least 8MBytes for the executable. Additional free space could be required to store recording data and for hosting operating system operation
Network I/F	One Gigabit Ethernet RJ-45 port, configured as: • DHCP: Disabled • IPv4 address: 192.168.1.142 • IPv4 Subnet mask: 255.255.255.0 • Network adapter Speed: automatic Note that even if the RIS-PC is release with auto negotiation enabled, to be representative of the real Radar I/F the adapter shall be connected to a full duplex 1Gigabit network (preferable via an Ethernet switch)
OS	Windows 10 Pro 64, Verison 22H2
Serial Port	Optional, PC is delivered without serial port. The SW support an RS232 compatible serial port (via USB adaptor), required to reproduce the serial output of the MMI of the Radar: • 115200 baud • 1 Stop bit • 8 data bits • Parity node • Flow control disabled

Table 3-3: HW/SW Environment requirements

External Device and Third part SW drivers:

MIL-STD-1553 Device	DDC BU-67202U100L-CA0, USB MIL-STD-1553 device.
MIL-STD-1553 Driver	DDC BU-69092S0 version 3.6.15. The BU-67202U100L-CA0 shall be configured as device number 0 (zero)
MIL-STD-1553 USB Cable	USB Type A Female (PC side) - Mini B (BU-67202U100L Side)

Table 3-4: External device

For details about the DDC device and its drivers, see §0.

3.4 Software organization and overview of operations

The RIS-SW is physically composed of a single executable module (GRIFO-E-RIS.exe) that realize all GRIFO-E Radar Interface Simulator capabilities.

It is logically subdivided into the following components:

Component	Role
Radar Engine	Replicate the behaviors of the Radar data processing, including Radar graphic display bitmap generation
Simulator Engine	Simulate the antenna, targets and detection
MMI Engine	Implements the RIS Man-Machine Interface command interpretation and execution
I/O Simulation	Simulate the Radar I/O in term of: • MMI I/O channel (video terminal) • Radar display • Remote Terminal MIL-STD-1553

Table 3-5: Logical components

The SW interface with the user and interfacing system with:

1. **MIL-STD-1553 Remote Terminal.**
2. A graphic window, hosting the simulation of the Radar display and suppling a minimum of user operation via context menu (applicable only to the RIS PC version, not to the real Radar video interface)
3. A windows console, hosting a command line oriented Man-Machine-Interface (MMI)
4. An optional Serial Communication port (RS232 compatible Windows "COM" port), hosting a video terminal interface. This is an input/output replication of the MMI available on the RIS window console and can be used to control the simulation from an external device

The MIL-STD-1553 Remote Terminal interface shall be connected to an external Mission Computer that shall provide all the input and process all the output produced by the RIS.

3.4.1 Performance

The RIS performance depends on:

- the non-real time characteristics of the hosting operative system;
- Ethernet traffic not under the control of the RIS (Microsoft Windows services);
- external device sharing the same Ethernet bus;
- PC disk activity in case of data recording (see §5.3.1.3);
- other user operation not directly related with the RIS operation described in this manual (for instance, opening other applications, changes in Windows configuration).

The RIS SW and the RIS PC released by Leonardo are configured to guarantee response time comparable with a real Radar. Nerveless, the above dependency could cause possible reply latency jittering, but without generate a MIL-STD-1553 minor frame reply overrun or RIS processing overrun.

During data recording (see §5.3.1.3), the RIS performance could be affected also by the disk access time. Even in this case, the PC configuration released by Leonardo should always guarantee overall performance adequate to simulate a real Radar performance. By the interfacing system point of view, the effect of this performance related factor will be viewed as jittering in the MIL-STD-1553 RIS TX messages update time.

3.5 Contingencies and alternate states and modes of operation

None.

3.6 Security and privacy

The RIS SW, RIS PC and every file or image produced by the SW execution, are submitted to security and privacy obligations in accordance with the Non-Disclosure Agreement (NDI) and any other license and loan limitations established by the Purchase Order.

3.7 Assistance and problem reporting

None.

4 ACCESS TO THE SOFTWARE

4.1 First-time user of the software

4.1.1 Equipment familiarization

The SW can be accessed from the Windows user RIS. After Windows boot, login as user RIS and double click on the GRIFO-E RIS Launcher desktop shortcut:

The Launcher opens a dialog that permits to execute the RIS SW as detailed in §4.2.

When executing, the SW presents a user interface composed of:

- the main, textual, MMI command window;
- the Simulated Radar Display graphic window;
- optionally, if connected, a replication of the MMI I/O on the PC serial communication port, that should be connected to a video terminal emulator compatible with legacy DEC VT100 video-terminal.

4.1.1.1 MMI Command window

The MMI command window:

- process user input from the PC keyboard and external video-terminal keyboard;
- display textual results, replicating them on the external video-terminal display.

The recommended command window dimension and external video-terminal window dimension is 132 columns per 32 rows (characters). By default, the local PC console windows is automatically resized if it is little then expected (see also §4.2)

The MMI display the prompt "**@#>**" when it is ready to accept command. The user can insert command as string terminated by the keyboard [ENTER] key:

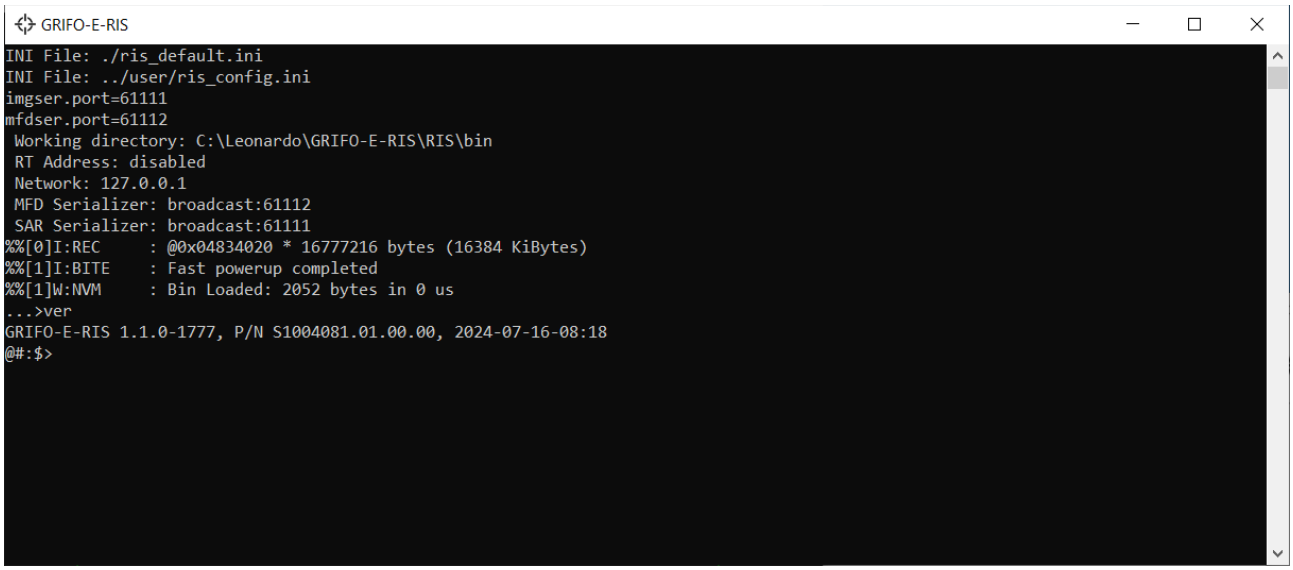


Figure 4-1: MMI I/O window

4.1.1.2 Radar Display window

The Simulated Display windows have the dimension of 640x484 pixels containing a replication of the Radar MFD (484x484 pixels), decorated with additional information about the Radar Status (that are not present on the real MFD serialized on the Ethernet).

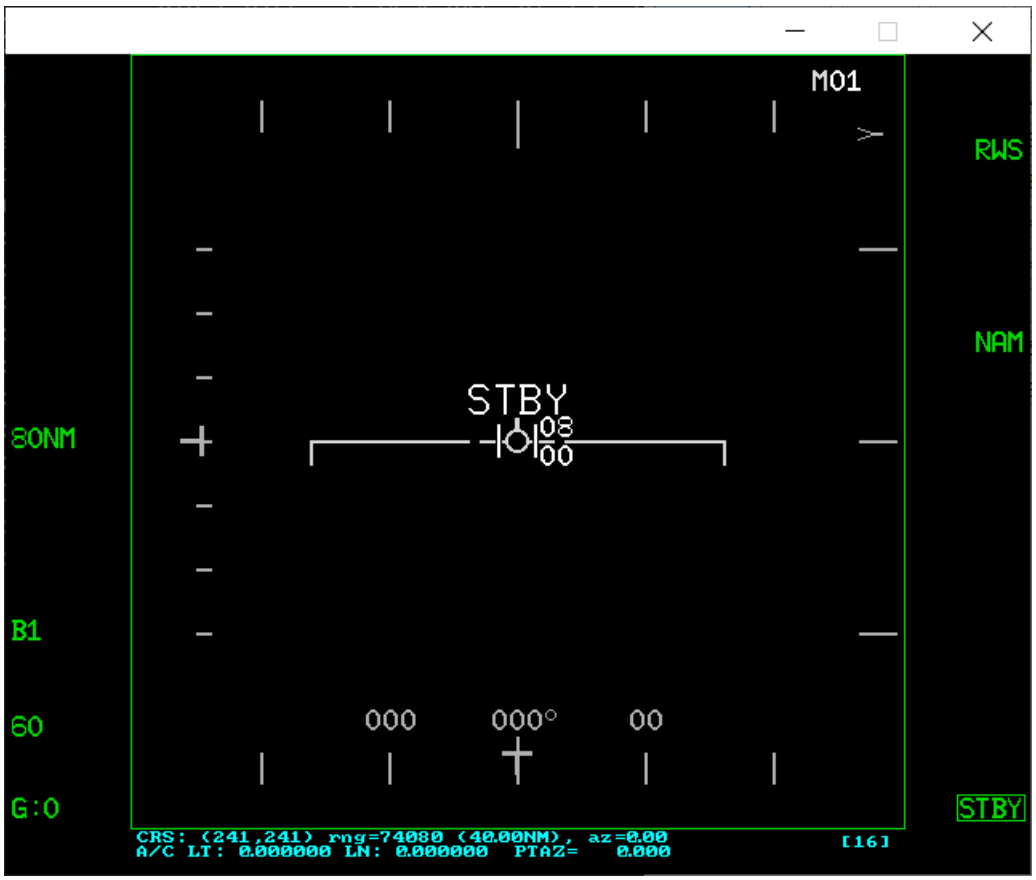


Figure 4-2: Simulated Radar display window

4.1.2 Access control

The RIS executes with hosting operating system user privilege and it does not provide by itself any access control capabilities.

