TMPropagator

Release 3.6

Release Note

28th January, 2026

Document Signature Table

	Name	**Signature**	Date
Prepared by	C. Peat	C.Peat	28th January, 2026
Approved by	C. Peat	C.Peat	28th January, 2026

Table of Contents

1 SCOPE

The document is the release note accompanying release 3.6 of the TMPropagator. This is a major release which fixes a number of JIRA issues. The issues requested to be fixed in this delivery were specified in TMPROP-537.

2 ISSUES FIXED BY THIS RELEASE

The following JIRA issues are considered fixed by this release;

TMPROP-266	[JasonCS/AR/2066] Deleted OPS commands not flagged as "Deleted" in TC History display of TMPropagator client
TMPROP-353	[MTG/AR/5030] [TMProp] Scrolling display missing Out of limit update
TMPROP-415	MTG OPE1 server not processing data received from GW
TMPROP-471	TM Propagator fails to display TC History correctly
TMPROP-472	[EPSSG/AR/2876] TM Propagator TC History MMI look and feel not aligned with MCS TC History MMI
TMPROP-481	TM Propagator TC History does not report OBTS cmds
TMPROP-484	[EPS/AR/20357] TM Propagator: New Mimic editor causes problem with current mimic
TMPROP-497	TMProp Client should inform the users when the wrong version is being used
TMPROP-498	TMPropagator Client log should report dates in format DD/MM/YYYY
TMPROP-499	Update of TMPropagator ICD
TMPROP-504	[JasonCS/AR/2779] TM Propagator: unavailability of performance parameters for retrieval from client (version 3.4.0.)
TMPROP-543	TMProp Server's Gateway input queue size sometimes starts increasing indefinitely
TMPROP-559	Handling of source code property of EUMETSAT
TMPROP-560	Documentation of the internal interface Server - Client

TMPROP-567	[CO2M/AR/82] [OVT-1A] TmPropagator becomes unresponsive while using Variable Packets page
TMPROP-572	TM Prop client stream selector dropdown menu is too small
TMPROP-575	Missing data on TMProp MTG-I1 OPE
TMPROP-578	OBEV streams on S3 randomly disconnect due to likely data type mismatch
TMPROP-586	Client Fixed TM Packets window columns not resizing
TMPROP-592	Find Page from Parameter function unavailable when used immediately after login
TMPROP-595	[Sen3/AR/9024] S3 GSRE TM Prop not receiving service 5 events from MCS
TMPROP-596	Out of memory exception on TMPropagator Client
TMPROP-609	Page Directory sometimes takes up to one minute to load after login

3 ISSUES STILL OPEN AFTER THIS RELEASE

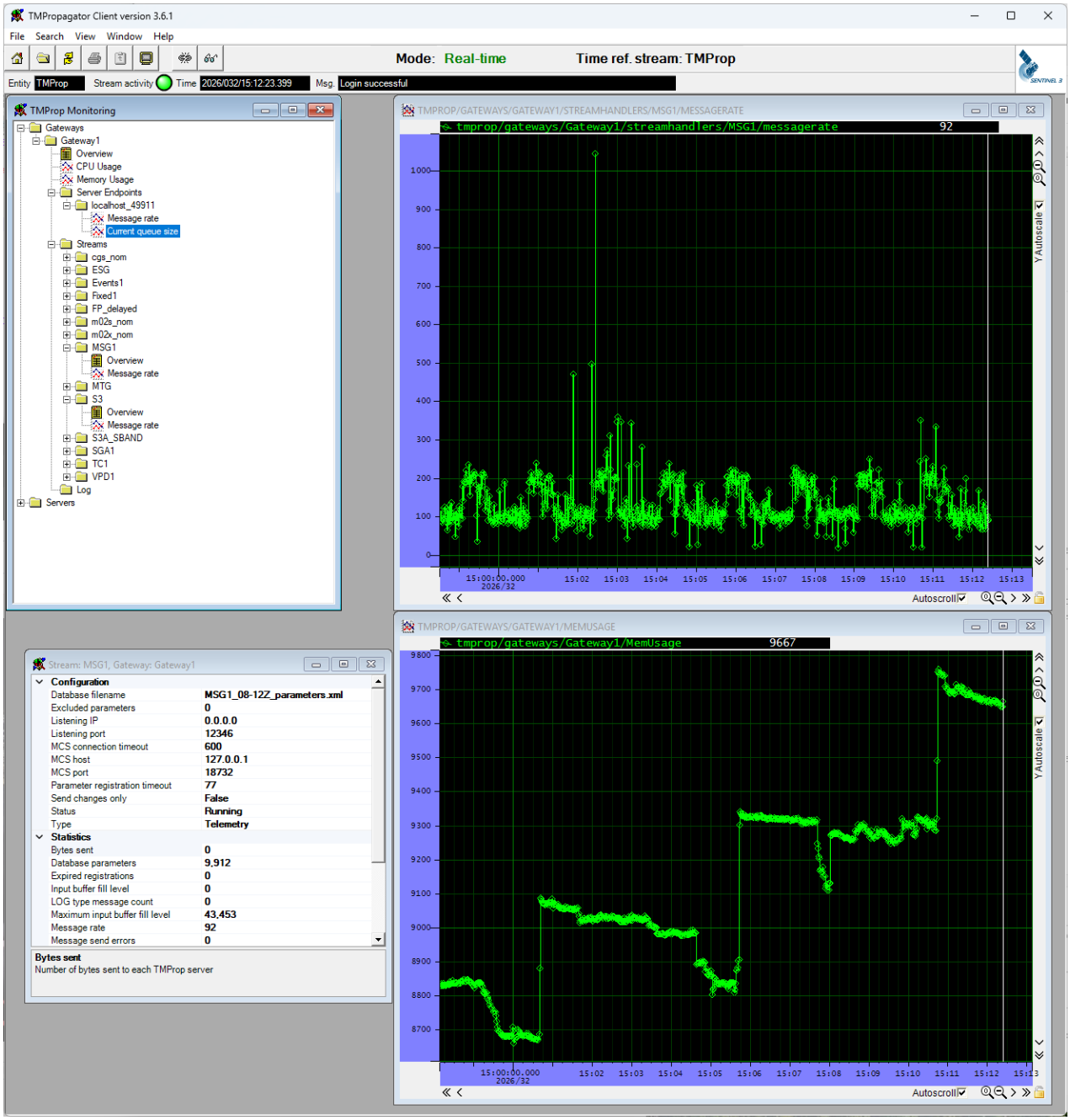
All the open issues concern the mimics page editor. Unfortunately, there was insufficient time and resources to fix them all before the deadline for delivery.

