

Guidelines for setting-up and operating an EPOS GNSS data node

Related documents	EPOS-GNSS_Node_Letter.docx EPOS-GNSS_GLASS_Node_Configuration_Letter.docx Glass Node Software Installation documentation 2021 webinar GLASS package presentation
Last update	2023-11-23
Updates	<p>2023-11-23 Update of the hardware and software requirements (sec. 3.1), update on RINEX data access (sec. 3.1). info on DQMS (sec. 5 & 9) Add information on metadata-monitor in sec. 7</p> <p>2023-05-15: Most updates concern the sections 3.1 and 3.2 Addition of http/https, python 3 requirements, software mailing list, shared directory Changes the figure 1, the node mailing list Deletion of the virtual machine possibility, software coordinator e-mail</p> <p>2023-03-06: Add guideline for backup</p> <p>2022-09-19: Small corrections</p> <p>2022-07-31: Addition of section 7 about the file metadata comparator tool</p> <p>2022-02-22: Correction of broken links and refinement based on user feedback</p> <p>2021-10-21: Refinement of sections 1 to 4, and 8 Addition of sections 5 to 7</p> <p>2020-06-09: Refinement</p> <p>2019-12-13: Addition of sections 2.3 and 3.3 Important revision of sections 2 and 3</p> <p>2019-06-25: Addition of section 3 Addition of links Refinement and clarification of the terminology</p> <p>2018-03-20: Creation</p>
	Mathilde Vergnolle (EPOS-GNSS Executive Board) and Jean-Luc Menut
Contact	gnss-dgw@oca.eu





TABLE OF CONTENTS

1. Foreword	4
2. Architecture of the EPOS GLASS network	4
2.1. Introduction and terminology	4
2.2. Schematic overview	5
2.3. Solutions to make EPOS data discoverable through a GLASS node	5
3. Guidelines for hosting and managing a GLASS node	6
3.1. Requirements for hosting a GLASS node	6
3.2. Responsibilities of the node manager	7
3.3. Agreed-upon responsibilities between the node manager and the data repository manager	8
4. Procedure for adding a GLASS node to the EPOS GLASS network	8
5. Procedure for making a GLASS node operational	9
6. Local node population	10
7. Monitoring the local node	10
8. Terminology for classifying nodes as the integration process proceeds	11
9. Additional Remarks	11



1. Foreword

This document is addressed to the EPOS *Node Manager* and describes the procedure for adding and managing a new *GLASS node* in the EPOS *GLASS network*.

Please contact the EPOS *GNSS Data Gateway* (gnss-dgw@oca.eu) to go through this procedure and to validate the integration of the *GLASS node* into the *GLASS network*.

2. Architecture of the EPOS GLASS network

2.1. Introduction and terminology

The EPOS GNSS data and metadata dissemination system is based on the integration of several independent nodes through a central gateway. The EPOS GNSS *Data Gateway (DGW)* is the centralized point of access of all metadata. The exchange of metadata between the *Data Gateway* and the other *GLASS nodes* is done through synchronization of the metadata.

Each node is hosting metadata relative to the data of one or several *Data Repositories*. It is worth noting that a *GLASS node* does not host data but the associated metadata.

The data dissemination system consists of two components:

- A. Virtual layer (*GLASS network*): represented by *GLASS nodes* and the hierarchy operating on metadata
- B. Physical layer: represented by *Data Repositories/Centers* storing GNSS data

Metadata can be grouped into:

- **T0** – *GLASS virtualization architecture metadata* (node-specific) which describes the topology between the various *GLASS nodes* and *Data Repositories/Centers*
- **T1** – *GNSS station metadata* (centralized, synchronized downward)
- **T2** – *GNSS file metadata* (decentralized, synchronized upward)
- **T3** – *GNSS data Quality Control (QC) metadata* (decentralized)

The *GNSS stations metadata* are added in the network through a unique point to ensure unicity and robustness of these metadata. The *GNSS file and data quality metadata* are entered locally into the system.

Data Gateway - DGW: The top-level federated *GLASS node* for data. It centralizes *station (T1)* and *file (T2) metadata* of the EPOS *GLASS network* and manages the *GLASS network architecture metadata (T0)*.