Access control is delegated to the hosting PC configuration (including Ethernet I/F, firewall and any other access control capabilities built-in in the hosting operative system).

The RIS PC is intended to be used only in a protected environment and it does not include antivirus software. In particular:

- The user shall always log in with the Windows user "RIS" credential
- the PC Ethernet port shall not be connected to public network. It shall be connected, directly or through a dedicated Gigabit switch, only to a Missions Computer or a secure machine;
- any USB memory device, connected to the PC USB port, shall be verified against virus prior to insertion.

The RIS PC is preconfigured with two Windows 10 users:

- RIS: the main user, intended for all the RIS PC use, but for exceptional maintenance activity.
- Admin: Administrator user, intended to be used only in the case of exceptional maintenance activity

Both users are not password protected.

Every input/output data access, in any form, is submitted to the same restriction applicable to the RIS software and cannot be divulgated in any form without the agreement of Leonardo.

These restrictions include:

- display images;
- input and output to/from the MMI, both on the Windows console and optional serial port;
- every run-time created files (such as MMI logs and processing data recorder).
- Ethernet traffic
-

This information can be recorded or copied only to produce reports to be submitted to Leonardo.

Local back up of RIS files on the RIS PC are always permitted (for instance, in case of SW update).

4.1.3 Installation and setup

The RIS-PC comes pre-configured with the RIS-SW already installed.

In case of software update, the RIS-SW is released as a Microsoft Windows installer that automatically install the software and configure Windows desktop short cuts and menu item. The installation procedure does not require any user choice.

The installer executable is:

GRIFO-E-RIS-Installer.exe

And shall be executed only with the credential of the RIS preconfigured Window user.

Installation procedure:

1. Using another support PC:
   a. Copy the installer executable on an empty USB memory device
   b. Check against virus the USB memory device
2. Insert the USB device in the RIS-PC
3. Log-in with the RIS user credentials
4. Execute the installer executable, following the displayed instruction

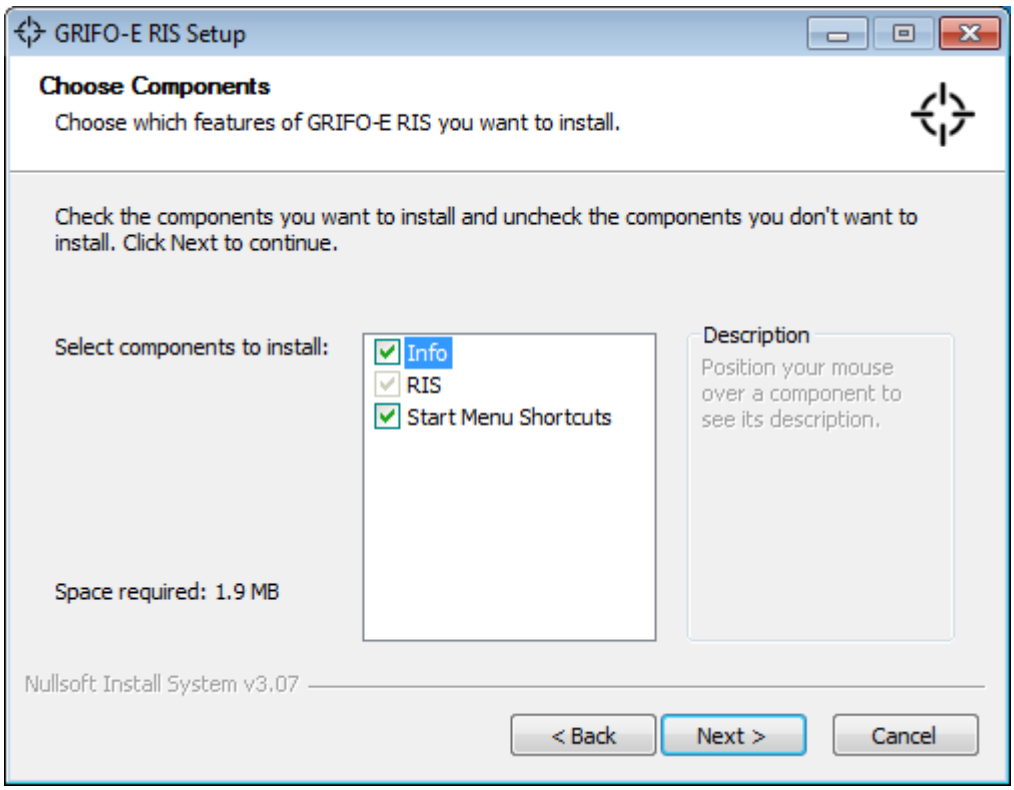


Figure 4-3: First installer screen

The first screen proposed during installation displays the components that will be installed. Leave everything selected and press the <NEXT> key (See Figure 4-3)

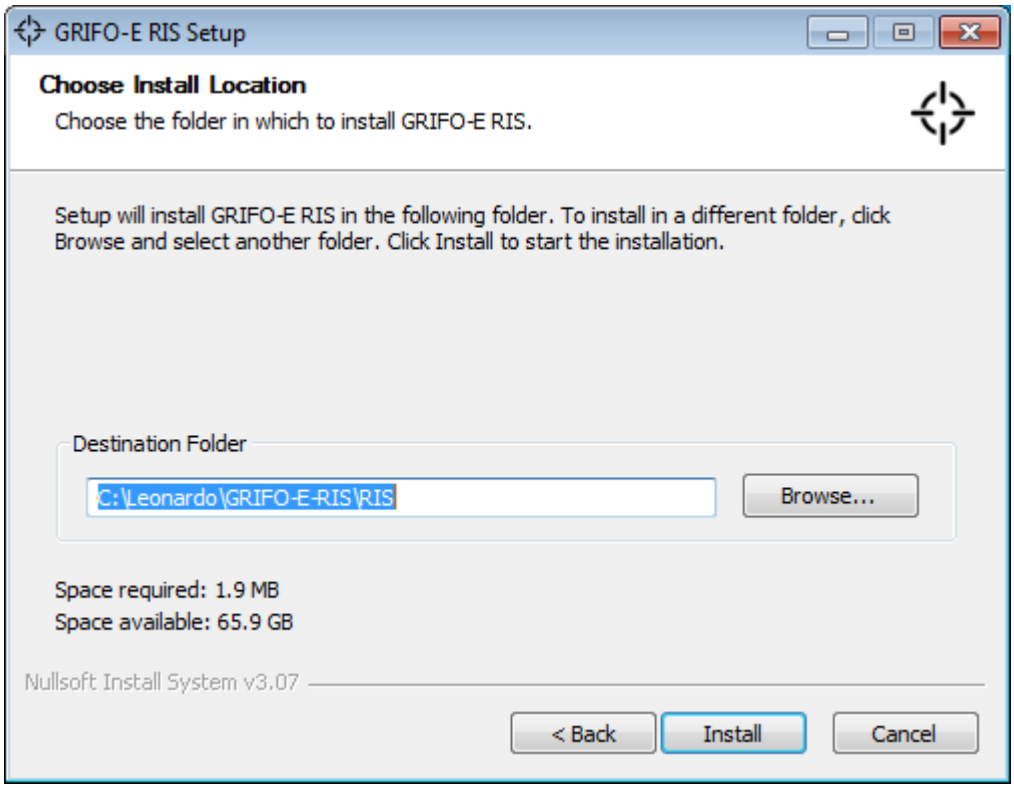


Figure 4-4: Choose install folder

In the next screen, the folder in which to install the software is proposed. Leave as default and press the <INSTALL> key (see Figure 4-4).

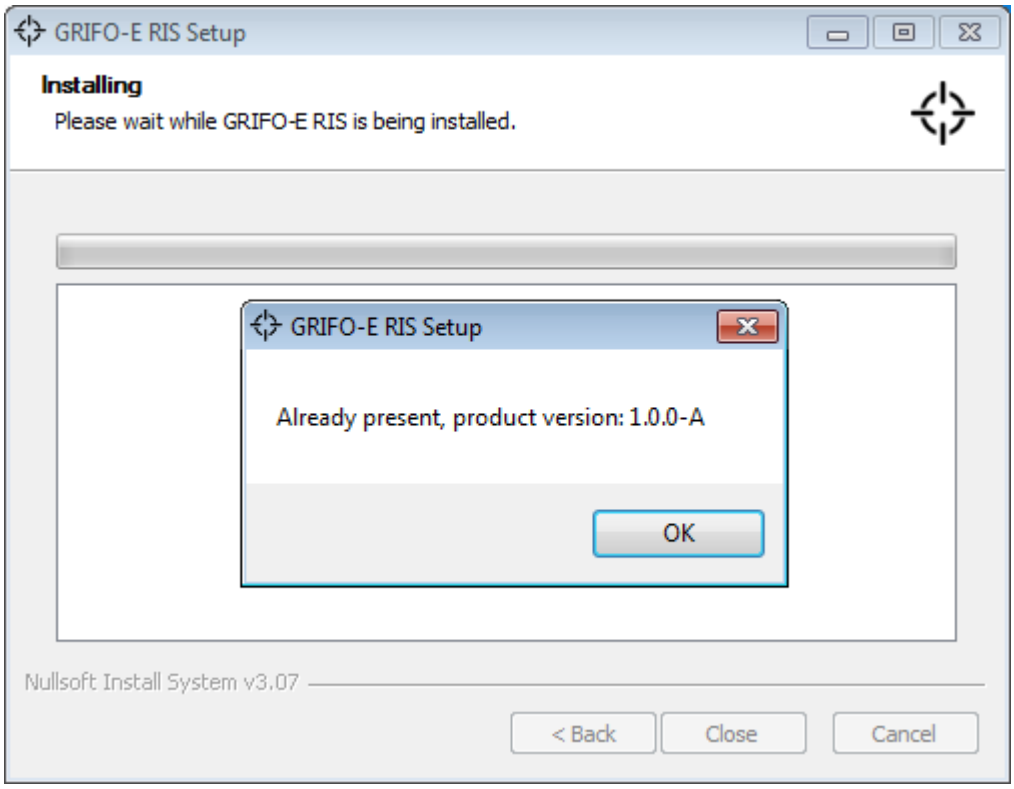


Figure 4-5: check if the software is present

The installer checks if a previous version of the software is already installed, and if so, tells you which version is already present. If no version is installed, it indicates that the current version will be installed (see Figure 4-5).