TMPROP-551	[MimicsPageEditor] Some custom templates not viewable on the custom template panel
TMPROP-552	[TMPropagator] Some attributes are not properly supported
TMPROP-553	[TMPropagator] Images pasted on a synoptic are not displayed
TMPROP-554	[MimicsPageEditor] Re-saving complex Mimics with the new editor breaks them
TMPROP-555	[MimicsPageEditor] Hyperlinks broken after re-saving Mimics
TMPROP-556	[MimicsPageEditor] Improvement of the User Manual

4 PERFORMANCE MONITORING (TMPROP-504)

4.1 Description

The monitoring of the TMProagator itself allows an authorized user to monitor the operation of the system remotely, using the TMPropagator client itself. An example of a TMProp monitoring screen is shown below.



Screen shot of a TMProp monitoring display.

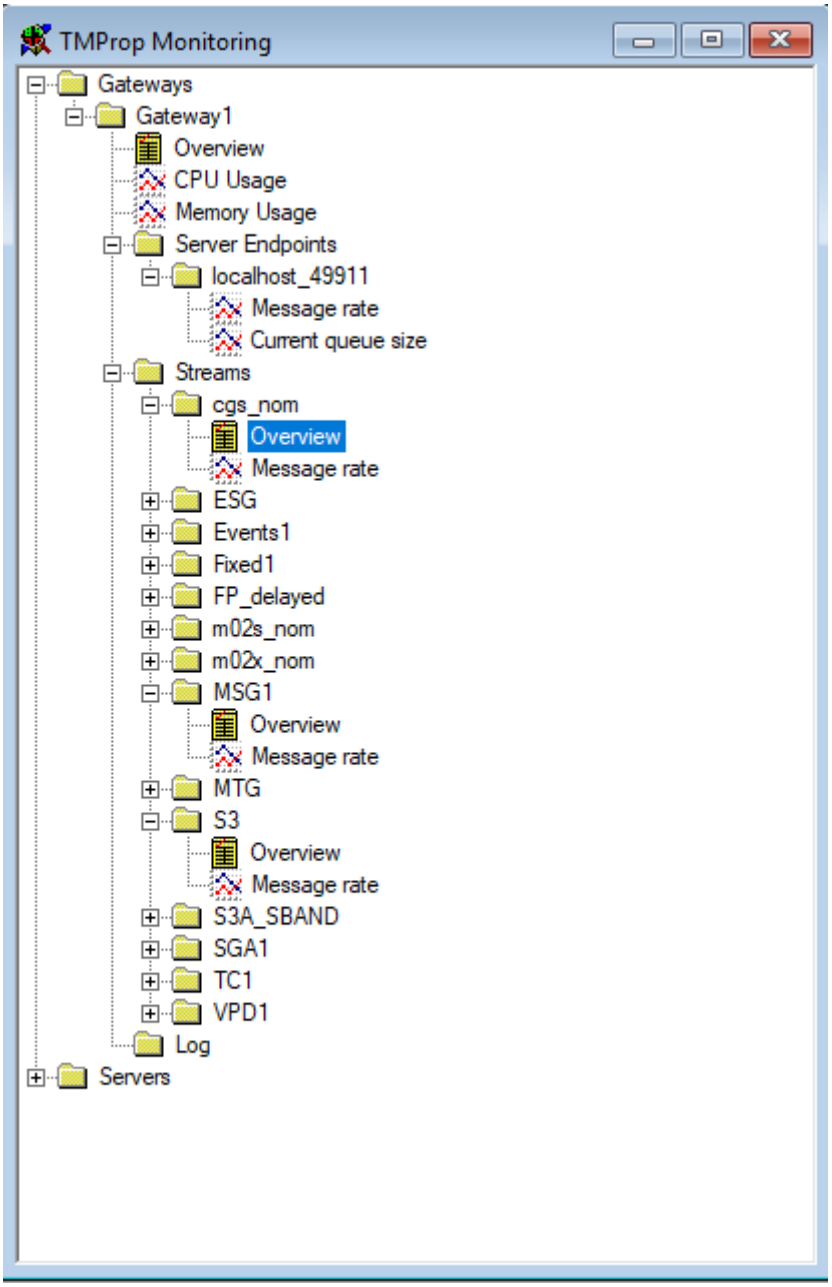
The monitoring is implemented as a special entity called “TMProp” which can be selected when starting the client, providing the user has the necessary authorization. The TMprop entity is not