Data Repositories: The physical locations of the data files for EPOS stations. It is the access point to the EPOS data files (<http/https> server, no ftp server).

GLASS network: The network of all the *GLASS nodes* in EPOS.

GLASS node: The server hosting the software necessary to disseminate GNSS data and metadata. It is an access point for database services accessed via an API. It stores all the metadata for the stations that belong to the node.

GLASS package: A set of tools installed at a *GLASS node*, providing the *virtual layer* for data dissemination. At each individual *GLASS node*, a subset of tools is activated according to the node functionality. Some of the tools may also be installed at *Data Repositories*.

M³G: Metadata Management and distribution system for Multiple GNSS networks. It allows to upload and validate GNSS station metadata (e.g. site logs). After validation, *M³G* transfers the GNSS *station metadata* of the EPOS stations to the EPOS *DGW*.

Node Manager: the person (or the team) managing a *GLASS node*.

2.2. Schematic overview

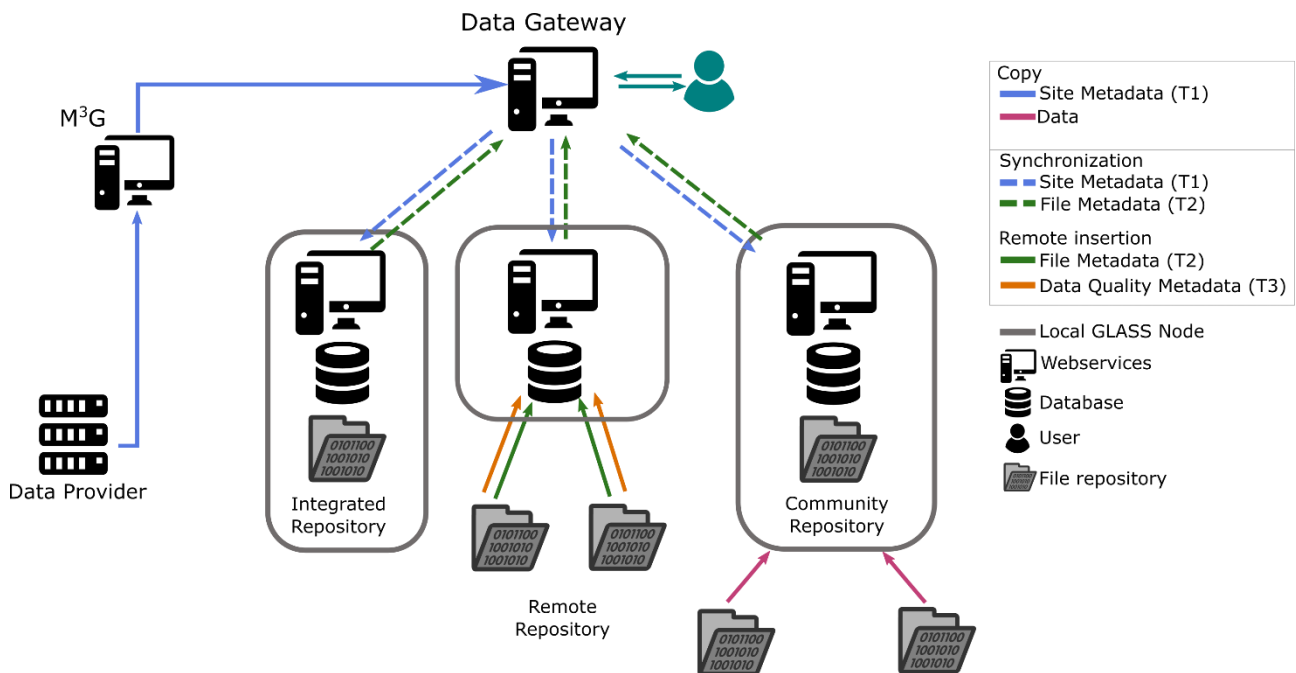


Figure 1: The flow of the GNSS metadata in the GLASS distributed network. This schema shows the three node/data repository models described in the next chapter.

2.3. Solutions to make EPOS data discoverable through a GLASS node

A data supplier can make its GNSS data discoverable to EPOS using three different data flow schemes (Figure 1).