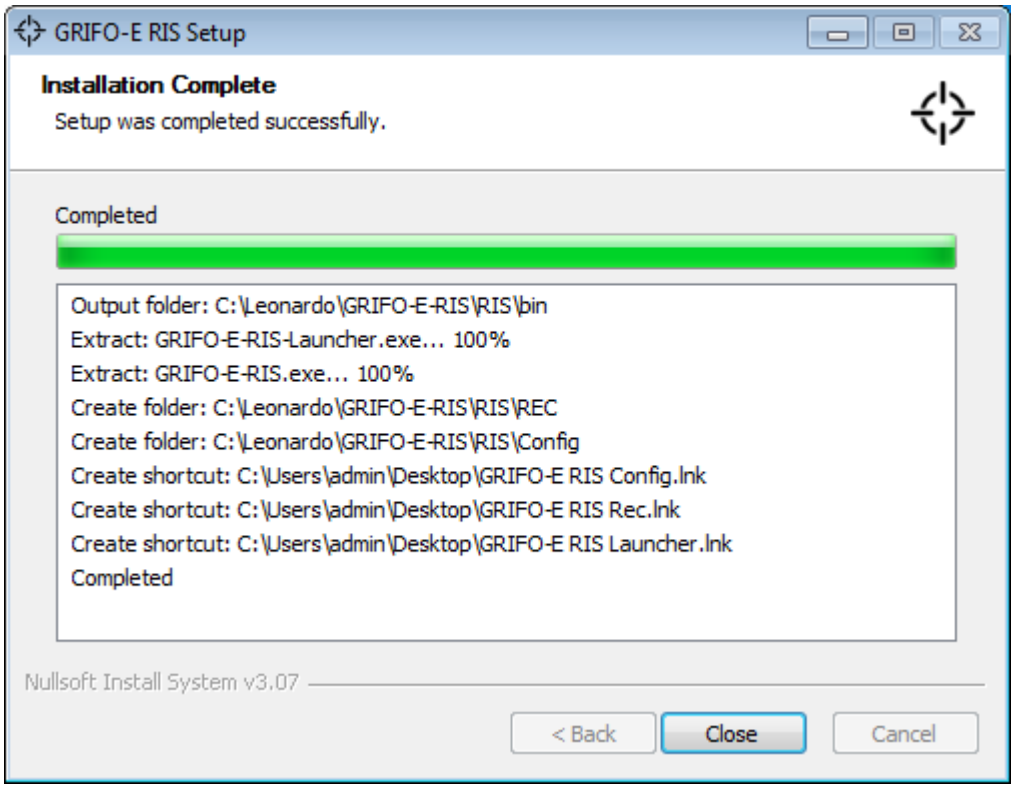


Figure 4-6: report of installer

The last screen summarizes what is installed (see Figure 4-6).

After the installation, the SW can be launched by means of the desktop short cuts "GRIFO-E RIS Launcher".

The installer does not alter the Windows configuration (with the exception of Windows user desktop shortcut). In particular:

- Does not create user
- Does not configure the Ethernet I/F properties
- Does not require privileged execution level (execute at user level privilege)

WARNING: The installation procedure does not execute any back-up of the previous installation. It is recommended to make a back-up of the previous version and every RIS produced files by copy the default installation directory C:\Leonardo\GRIFO-E-RIS\RIS:

- Open Windows Explorer and select directory C:\Leonardo\GRIFO-E-RIS
- Select the folder RIS (left single click)
- Press [**CTRL+C]**
- Press **[CTRL+V]**
- Windows will automatically create a new directory, named in such a way to not conflict with already existing copy.

4.2 Initiating a session

To initiate a session, double-click on the Desktop shortcut GRIFO-E RIS Launcher

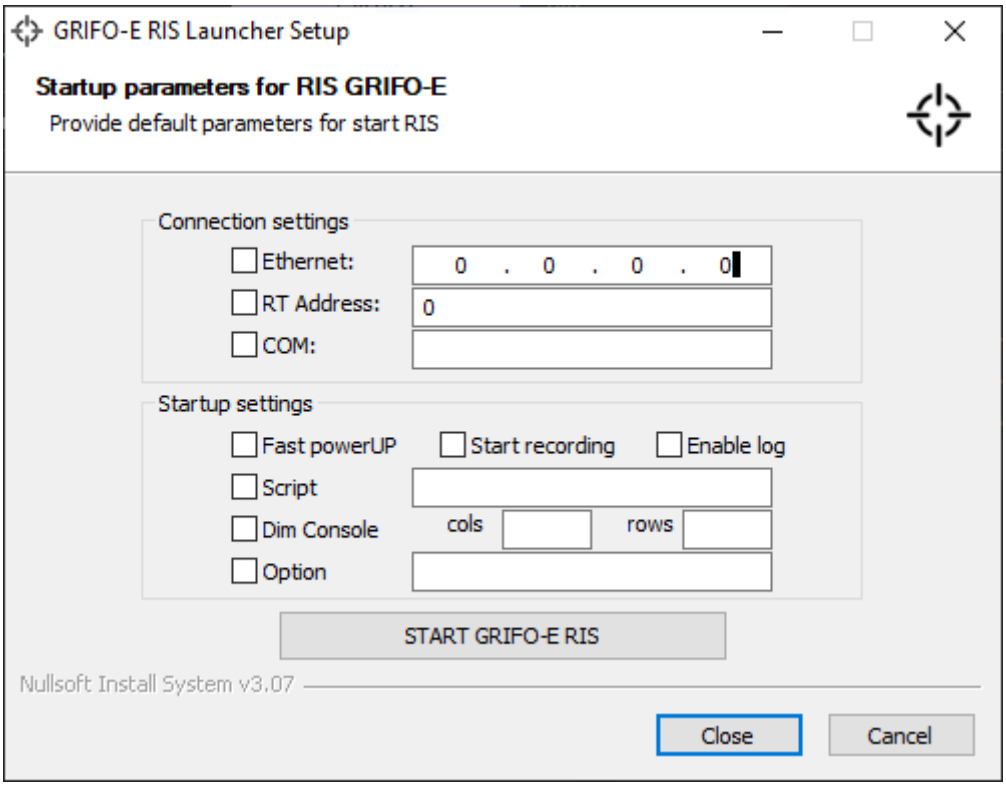


Figure 4-7: start launcher

And this window will appear where you can change the program launch parameters for GRIFO-E RIS, in the same way described in the paragraph §0.

Once the correct parameters have been entered, by pressing the "Start GRIFO-E RIS" key the program execution is launched, passing it the correct configuration in the correct format.

The GRIFO-E RIS can be also manually launched form the Windows command console (see §0).

The default Ethernet IPv4 address is **192.168.2.142**. The connections parameters should not be changed but for experimental purpose.

The Ethernet IPv4 **shall match the IPv4 address of the Ethernet of the RIS-PC**, managed by Windows 10.

For instance, if the user changes the Windows 10 IPv4 address to "192.168.1.142", the RIS shall be launched with Connection Settings Ethernet address 192.168.1.142 and Ethernet IP override checked.

For example, if you want to run the RIS execution with the following parameters, just specify them in the installer interface in this mood and press the <START GRIFO-E RIS> key (see Figure 4-8).

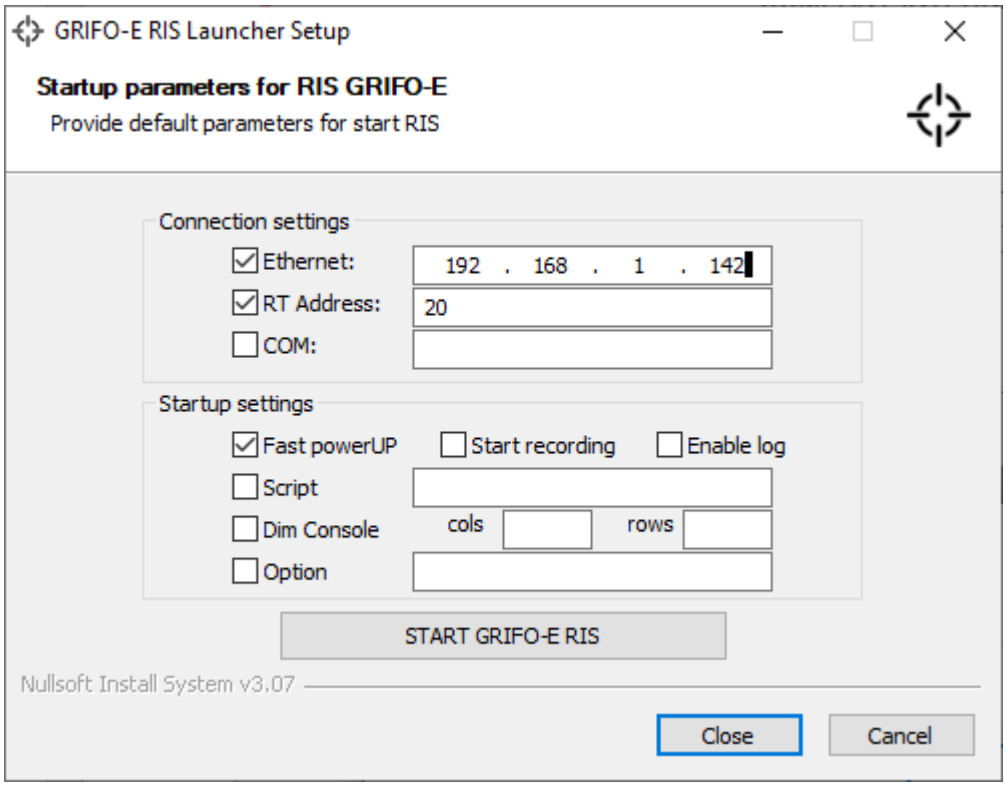


Figure 4-8: example of configuration

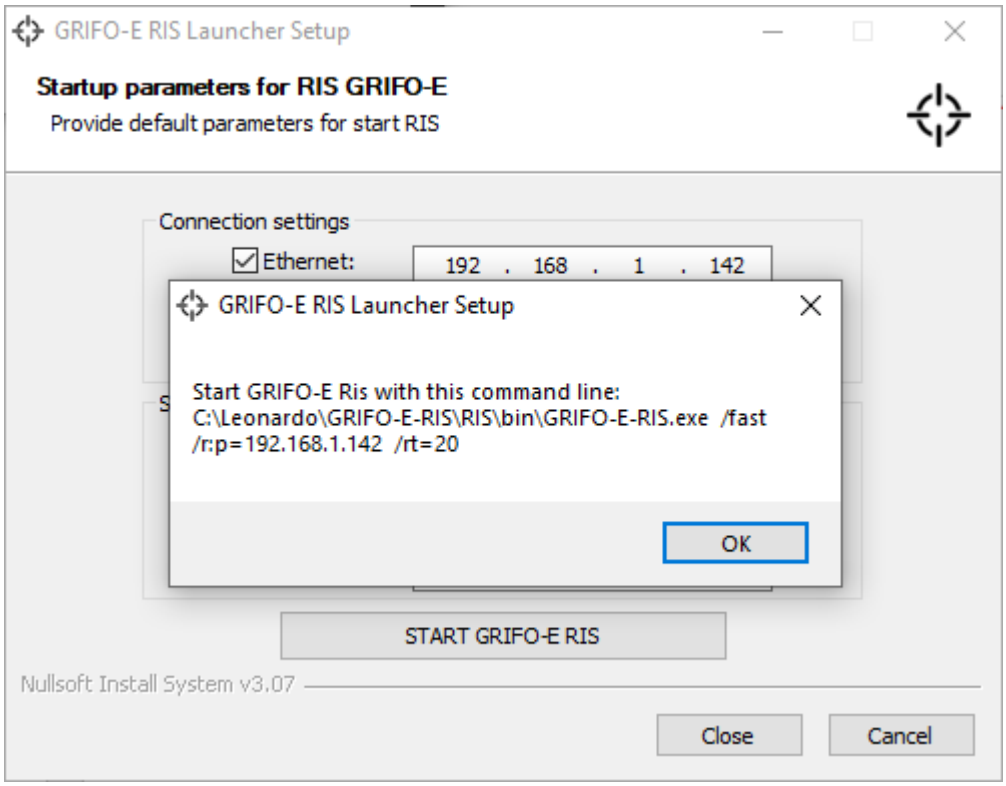


Figure 4-9: example command line

The command line (see Figure 4-9) used to launch the GRIFO-E RIS is displayed and its execution is also launched.

It is possible to use the "Option" field by inserting all those possible options of the Table 4-1, but which do not have a flag or a specific field in the launcher

4.2.1 Command line parameters

The software can be launched by means of:

- From Windows Explorer: double click on the executable **C:\LeonardoCompany\GRIFO-E-RIS\RIS\bin\GRIFO-E-RIS.exe**
- Form Windows console (cmd.exe):
  - **cd C:\LeonardoCompany\GRIFO-E-RIS\RIS\bin\GRIFO-E-RIS.exe**
  - **start GRIFO-E-RIS.exe** (optionally followed by command parameters)

When manually launched form Windows console, the RIS accept the following parameters (square bracket means optional):

Command line parameter	Description
/rt=<rtaddress>	Override the default MIL-STD-1553 Remote Terminal Address (default is 20)
/fast	Fast power-up (skip PBIT)
/rec	Immediately start recording.
/exe=<script>	Inject an MMI script at start-up, where <script> is the filename
/log	Log MMI I/O on file ris_log.txt (saved in the execution current directory)
/cdx=<n>	Set the console window columns (2)
/cdy=<n>	Set the console window rows (2)
/r:p=xxx.xxx.xx.xx	Override the Ethernet IPv4 address, required when the Windows IP is not 192.168.2.142 or when running without an Ethernet connection (IP 127.0.0.1)

Table 4-1: Command line parameters

Examples:

- Override MIL-STD-1553 RT Address: **GRIFO-E-RIS.exe /rt=10**
- Fast power-up: **GRIFO-E-RIS.exe /fast**

The user is granted to create as many desktop short-cut as required to pre-configure the different starting scenarios. Pay attention: a RIS-SW update or reinstallation could delete any user files in the installation directory.

Notes:

(1) The RIS-PC is delivered without an RS232 port. The user is granted to add a COM port by means of an USB adapter USB to RS232

(2) <n> shall be in range 0…256. If the requested console dimension is little then the current dimension, the windows is not resized neither moved. When resized, it is also moved near to the display top-left corner. Pass zero to both dimensions to left the current console dimension and position unchanged.

4.2.1.1 Initialization Files

At startup the RIS try to read in initialization files in the following order:

- ../user/ris_config.ini
- ./ris_config.ini

The initialization files format shall be standard Windows INI text file formats.

Warning:

- the RIS shall always run with working directory equal to the RIS binary directory C:\Leonardo\GRIFO-E-RIS\RIS\bin
- the RIS performs minimal checks on the INI file formats. Errors in the INI files syntax can raise to undefined behavior

The INI file can specify the following configuration parameters:

Section	Paramter	Description
[imgser]	Generic Image serialization parameters	
	ip	MFD and SAR image receiver IP
	port	MFD and SAR image receiver port
[mfdser]	MFD specific image serialization parameters	
	ip	MFD image receiver IP
	port	MFD image receiver port
[sarser]	SAR specific image serialization parameters	
	Ip	SAR image receiver IP
	port	SAR image receiver port

Warning: the RIS does not verify the coherence between the image serializer receiver and the real PC IP address. The receiver IPv4 address shall be on the same subnetwork of the PC (that is, "192.168.2.XX").

4.2.1.2 Simulated Radar display

The RIS-PC display a Window reproducing the Radar display, with the addition of some additional information to help during system integration.

The Radar MFD 484x484 window is displayed centered on an extended 640x512 window. The extended border display information on the current Radar mode.

There additional informations are not presented in the serialized MFD image.

4.3 Stopping and suspending work

The user can terminate the execution of the software in any moment by:

- Typing key sequence **[CTRL+C]** in the MMI console window
- Terminate the execution with standard Windows commands (close the console window)

In case of data recording in progress, please stop the recorder prior to terminate the application (see §5.3.1.3).

In case of abnormal termination, the software is not able by itself to provide any relevant information to permit crash investigation. When the crash is deterministic, a step-by-step report of the inputs that generate the crash can be useful for the investigation.

The report could include:

- description of the user operations (MMI command and MIL-STD-1553 data)
- external recoding of the Ethernet
- RIS internal data recording
- RIS MMI log
- Every information supplied by a Microsoft Windows Just-in-Time debugger, if there is one installed on the hosting PC

5 GENERAL PROCESSING REFERENCE GUIDE

5.1 Capabilities

Note: This manual describes only the RIS capabilities implemented in the Man-Machine-Interface of the RIS (the Radar capabilities are out of the scope of this document).

The RIS MMI permit to access the following capabilities:

Capability	Description
General purpose commands	Commands not related to a specific capability but required as support for the other commands
Message monitor	Monitoring of the virtual MIL-STD-1553 RT and decoding of the DECD data
Target simulation	Target simulation commands
BIT failure	Simulation of BIT failure (PBIT, CBIT, IBIT)
Scripting	Script execution

5.2 Conventions

A MMI command line is a string of printable, human readable, ASCII characters composed by a command name followed by, maybe optional, parameters and qualifiers separated by ASCII character SPACE (ASCII code 0x20):

```
name parameters qualifiers
```

Where:

- **"name"** is the command name
- "qualifier" is a string prefixed by the character '/' (slash)
- "parameters" is a string representing a number or a symbolic value in accordance with the command syntax

Qualifiers are always optional and position independent (that is can be entered in any order), while parameters are always position dependent and can be:

- required. Always comes prior to optional parameters
- optional: can be omitted. Optional parameter can be followed only by other optional parameters. Since the optional parameters are also position dependent, to insert some optional parameters all the preceding parameters, both required or optional, shall be entered.
- skippable: optional parameters that can be skipped, that is non entered, before move to the next optional parameters. To be skipped, an skippable parameter shall be replaced by the placeholder character "*" (asterisk)

Parameters and qualifier value could be:

- integral signed number, expressed as a sequence of digits (or hexadecimal digits if prefixed by "0x")
- floating point number expressed as a sequence of digits, an optional decimal part separated by "." and decimal exponent prefixed by "E" or "e": 1234.5E-2
- A symbolic string, composed by alphanumeric ASCII characters

The values are strongly typed, that is the expected type depend on the specific command:

- A floating-point typed value will be converted to floating point even if the user enters a simple integral value
- An integral typed value will raise an error if a floating point value is entered
- A symbolic string will be always interpreted as string even if the value is composed only by digit. The commands will raise an error if the symbolic value in out of the acceptable symbols

In the command description, the syntax is expressed with the following convention:

- **[STRING]**: bracket bold Arial text is used to identify a field from the ICD.
- **<N>**: a required parameter
- **[N]**: an optional parameter
- **{N}**: a skippable, parameter, that is can be replaced by an asterisk (*)
- Parameters value are described as N%p, where:
  - N is the parameter name
  - %p is optional and is the type specifier and can be:
    - u: unsigned integral number (as 1234 or 0x1234)
    - s: signed integral number
    - f: floating point number
    - e: enumerator symbolic value representable as an integral value. Accept both an enumeration symbolic value (a string beginning with al alphabetic character) or an integer value (beginning with a digit)
    - s: symbolic number
  - %s is the default type, applicable when type specifier is omitted
- **[/N]**: a qualifier, always optional. It could carry a required or optional value, defined with the meta-syntax:
  - [/q=N]: required qualifier value
  - [/q[=N]]: optional qualifier value

Example: The following command description:

tgtinit <p1> {p2%f} {p3%e} [p4%u] [p5%s] [/q] [/r[=v1%u]]

means a command with the following characteristic:

filed	Meaning
tgtinit	Command name (that is the string to be send to the MMI
<p1>	A required signed integer parameter called p1
{p2%f}	A skippable floatting-point parameter called p2
{p3%e}	A skippable enumerative parameter called p3
[p4%u]	An optional unsigned integer parameter called p4
[p5%s]	An optional symbolic parameters called p5
[/q]	An optional qualifier called q, without value
[/r[=v1%u]]	An optional qualifier called r, with an optional unsigned integral value

And it can be entered as:

example	Meaning

tgtinit 1	P1=1, all other omited
tgtinit 1 2 3	P1=1, P2=2, P3=3, P4 and P5 omitted
tgtinit 1 * 3	P1=1, P2=skipped, P3=3, P4 and P5 omitted
tgtinit 1 * * 4	P1=1, P2 and P3 skipped, P4=4
tgtinit 1 * * * 5	**Error**, P4 cannot be skipped
tgtinit 1 /q	P1=1, all other omitted, qualifier "q"
tgtinit 1 /r 2	P1=1, P2=2, all other omitted, qualifier "r"
tgtinit /r 1 /q	P1=1, all other omitted, qualifier "q" and "r"
tgtinit 1 /r=A	P1=1, all other omitted, qualifier "r" value equal to enumerative symbol A (that always map to a specific integral value)
tgtinit 1 /r=1	P1=1, all other omitted, qualifier "r" value equal to integral value 1

The default values of not required parameters (optional or skippable) depend on a command-by-command basis.