One of them will not be presented with details here as it does not require to install a GLASS node or components of it. This is the case in which the data owner transfers the (RINEX) files to a *Community European GLASS node and Repository* (“community repository” model in Figure 1).

Then, we can distinguish between a full **GLASS node** installation (option 1) and a light installation (option 2). In both options, the data owner keeps his GNSS observation files in his local repository.

- **Option 1: GLASS node and Data Repository(ies)** hosted at the same location and managed by a single agency. Activities are centralized at the GLASS node, in particular the generation of file and QC metadata (*T2 & T3*) and the population of the node. This is the “Integrated Repository” model in Figure 1
- **Option 2: GLASS node and Data Repository(ies)** physically independent and managed by different agencies. The data repository owner and the *Node Manager* have to agree upon who is generating the *file and QC metadata (T2 & T3)* and populating the node. This is the “Remote Repository” model in Figure 1

In any case the node and the EPOS-GNSS Data infrastructure must only distribute one RINEX file for a set of data. Especially, RINEX 2 And 3 cannot be distributed simultaneously.

3. Guidelines for hosting and managing a GLASS node

3.1. Requirements for hosting a GLASS node

A *GLASS node* provides a service to the GNSS community both directly and as a part of the entire *GLASS network*.

Hosting a GLASS node requires having the technical capabilities to install and update the GLASS package and to set up the population routines adapted to the local specific environment based on the tools provided (see [the Glass Node Software Installation documentation](#) and the [2021 webinar GLASS package presentation](#)).

The other requirements to ensure a good integration of the node with the rest of the network are:

- The node cannot exist before being added to EPOS (no solution exists to include such node).
- It is not possible to have a node sharing EPOS and non-EPOS stations.
- A GLASS node should host at least ~50 new stations. For less stations, it is preferred to host the stations in the national node of the country or in the Pan-European node. A node hosting less stations could be accepted after discussion with the EPOS GNSS Consortium Board.
- The physical server and the network infrastructure must be scaled for the expected usage, especially for the daily or sub-daily file metadata update and synchronization towards the *DGW*. The server should have at least 8 Gb of RAM and 4 cores (this is a low estimate, to be checked with *DGW*, if needed).
- The *GLASS node* must be able to run python 3.10 and install libraries through pip. It must be able to run postgresql 14.x, Payara 5.x and java 1.11 (see [the Glass Node Software Installation documentation](#)).

- The database must be directly accessible from the *DGW* (for synchronization). It means that the port of the database must be open to the *DGW* and potential intermediate nodes.
- The *GLASS* API must be accessible from at least the *M3G* and the *DGW*
- The tools generating the *file and QC metadata* do not need to be installed on the *GLASS node* but **MUST** have access to the *RINEX* files **AND** to the population API.
- The *GLASS node* must tend towards a 24/24 availability (on a best-effort basis).
- The *RINEX* data provided by the node must be accessible through a *http/https* server (**not** *ftp*)
- *RINEX* data should be accessible without authentication
- *RINEX* data should be accessible with a tool like *wget* without special configuration

Note about the national node: This is a node having vocation of disseminating data and metadata from the GNS stations of a country. A node cannot be declared national by its maintainers, but need several instates to declare planning to use this node for distributing their data and metadata.

3.2. Responsibilities of the node manager

The *Node Manager* must ensure proper operation of the *GLASS node*, especially the management of the node and its interactions with the associated *Data Repository(ies)* and the *DGW*. Once a node is fully operational, the *Node Manager* should distribute the data and metadata as soon as they are received from the *Data Repository*.

Notably, she/he is responsible for:

- Managing and monitoring the node to ensure a proper service (e.g. troubleshooting server problems and service failures as soon as possible).
- Participating to the update of the *GLASS software package*, in synchronization with all other nodes.
- Verifying that the metadata synchronization between the node and the *DGW* (and potential intermediate node, see Section 2.2) works properly.
- Keeping in contact with the *Data Repository* managers and answering their requests.
- Warning as soon as possible the *DGW* in case of service failure.
- Warning the *DGW* in case of any changes in the node: hardware modification, service impacting software updates, IP or contact changes, and consequently for updating and sending the *EPOS GNSS Node configuration letter*.
- Checking that all the expected file metadata are synchronized at the *DGW*, and that there is no difference in the file metadata between the local node and the *DGW*. The node manager must check that the file metadata are inserted in the node, that their status are greater than zero, and that the metadata are successfully synchronized (a tool is provided to help the node manager).
- Backing up the database daily.
- Monitoring the usage of the node (a procedure will be provided).
- Verifying that the disseminated files are accessible and accessible without authentication.