Some command can be executed "in loop" to permit continuously monitor of the results values. To execute a command in loop, the command string shall be terminated by the special qualifier "**/&**". To terminate the loop the uses shall send the single ASCII character "**q**".

For instance, the command:

tgtset /&

Will continuously display the position of all the active targets.

5.3 Processing procedures

5.3.1 MMI General purpose commands

Command	Description
`ver`	Display application version
`sleep <MS%u>`	MMI Sleep for required time (milliseconds)
`recorder`	Monitor, start or stop data recording

5.3.1.1 Application version (ver)

Command: **`ver`**

Display the RIS SW version, producing a video terminal output similar to:

```
GRIFO-E-RIS 1.0.0-3548, P/N S1004081.01.00.00, 2021-11-16-08:03
```

Version number and date can vary, depending on the installed version.

5.3.1.2 MMI Sleep (sleep)

Command: **`sleep <MS%u>`**

The MMI I/O channel is frozen for <MS> milliseconds. This command could be useful in conjunction with start-up script, to delay the execution of the next command, for instance to give time to the RIS to process the first A/C data from UDP1553.
Example:

```
;Sleep 1 second
sleep 1000
;Start scenario 1, use current range scale, platform azimuth and altitude
scenario 1
```

Without the sleep command, the scenario could be initialized prior to receiving the firsts UDP1553 message, therefore using the default range scale (20NM), altitude (0) and platform azimuth (0).

5.3.1.3 Internal data recorder (recorder)

Command: `recorder [/start][/stop]`

The qualifier /start start recording:

```
recorder /start
Recorder: WAITING
  File:
  Size:          0KiB
```

The qualifier /stop stop recording:

```
recorder /stop
Recorder: INACTVIE
  File: GRIFO-E-RIS-2111160803-1.0.0-3548-21-10-16-13-25-38.rec
  Size:      11666KiB
```

Without qualifier, display the status of the data recorder, for instance:

```
Recorder: INACTIVE
  File:
  Size:          0KiB
```

When recording is active but not data has been already reorder, it displays:

```
Recorder: WAITING
  File:
  Size:          0KiB
```

During recording display the file name and the so-far recorded size:

```
Recorder: RECORDING
  File: GRIFO-E-RIS-2111160803-1.0.0-3548-21-10-16-13-25-38.rec
  Size:      11666KiB
```

The file name encodes the SW version and the date and time of the recording session start.

5.3.2 Message monitor

The MIL-STD-1553 traffic and data interpretation can be monitored with the following command:

Command	Description
`rtsts`	Display information on the Ethernet traffic
`message`	Display information about DECD messages

5.3.2.1 MIL-STD-1553 traffic monitor (rtsts)

Command: `rtsts [/m][/&]`

`/m`: does not display per message report

`/&`: display continuously (type 'q' to terminate)

Display the status of the UDP1553 traffic:

```
1: Link: ?     Speed:          0/   0  IEEE SW: 0x0000
  Dropped: Eth=       0, IP=       0
Received Datagrams:     1224
  Wrong marker      :         0
  Too short         :         0
  Wrong BC TA       :         0
Valid Packets       :      1224
  Valid messages    :     13953
  Invalid messages  :         0

RX  : received ( invalid)   ms  | TX  : transmit ( invalid)   ms
RX00:        0 (        0)    0  | TX00:       81 (        0)  200
RX01:      163 (        0)  100  | TX01:        0 (        0)    0
RX02:      408 (        0)   40  | TX02:        0 (        0)    0
RX03:      164 (        0)  100~ | TX03:        0 (        0)    0
RX04:      816 (        0)   20~ | TX04:        0 (        0)    0
RX05:      816 (        0)   20~ | TX05:        0 (        0)    0
RX06:        0 (        0)    0  | TX06:        0 (        0)    0
RX07:      163 (        0)  100~ | TX07:        0 (        0)    0
RX08:      163 (        0)  100~ | TX08:        0 (        0)    0
RX09:        0 (        0)    0  | TX09:        0 (        0)    0
RX10:        0 (        0)    0  | TX10:        0 (        0)    0
RX11:        0 (        0)    0  | TX11:      163 (        0)  100
RX12:        0 (        0)    0  | TX12:      163 (        0)  100
RX13:        0 (        0)    0  | TX13:      163 (        0)  100
RX14:        0 (        0)    0  | TX14:      817 (        0)   20~
RX15:        0 (        0)    0  | TX15:      817 (        0)   20~
RX16:        0 (        0)    0  | TX16:      163 (        0)  100~
RX17:        0 (        0)    0  | TX17:      408 (        0)   40
RX18:        0 (        0)    0  | TX18:      163 (        0)  100~
RX19:        0 (        0)    0  | TX19:      818 (        0)   20
RX20:        0 (        0)    0  | TX20:      818 (        0)   20
RX21:        0 (        0)    0  | TX21:      818 (      818)   20
RX22:        0 (        0)    0  | TX22:      818 (      818)   20
RX23:        0 (        0)    0  | TX23:      818 (      818)   20
RX24:        0 (        0)    0  | TX24:      818 (      818)   20
RX25:        0 (        0)    0  | TX25:      818 (      818)   20
RX26:        0 (        0)    0  | TX26:      818 (      818)   20
RX27:        0 (        0)    0  | TX27:      818 (      818)   20
RX28:        0 (        0)    0  | TX28:      818 (      818)   20
RX29:        0 (        0)    0  | TX29:        0 (        0)    0
RX30:       82 (        0)  200  | TX30:       82 (        0)  200~
RX31:        0 (        0)    0  | TX31:        0 (        0)    0
```

The first two lines (Link status and dropped messages are not applicable to PC version).

DECD Message monitor (message)

Command: `message <ID> [/&]`

<ID>: message identifier as per DECD (A1, A2, A3, ...)

/&: display continuously (type 'q' to terminate)

Display the decoded contents of a DECD message, using symbolic names where possible and units as per DCEC. Where relevant, some internal states are displayed (for instance, in case of transient event fields such as designation control or transition in progress). When an RX and TX message are logically correlated, they are displayed together (as in case of A2 and B7)

For instance, command **`message A2`** (or command **`message B7`**) display:

```
A2->B7:
  Radar Master Mode        :    3 [T&S                ] ->   3 [T&S                ]
  Designation Control      :    7 [NOT VALID          ] ->   3 [T&S with HPT       ]
  INT-BIT                  :    0 [NORMAL             ] ->   0 [NORMAL             ]
  STBY                     :    0 [STBY OFF           ] ->   0 [STBY OFF           ]
  FREEZE                   :    0 [OFF                ] ->   0 [OFF                ]
  Power-Up Stop            : TRUE
  Silence                  :    0 [OFF                ] ->   0 [RADIATION ON       ]
  SAR Type                 : FALSE
  RWS Submode              :    0 [NAM                ] ->   0 [NAM                ]
  SPOT function            :    0 [OFF                ] ->   0 [OFF                ]
  ACM Submode              :    0 [BORESIGHT          ] ->   0 [BORESIGHT          ]
  GM Submode               :    0 [RBM                ] ->   0 [RBM                ]
  Expand                   :    0 [NORMAL             ] ->   0 [NORMAL             ]
  Range Scale              :    1 [40NM               ] ->   1 [40NM               ]
  Number of Bars           :    0 [0                  ] ->   0 [1                  ]
  Velocity Scale           :    0 [2400Kts            ]
  Scan Width               :    0 [60(45)             ] ->   0 [60                 ]

Events:
    Transition             : FALSE            1, last duration (ms):  140
  Designation Events       :    0    7 [NOT VALID          ]
```

The last two lines display, respectively:

- Radar transition in progress status (B7), plus counter of the number of transition occurred since start-up and last transition time
- Number of designation events (A2) and last recognized event

Without parameters, the command displays a list of the accepted message acronym

5.3.3 Target simulation

The target simulation permits to simulate up to 16 targets, identified by the number from 0 to 15, using the commands:

Command	Description
tgtinit	Initialize target(s)
tgtset	Changes target(s) attributes
tgtreset	Reset target(s) to initial state (inactive, traceable, all other attribute equal zero)
tgtrng	Changes target range
tgtaz	Changes target azimuth
tgtvel	Changes target velocity
tgthead	Changes target heading
tgthalt	Changes target altitude
aclatch	Latch A/C data
acunlatch	Unlatch A/C data
pause	Pause the simulation
continue	Continue the simulation

The simulator can be in two main states:

State	Description
ACTIVE	The simulation is active: the active targets are detectable and moving in the Navigation polar frame, in accordance with the specific target parameters. This is the default state.
INVACTIVE	The simulator is inactive. All the targets are not detectable and their position is not updated, regardless of the per-target state

Table 5-1: Simulator states

Every target has its own state:

Target State	Description
ACTIVE	The target is active: it is delectable and its position is updated.
INVACTIVE	The target is inactive (default): it is not detectable and its position is not updated.

Table 5-2: Target states

And the following attributes:

Attribute	Description
TRACEABLE	When false, the target is detectable but cannot be tracked
RANGE	Range relative to A/C position
AZIMUTH	Absolute azimuth in NP-Frame
VELOCITY	Absolute velocity magnitude
HEADING	Absolute heading azimuth in NP-Frame
ALTITUDE	Absolute altitude (barometric)

Table 5-3: Target attributes

The targets can be initialized or restarted by means of command **tgtinit**, then modified by means of command **tgtset** (for instance to change the actual heading without modifying the value used to restart the target when requested).

The targets live in the navigation reference system called NP-Frame, as detailed in §6.2.

A target is initialized by **tgtinit** at the command invocation instant, using, when applicable, the actual A/C.
To initialize more than one target in the same context:

1. freeze the simulation: **pause**
2. latch the A/C data: **aclatch**
3. initialize the targets by means of multiple **tgtinit** command
4. unlatch the A/C data: **acunlatch**
5. unfreeze the simulation: **continue**

The aclatch/acunlatch critical sections protection required only if the A/C date are not constant. The pause/continue critical section protection is required only if it is required to start the all the targets trajectories at the same instance.

The following paragraphs are described the MMI commands. Detailed information on the target simulation are presented in §6.2.

5.3.3.1 Target initialization (tgtinit)

Command:

`tgtinit {<ID>} {<RNG>} {<AZ>} {<SPEED>} {<HEADING>} {<ALT>} [/r] [/s] [/t]`

All the parameters are skippable, that is replaced my character '*', so to permit to set only the desired attributes.

Parameter	Description
`<ID>`	Target ID (0…15). If omitted, the command is applied to all the active targets
`<RNG>`	Range (NM). Prefixed by '%', it is interpreted as a percentage relative to current commanded range scale. For instance: **`10`**: set range to 10NM **`%50`**: set range to half the current radar range scale
`<AZ>`	NP-frame target azimuth. Prefixed by '%', it is interpreted relative to the current platform azimuth. For instance: **`120`**: set azimuth to 120° (Navigation) **`%10`**: set azimuth to 10° plus platform azimuth
`<VELOCITY>`	Velocity (feet/s). Prefixed by '%', it is interpreted relative to the current A/C velocity (velocity=velocity+ac_velocity)
`<HEADING>`	Heading (°). Prefixed by '%', it is interpreted relative to the current platform azimuth (heading=heading+ac_platform_azimuth)
`<ALT>`	Altitude (feet). Prefixed by '%', it is interpreted relative to the current A/C altitude (alt=alt+ac_inertial_altitude)
`/r`	Restart target.
`/s`	Start the target
`/-s`	Stop the target
`/t`	Traceable
`/-t`	Not traceable

Depending on the value of **`<ID>`**, the command operates on:

- **`0..15`**: the selected target
- **`*`**: only all the active targets, if **`/s`** is not specified
- **`*`**: all the targets, if **`/s`** is specified

5.3.3.2 Target position (tgtset)

Command:

`tgtset {<ID>} {<RNG>} {<AZ>} {<SPEED>} {<HEADING>} {<ALT>} [/r] [/s] [/t]`

Command tgtset takes exactly the same parameters as **`tgtinit`**, with the difference that the values are applied only to the current position and the target is not reinitialized. When restarted, the target will reuse the values supplied in the last **`tgtinit`** command.

For instance, to change the actual speed and heading of an already started target:

```
tgtset 1 * * 200 30
```

Note that a target under track is not lost due to a position change, even if its position changes significantly, while it is lost in case of restart.

5.3.3.3 Target Reset (tgtreset)

Command: **`tgtreset <ID>`**

Reset target(s) to:

- Inactive
- Traceable
- All other attribute equal zero

<ID> select the target to be reset. If equal to -1, all the targets are reset.

5.3.3.4 Target attribute (tgtrng, tgtaz, tgtvel, tgthead, tgtalt)

Some actual target attributer can be changed with the commands:

Command	Description
`tgtrng <ID> <RANGE>`	Changes range of target number <ID>
`tgtaz <ID> <AZ>`	Changes azimuth of target number <ID>
`tgtvel <ID> <VEL>`	Changes velocity of target number <ID>
`tgthead <ID> <HEADING>`	Changes heading of target number <ID>
`tgtalt <ID> <ALT>`	Changes altitude of target number <ID>

The <ID> and values are interpreted as the corresponding attribute of the command **`tgtset`**.

5.3.3.5 A/C data latch (aclatch)

Command: **`aclatch [/r]`**

`[/r]:` release the latch

Latch (or unlatch, if followed by /r) the A/C data used to initialize the target. All the target initialization command executed after an A/C latch will use the A/C altitude and platform azimuth of the latched data, instead of the last received by the Radar.

Together with commands **`pause`** and **`continue`**, should be used to initialize a multiple target scenario using the consistent A/C altitude and platform azimuth.

5.3.3.6 Pause and continue

Commands:

`pause`

`continue`

`pause` stops the target simulation while **`continue`** restart the simulation. The Radar mode simulation is never stopped, that is when the simulation is paused, the simulator cannot detect targets with the consequence that:

- All search targets disappear in accordance with the antenna scan motion
- All tracked target will be lost (possible causing change in the Radar designation status)

5.3.3.7 Built-in scenario (scenario)

Command: **`scenario [<ID>] [/r]`**

`[<ID>]:` optional numeric ID of the built-in scenario

`[/r]:` reset and invalidate all the targets prior to reconfigure the scenario

Launch a scenario, selecting between the built-in ones. Without argument display a list of the available scenario with the identification number.

Built-in scenarios are always launched atomically, that is:

- latch the A/C data
- pause the simulation
- configure the targets
- unlatch the A/C data
- continue the simulation

Launch a scenario reinitializes only the targets preview by the scenario. The other targets are invalidated only if the command is invoked with the qualifier "/r".

5.3.3.8 Override not simulated data

The MMI permits to override the tracked target attributes that are not generated by the target simulator, using the command:

`ris.ovr.<attribute_name> = <value>`

Where **`<attribute_name>`** and **`<value>`** can be:

Attribute	Value
`acc_valid`	Acceleration validity: 0 = invalid, 1=valid
`pos_accuracy`	Position accuracy level. Can be a number from 0 to 3.
`vel_accuracy`	Velocity accuracy level. Can be a number from 0 to 3.
`acc_accuracy`	Acceleration accuracy level. Can be a number from 0 to 3.
`acc_x`	Acceleration X (1)
`acc_y`	Acceleration Y (1)
`acc_z`	Acceleration Z (1)
`stddev_x`	Standard Deviation of Position X estimate
`stddev_y`	Standard Deviation of Position Y estimate
`stddev_z`	Standard Deviation of Position Z estimate

This override does not affect in any way the simulation and there is applied to all the active targets (it is not possible to override on a per target basis). Overrides are provided to permit to test the correct data interpretation by the Mission Computer.

The command **`tgtreset`** rest to zero all the overridden attributes.

Example:

Command	Meaning
`ris.ovr.acc_x=100`	Targets acceleration overridden to 100 feet/s^2
`ris.ovr.pos_accuracy=1`	Position accuracy equal to level 1
`ris.ovr.acc_validity=1`	Targets acceleration valid
`ris.ovr.acc_validity=0`	Targets acceleration invalid (the default)

Notes:
(1) Target Acceleration (Magnitude) is calculated using Target Acceleration X, Y and Z.

5.3.4 BIT failure simulation

The RIS can simulate over temperature and simple failure at LRU level, by means of the commands:

Command	Description
bitereset	Reset all the failures and over temperature alarms
biteset	Set an LRU failure
bitclear	Clear an LRU failure

The behavior depends on the Radar state:

State	Description
PBIT or IBIT in progress	The health status and BITE report will be updated at the end on the Built-In-Test
Other mode	The health status and BITE report will be updated in accordance with the type of event: • FAIL ONSET: in case of a new failure • FAIL STOP: in case of a previous failure is cleared

In any case and in accordance with the DECD, the contents of the 1553 BITE report message shall be ignored when the BITE REPORT AVAILABLE is not active (message B6).

5.3.5 Failure reset (bitereset)

Command:

bitereset

This command takes no arguments and simple reset all the failure and over-temperature alarms.

5.3.6 Set an LRU failure (biteset)

Command:

bitereset <LRU> [/ovr>

This command set a failure in the specified LRU, or set the over temperature if the qualifier "/ovr" is present. <LRU> can be:

- LCU
- AESA
- PPS
- REP

5.3.7 Reset an LRU failure (biteclear)

Command:

```
biteclear <LRU>  [/ovr>
```

This command takes exactly the same parameters of biteset, but clear the failure.

5.3.8 Scripting capabilities

The SW has basic scripting capability, mainly designed to permit injection of targets scenarios, and can process a script in the form of simple text file containing a sequence of MMI commands lines.

A script file:

- shall begin with the line ";++++"
- shall contain only ASCII (UTF-8) printable characters
- shall has a file size less then 2Kbytes
- shall contains only MMI accepted command, separated by new-line
- can contains comment, that is lines beginning with character ";"
- should be stored under the directory C:\Leonardo\GRIFO-E-RIS\RIS\user (at last to benefit of some short-cut command available in the Radar Display Window)

To inject a script file, the user can:

- drag and drop the file onto the Radar Display Window;
- Use Radar display windows short-cuts (see §5.8.1). The user can store up to 7 custom script under the directory C:\Leonardo\GRIFO-E-RIS\RIS\user and name then scenario_1.txt, scenario_2.txt and so on up to scenario_7.txt;
- inject key sequence into the Serial communication port. In this case the SW cannot distinguish between human input and script input, so the file size check and begin marker check are not performed;
- Launch the SW with the parameters **"/exe=<filename>"** (see §4.2).

WARNING:

- SW does not perform any script validation, except for the file size and begin marker. The script contents are under the responsibility of the user. Improper formed script can cause unpredictable results.
- The script is processed only when the MMI is ready to accept new command (display the prompt on the MMI display), The script is not correctly processed when injected during a command executed in loop. Prior to inject a script, be sure to wait the termination of an in progress command (or another script injection).
- Due to the above limitation, please never use loop command in script

5.4 Related processing

The RIS-SW store some internal configuration data in the file:

C:\Leonardo\GRIFO-E-RIS\RIS\Config\grifo_stored_config.bin

If not already present, this file is automatically created at the first execution and updated in background when Radar stored parameters and BITE results are updated.

All other processes are always related to user commands:

- Internal data recording is created under directory C:\Leonardo\GRIFO-E-RIS\RIS\REC, named as GRIFO-E-RIS-<VER>-<TIMENOW>.rec, where <VER> identify the SW version and <TIMENOW> the file creation time.
- MMI log are always named C:\Leonardo\GRIFO-E-RIS\RIS\REC\ris_log.txt. Note that a log session will delete the information saved in previous sessions.

The user has the responsibility of monitor the hosting PC file system usage and the size of the recorded files to prevent overrun of the disk space.

WARNING: if the SW is not launched from the default directory C:\Leonardo\GRIFO-E-RIS\RIS\bin, the files will be created relative to the current execution directory, that is:

- Configuration data: ..\Config. if the folder ..\Config does not already exist, the file grifo_stored_config.bin is created under the current directory.
- Recorded data and log: ..\REC. If the folder ..\REC does not exist, the file are not created and the information are lost.

5.5 Data backup

The RIS-SW does not require any operative data backup procedure and normally never write to existing files, with the exception of its store configuration file and MMI log file.

To prevent data loss, the user should copy on an external device recorded data and MMI log file of interest (from directory C:\Leonardo\GRIFO-E-RIS\RIS\REC).

The configuration data file is automatically recreated by the SW. TO restore a specific configuration, the Mission Computer shall resend all the parameters of interest (see [DECD]).