- Check that no duplicate file is being distributed by the node. Specifically, the file is either RINEX 2 or 3 (or later), but not both for the same station and the same day.
- Check that the spelling of the file name is consistent with the RINEX format used.
- To contact the relevant person in case a problem has been identified in the previous procedure.

The *Node Manager* must subscribe to the *GLASS network* mailing list (nodes@gnss-epos.eu) to inform and be informed of news and changes within the network. Please contact gnss-dgw@oca.eu for subscription.

3.3. Agreed-upon responsibilities between the node manager and the data repository manager

The *Node Manager* and/or the *Data Repository manager*, depending on the option chosen in Section 2.3 **Erreur ! Source du renvoi introuvable.**, should comply with the additional responsibilities listed below.

In case of option n. 2 (“Remote Repository” model, Figure 1), the *Node Manager* and the *Data Repository manager* have to agree on how to share the responsibilities, while in case of options n°1 (“Integrated Repository” model, Section 2.3), the *Node Manager* is the sole responsible for all the following tasks:

- Ensure the generation of *file and QC metadata* AND the transfer of these metadata to the node as soon as the data become available at the data repository. By default, the data repository manager is responsible, except if an alternative solution is agreed upon with the *Node Manager*.
- Ensure that the node distributes data to the EPOS *DGW* only with permission of the data owner. It can be verified via *M³G* (<https://gnss-metadata.eu/>) that the *file metadata* added to the node belong to a station proposed to EPOS and that the node has been granted permission to distribute its data. By default, this is a shared responsibility, except if an alternative arrangement for the former point is in place.
- Manage and monitor the proper generation of *file and QC metadata* and warn who is responsible to solve any problems as soon as possible. Solving some issues will require to warn the data owner. Responsibility for managing, monitoring and/or warning should be agreed upon in advance.

4. Procedure for adding a GLASS node to the EPOS GLASS network

1. Contact the **Data Gateway** (gnss-dgw@oca.eu) to check that all requirements to propose a node are fulfilled.

2. Download the “*EPOS-GNSS_Node_Letter.docx*”, supplied at https://gnss-metadata.eu/Guidelines/EPOS-GNSS_Node_Letter.docx and send the completed node document to the *M³G* (m3g@oma.be), with the *Data Gateway* in copy (gnss-dgw@oca.eu).
3. Install and configure the *GLASS package*. Run the test to check if the node works properly. See the [Glass Node Software Installation documentation](#).
4. Download the document “*EPOS-GNSS_GLASS_node_configuration_letter.docx*” ().
 - a. Collect the parameters of the *GLASS node* (name, IP, port, database name, username for the database). Although there is an entry for the password, it is more appropriate the contact the *DGW* to provide this information more safely.
 - b. Collect the information concerning the *Data Repository(ies)* linked to the *GLASS node* (name, acronym, agency information, server parameters).
 - c. Fill in the document and send it to the *Data Gateway* (gnss-dgw@oca.eu).

The *GLASS node* is now ready to be made operational.

5. Procedure for making a GLASS node operational

1. Ask *Data Repository(ies)* to add stations following the procedure described in the “*Procedure for including GNSS Stations in EPOS*” (<https://gnss-metadata.eu/Guidelines>).
2. Inform the *Data Repository(ies)* about your *GLASS node* name.
3. In contact with the *Data Gateway and Data Repository(ies)*:
 - a. Check if the database contains the *station metadata (T1)*.
 - b. Add *file metadata (T2)* and *QC metadata (T3)* for one test station using the provided tools (IndexGD - <https://gitlab.com/gpseurope/indexGD>, and RunQC - <https://gitlab.com/gpseurope/RunQC>).
 - c. Synchronize the file metadata to the DGW (using the *EPOS_Sync_System* tool https://gitlab.com/gpseurope/EPOS_Sync_System).