5.6 Recovery from errors, malfunctions, and emergencies

The RIS does not have any errors, malfunctions or emergencies recovery procedure.
Nevertheless, in case of SW crash occurring immediately at the application launch, try:

1. Launch the application without parameters
2. Remove the configuration file C:\Leonardo\GRIFO-E-RIS\RIS\ grifo_stored_config.bin
3. Remove the Ethernet connector

5.7 Messages

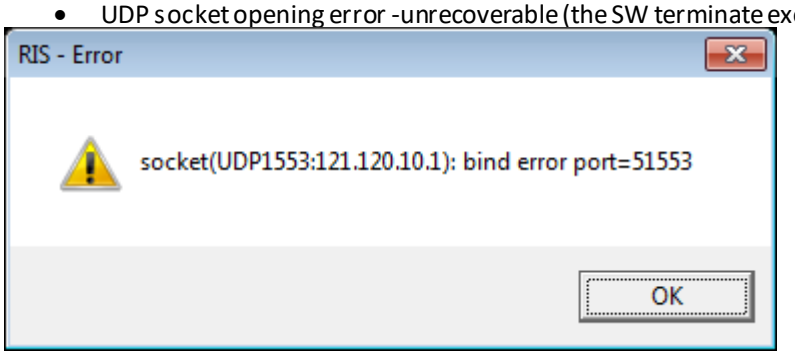
5.7.1 RIS Executable error messages

At the invocation, in case of invalid parameters the RIS exit with error code 1, after displaying the reason of the error:

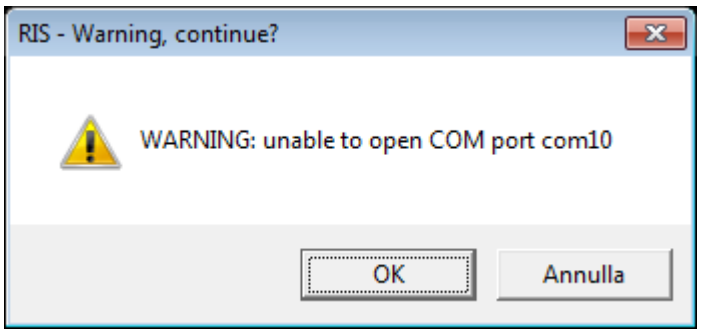
Message	Meaning
`UDP1553: invalid port`	Invalid port specified with parameter "/1553"
`UDP1553: invalid IP address`	Invalid IP address specified with parameter "/1553"
`Invalid COM port`	Invalid COM port specified with parameters "/com"
`Invalid script`	Invalid script file specified with parameters "/exe"

Some run-time errors can be detected only in later stage of execution and will be signaled both by a console message and a pop-up window:

- UDP socket opening error -unrecoverable (the SW terminate execution)



- COM port access error - the SW does not terminate the execution but stop I/O on the COM port and prompt the user to request to continue or terminate



5.7.2 Radar Display window message

The graphic presentation (on the Display window) display a RIS message to highlight some invalidity or special condition that affect the normal operation of the Radar.

The message is presented over imposing s short, cyan colored string in the upper-right corner:

Message	Meaning
MC FAIL	Total or essential missing MIL-STD-1553 traffic (mainly message A2)
INU FAIL	Missing MIL-STD-1553 A5 message or invalid time-tag
SIL	Silence commanded by means of message A2 (simulated targets cannot be detected). Displayed only when the Radar i snot already in silence due to other reason (for instance, standby or WoW)
RAD	Radiation
WOW	Weight-on-Wheels active
INV MODE	Invalid mode commanded (in message A2)

5.8 Quick-reference guide

The user can display the list of available commands with of command **`? ris:`**

```
@#:$>? ris
  message         : Show 1553 message
  ethsts          : Show Ethernet status
  scenario        : manage scenarios
  tgtinit         : manipulate targets
  tgtset          : manipulate targets
  aclatch         : manipulate targets
  pause           : manipulate targets
  continue        : manipulate targets
  untraceable     : mark target as untraceable
  recorder        : start/sotp recording
  wow             : simulate WoW
  sleep           : script sleep
```

The use can display a specific command help typing the command name followed by /?:

```
@#:$>tgtinit/?
target: manipulate targets
  id              : target id
  rng             : range (NM)
  az              : azimuth (deg)
  speed           : speed (feet/s)
  heading         : heading (deg)
  altitude        : altitude (feet)
  /r              : restart
  /s              : stop
  /p              : set only position
```

Where the string `@#:$>` is the RIS MMI command prompt.

5.8.1 Radar Display window

The simulated Radar display windows make available a context menu (right-click) with the following commands:

Menu item	Description
Targets restart	Restart all the targets
Targets running/stop	Toggle targets simulation ON/OF
WoW	Toggle simulated Avionics discrete Weight-on-Wheels
Data Recording	Toggle Data recording ON/OFF
Open/close log file	Toggle recording of MMI operation onto textual log file
Scenarios	Display a sub-menu to inject predefine scenarios and custom scenarios: • Reset: invalidate all the targets • A single target at azimuth 0, relative altitude 0, range equal to 3/4 of the current range scale • Ladder: 16 target scenario, uniformly distributed in the current scan volume and decreasing range from left to right • Custom 1 to Custom 7: custom script short-cuts (1)

Display	Enable or disable (toggle) the presentation of:
	• Mode labels (the same required in case of FCR BACK-UP mode)
	• Extra informational messages on the RIS operation (displayed in cyan color):
	  ○ Antenna body caret: show the antenna body position
	  ○ Number of active targets (3)
	  ○ LOCK-N and REJECT events as transient label (that disapper after 2 seconds)
	  ○ Tracked target identifiers (4)

Open script folder	Launch an instance of Windows Explorer on the user folder C:\Leonardo\GRIFO-E-RIS\RIS\user (2)
Open Data folder	Launch an instance of Windows Explorer on the data recorder folder C:\Leonardo\GRIFO-E-RIS\RIS\REC (2)

Notes:

1. custom scenarios short-cuts use relative file name and are fetched form the directory "..\user", that should always resolve to C:\Leonardo\GRIFO-E-RIS\RIS\user
2. "Open folder" command use absolute path names. The user directory used in (1) could be different
3. In the bottom-left corner, the RIS overlay the string “RIS: <T>/<N> (<HPT>) targets, where <T> is the number of targets under track, <N> is the number of active target and <HPT> is the internal Target ID of the HPT (-1 when not valid). In T&S, the string is followed by “HPT AUTO” when the hpt selection is automatic, or “HPT MANUAL” when the hpt selection is controlled by Mission Computer
4. Over the track target symbol appears the string “<TID>/<RID>”, where <TID> is the unique track ID used in the 1553 track messages and <RID> is the internal simulator target ID

The display window support "drag and drop" of MMI script files.

6 NOTES

6.1 Acronyms

Acronym	Meaning
RIS	Radar Interface Simulator
SW	Software (always referenced to the RIS software)
PC	Personal computer
RIS-PC	RIS WINTEL Personal Computer
INU	Inertial Navigation Unit
BIT	Built-In Test
IBIT	Interruptive BIT
WoW	Weight-on-Wheel
NP	Polar Navigation reference system
DECD	Data Exchange Control Document
An	Where n is a number between 1 and 5: identified MIL-STD-1553 RX message as per DECD
Bn	Where n is a number between 1 and 5: identified MIL-STD-1553 TX message as per DECD
WINTEL	Wintel is the partnership of Microsoft Windows and Intel producing personal computers using Intel x86-compatible processors running Microsoft Windows. The word Wintel is a portmanteau of Windows and Intel.
FCR	Fire Control Radar (thats is the GRIFO-E Radar)
NM	Nautical Miles

6.2 Target simulation

Simulated targets are detected only if:

- the Radar is not in stand-by
- the Radar is not in silence
- the target position (range, azimuth and altitude) is inside the field of view of the antenna
- the target is active
- the simulation is active

The targets live in the NP-Frame reference system, that is Navigation polar reference system centered on the A/C, that is the range is relative to the A/C, while all other attributes are absolute.

The user can initialize (**tgtinit**) or modify (**tgtset**) target(s) attribute relatively to the A/C but after the execution of the command the target(s) continue moving in the NP-Frame:

Attribute	Absolute init/set	Relative (parameter prefixed by '%')
RANGE	Relative to A/C	Percentage of the current range scale RANGE=RANGE/100+range_scale
AZIMUTH	Absolute (NP-Frame)	Relative to platform azimuth: HEADING=HEADING+platform_azimuth
VELOCITY	Absolute (orientation relative to **HEADING**)	Relative to A/C velocity: VELOCITY=VELOCITY+(sqrt(vx^2+vy^2)) Where VX and VY are the A/C Velocity X and Y
HEADING	Absolute (NP-Frame)	Relative to platform azimuth: HEADING=HEADING+platform_azimuth
ALTITUDE	Absolute (inertial)	Relative to A/C altitude: ALTITUDE=ALTITUDE+intertial_altitude

The targets trajectories are updated every 10ms, by integrating the targets relevant attribute and A/C motion:

1. Target NP-Frame (RANGE,AZIMUTH) position is converted to rectangular N-Frame (see [DCCD])
2. Target velocity vector (HEADING,VELOCITY) is converted to rectangular N-Frame
3. Target rectangular position is updated integrating A/C N-Frame velocity and target rectangular velocity
4. Target rectangular position is converted back to polar NP-Frame
5. Range rate is calculated as difference between old polar range and actual polar range

A target is detected when it falls inside the filed-of-view of the Radar scan volume.

6.2.1 NP-Frame: Target Reference system

The targets position is computed in the Navigation Polar frame (NP-Frame), centered at the force origin of the INU (left-oriented, see [ICD]§6.2.3), that is:

- Target $X = \text{range} \cdot \cos AZIMUTH$
- Target $Y = \text{range} \cdot \sin AZIMUTH$
- Target $VX = speed * \cos HEADING$
- Target $VY = speed * \sin HEADING$

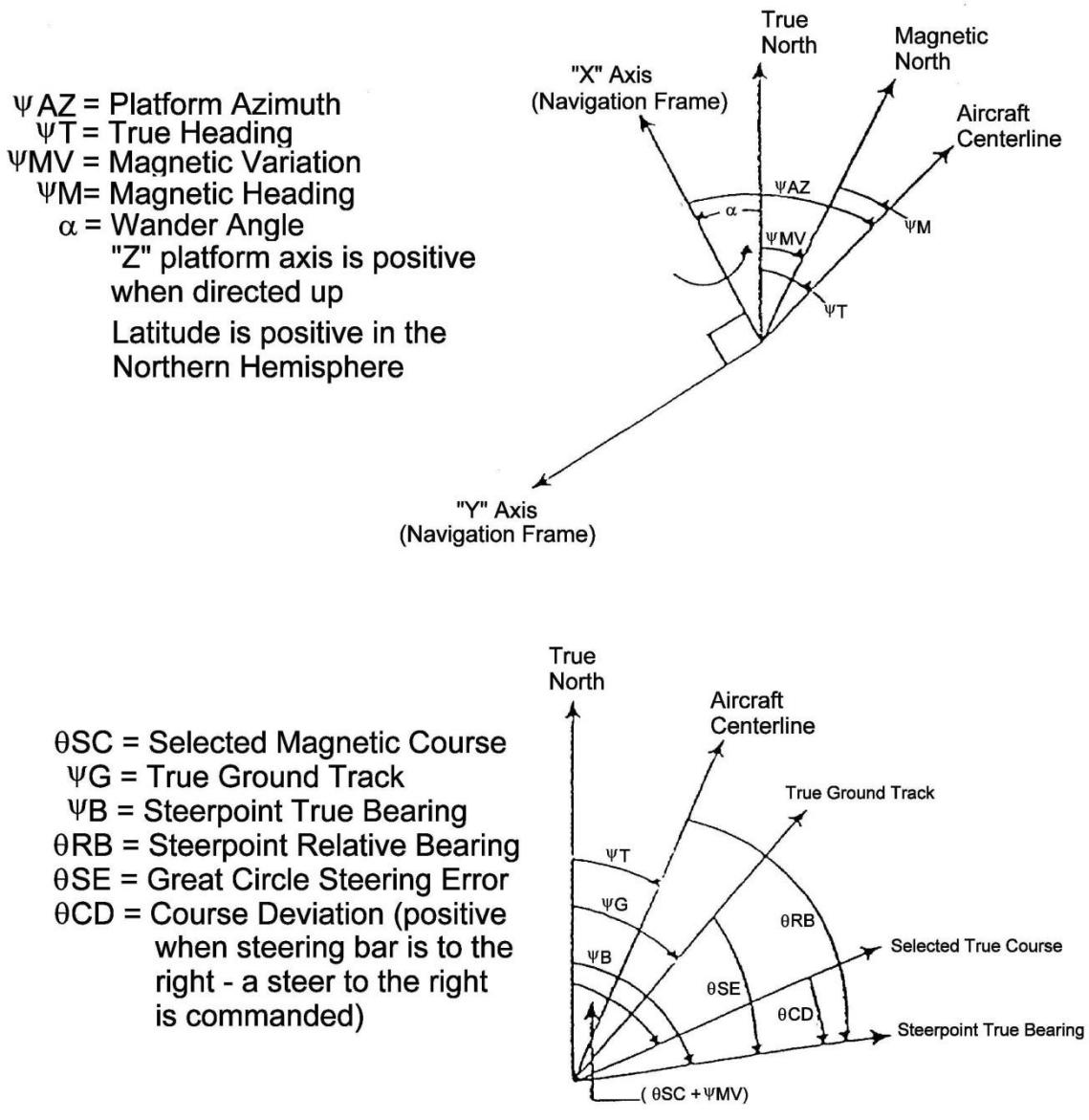


Figure 6-1: NP-Frame Angle definition

6.2.2 Targets ID

Targets are identified by an internal identifier between 0 and 15 (inclusive), permitting control and simulation of up to 16 targets.

When a target became a track, it is associated to a unique track identified (0…255) and to a 1553 track message (B5, B4, B11, B12, B13, B14, B15, B16, B17, B18). When a track is replaced by another target (for instance in T&S when a higher priority target is detected), a new unique track identifier is generated and the track id in the track message is changed in accordance.

6.2.3 Usage examples

Inject a target at navigation azimuth zero, 3/4 of the current range scale, relative velocity 0, heading north:

```
tgtinit 0 %75 0 0 %0 0 0 /s
```

Modify the target trajectory to go 45° right, a bit faster than the A/C:

```
tgtset 0 * * %100 45 0
```

Now override acceleration and accuracy:

```
ris.ovr.acc_x=100
ris.ovr.acc_valid=1
ris.ovr.acc_accuracy=2
```

And make it not traceable:

```
tgtset 0 /-t
```

Then traceable:

```
tgtset 0 /t
```

And now restart it

```
tgtinit * /r
```

Display "in loop" target position:

```
tgtset /&
```

Now type "**q**" to terminate the loop.

6.3 Video terminal character set and control functions

The MMI is based on the ISO/IEC 646 7-bit coded character set (a.k.a. ASCII, see [ASCII]), corresponding to the standard ECMA-48 (Control Functions for Coded Character set, Fifth edition – June 1991, European Computer Manufacturers Association).

For advance output formatting and keyboard interactions, the MMI is based on the standard ISOC/IEC 6429 control functions (a.k.a. as ANSI escape sequences).

The MMI use a limited subset of the standard control functions.

Input accepted control functions:

ASCII	ANSI	Meaning	VT100 Keyboard
EXT	N/A	End-of-Text	<CTRL+C>
BS	N/A		<BACK SPACE>
DEL	N/A		<DELETE>
DC3	N/A	XOFF – pause transmission	<CTRL+S>
DC1	N/A	XON – resume transmission	<CTRL-Q>
HT	N/A	Horizontal tabulation	<TAB>
	CSI0A	CUU: Cursor up (only one vertical cell is accepted)	
	CSI0B	CUD: Cursor down (only one vertical cell is accepted)	
	CSI0C	CUF: Cursor forward (only one vertical cell is accepted)	
	CSI0D	CUB: Cursor backward (only one vertical cell is accepted)	

6.4 Hosting PC Configuration

6.4.1 Third-part software

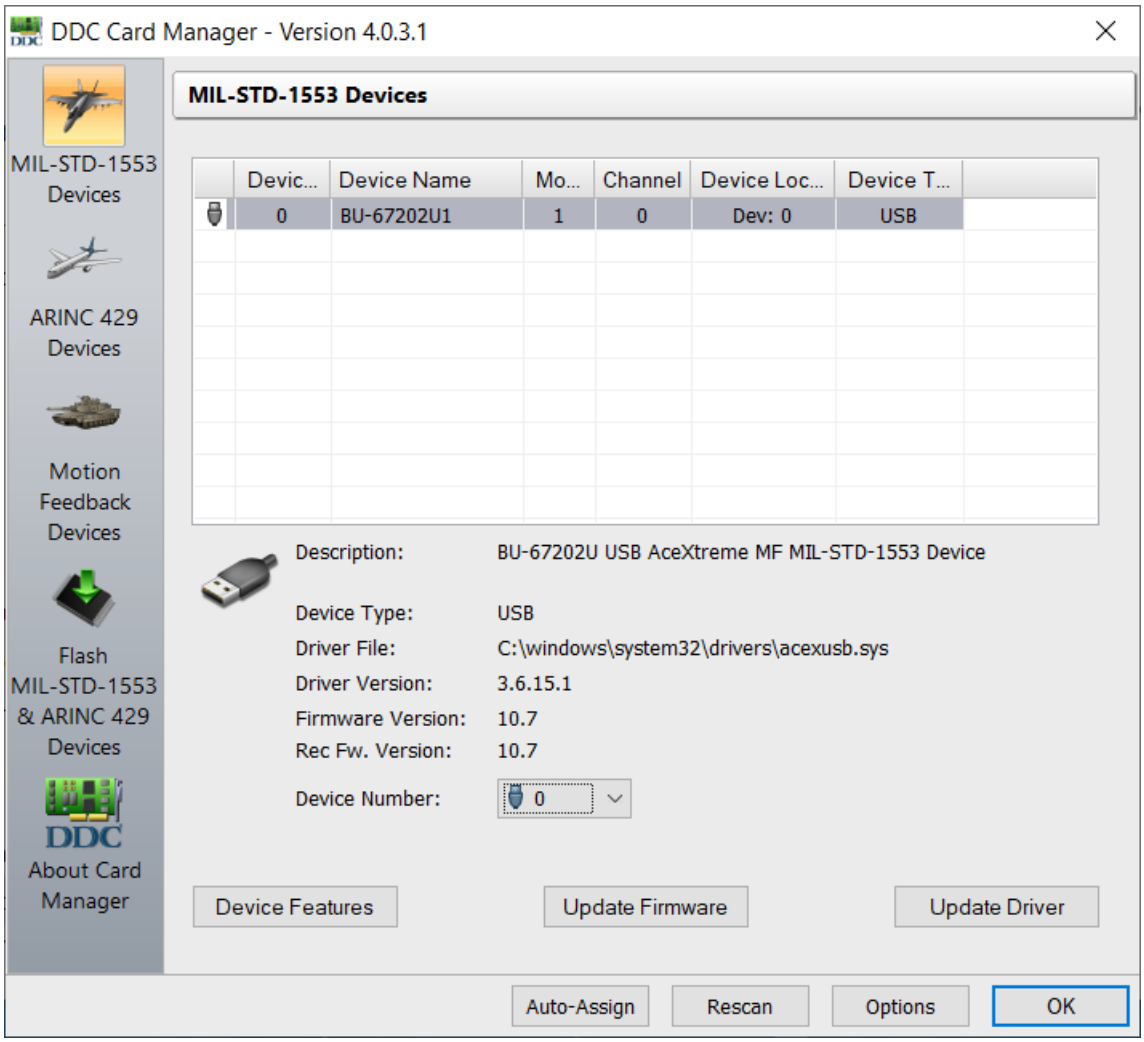
The RIS PC is released with a preinstalled version of free version of Wireshark debugger.

Licenses information can be accessed via the About menu of Wireshark.

6.4.2 MIL-STD-1553 Device

The RIS PC is released with a preinstalled version of DDC (Data Device Corporation, http::/ https://www.ddc-web.com/en) device driver BU-69092S0, version 3.6.15 for the DDC MIL-STD-1553 USB board P/N BU-67202U100L-CA0.

The BU-67202U100L shall be assigned the device number zero, by means of the DDC Control Panel (that requires the Administrator rights).



BU-67202U100L Connection:

- Connect the USB cable Type A female connector to one of the USB connector of the PC
- Connect the USB cable Type Mini B connector to the BU-67202U100L USB "primary" connector (the central one)
- Connect the BU-67202U100L Mil-STD-1553 channel CH-1A to the MIL-STD-1553 network with a proper MIL-STD-1553 cable (not supplied with the RIS)

Troubleshooting:

- The RIS SW depends on the DDC DLL emacepl.dll. In case of driver installation problems, at the startup the RIS SW will display a Windows dialog error informing of the absence of the emacepl.dll file.
- In case of disconnected device or incorrect device identification number, at the startup the RIS SW will display a dialog and give the opportunity to continue or terminate the application

BU-67102Ux / BU-67202Ux