Once the metadata related to the test station are discoverable on the *Data Gateway*, the *GLASS node* is operational. The information concerning the *GLASS node* (in the *EPOS-GNSS_GLASS_node_configuration_letter.docx*) will be sent to the *M3G* to update the system with this new node.

4. Check the status of the files and that the synchronization worked using *fmd_comparator* (https://gitlab.com/gpseurope/fmd_comparator, section 7).
5. Populate the database with the rest of the data (section 6). Inform the *DQMS* of the population of your node

6. Local node population

The population of the local **GLASS node** is done in 4 steps:

1. Receiving the station metadata from the Data Gateway
2. Generating the file metadata and populating the database
3. Generating the QC metadata and populating the database
4. Synchronization of the file metadata to the Data Gateway

The first step is done by the DGW and the others at the local GLASS node.

The second step is done by using indexGD on the RINEX file: It generates the file metadata (the url of the files and some associated metadata) and send them to the web server FWSS, which insert the metadata into the database. The output of indexGD and the log of FWSS give information about the success or the failure of the process. The file metadata receive a status of 0 at this point. The documentation of indexGD can be found at <https://gitlab.com/gpseurope/indexGD>

The third step is done using RunQC: it generates QC metadata and send them also to FWSS to be inserted in the database. Depending on the result of the process, it changes the status field in the file metadata to -3 (error in the RINEX header), 1 (no error and no warning), 2 (small inconsistencies in the RINEX header) and 3 (metrics in the RINEX file not optimal with respect to what is expected) . The output of RunQC and the QC log of FWSS give information about the success or the failure of the process. The documentation on RunQC is available at <https://gitlab.com/gpseurope/RunQC>

The last step is done by running the synchronization process to synchronize the file metadata to the **Data Gateway**. The QC metadata are not synchronized. Only the file metadata with a status ≥ 1 are synchronized.

The tool to be used is EPOS_Sync_System. The software can be set up to display more or less information about the synchronization process, and to work either as a command line tool or a daemon. The documentation is available at https://gitlab.com/gpseurope/EPOS_Sync_System

Note: it is forbidden to add a RINEX3 and a RINEX2 file for the same station and the same day

7. Monitoring the local node

Two tools are provided to the **Node Managers** to help them monitor their node:

1. fmd_comparator (https://gitlab.com/gpseurope/fmd_comparator). It's a python tool that compares the file metadata output from the DGW and from a local node GLASS-API. An email is sent with a summary and a detailed report of the differences (along with a README to understand the results).
2. Metadata-comparator (<https://gnssdata-epos.oca.eu/metadata-comparator/>). It's a web service that compares the stations and file metadata of the DGW and the local node through the GLASS-API,

as well as display the availability of the API for both the files and stations metadata. Request access to DGW (gnss-dgw@oca.eu) when your node is added to the EPOS-GNSS node network.

8. Terminology for classifying nodes as the integration process proceeds

Each node included in EPOS-GNSS will pass through the following steps as the integration process proceeds:

- **Planned:** The DGW and M3G teams have been informed about the plan to operate a data node so that GNSS stations can already be associated to the node.
- **Under testing:** The DGW and M3G teams have received the node letters and the node is currently being installed and tested.
- **Pre-operational:** The DGW and local node teams have successfully tested the node and it is technically ready to be populated and synchronized with the DGW.
- **Operational (partly populated):** The node is partly populated (some historical data or new incoming data are still missing). All data added to the node are discoverable from the DGW.
- **Operational:** The node is populated with historical data and new incoming data, which are all discoverable from the DGW.

9. Additional Remarks

1. Intermediate *GLASS nodes*, i.e. between local *GLASS nodes* and the *DGW*, could also be installed. The procedure is mostly the same, except for synchronization. Contact the *Data Gateway* if needed.
2. A shared directory is used to distribute node meeting summaries and presentations, an action item list related to node management, and several other documents
3. Important e-mails:
 - *EPOS-GNSS Nodes* mailing list: nodes@gnss-epos.eu
 - *GLASS Software* mailing list: software@gnss-epos.eu
 - *DGW* contact: gnss-dgw@oca.eu
 - *M³G* contact: m3g@oma.be
 - *DQMS* contact : epos@oma.be