visible in the Server GUI, but is an intrinsic part of the server and is automatically started with the server. Users are given permission to use the TMProp entity using the user database manager on the server in the usual way.

4.2 Page Types

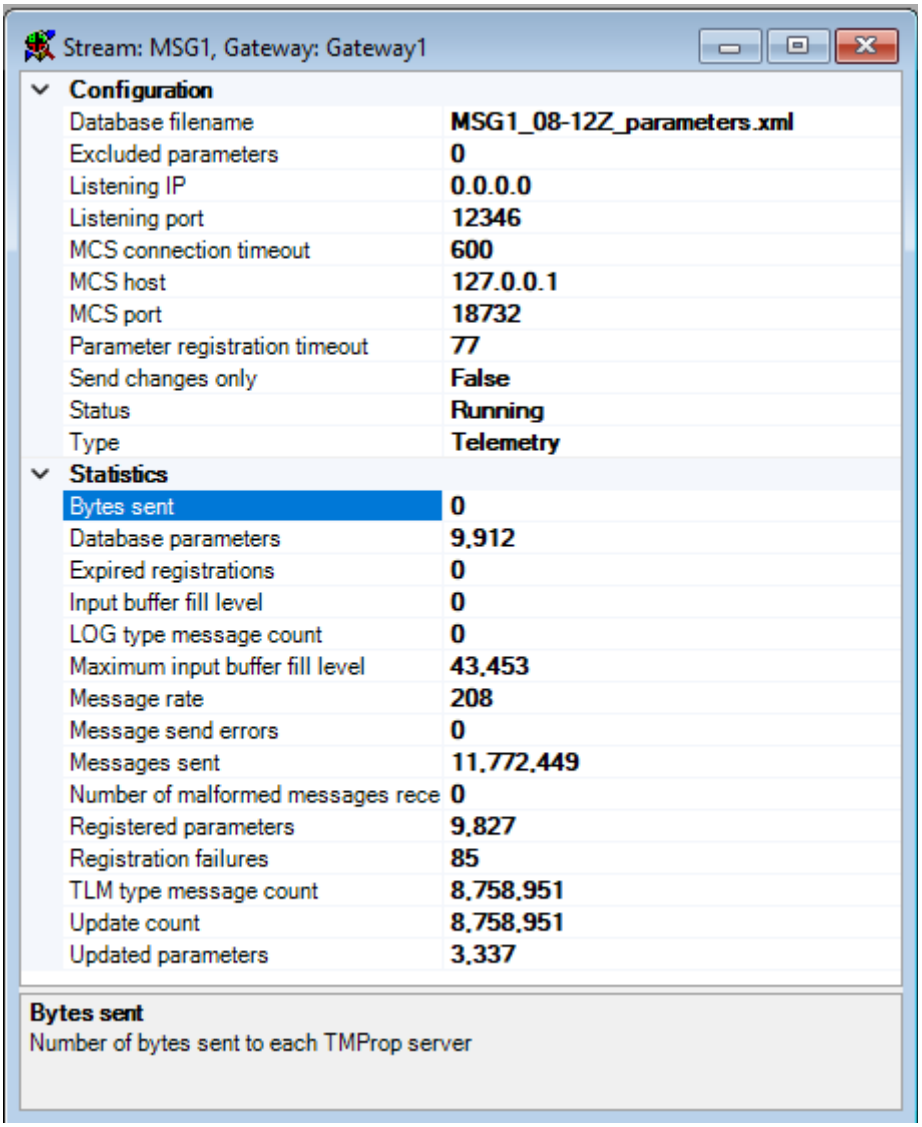
4.2.1 Page Directory

Unlike satellite entities, the TMProp entity generates its page directory dynamically. This avoids the need to manually update the directory, as the TMProp modules are reconfigured, for example by adding additional streams, entities or even Server of Gateway instances.



4.2.2 Alphanumeric

An alphanumeric page shows the parameters of a particular module (e.g. Gateway, Server, Stream Handler) is a dynamically generated display which looks like this:



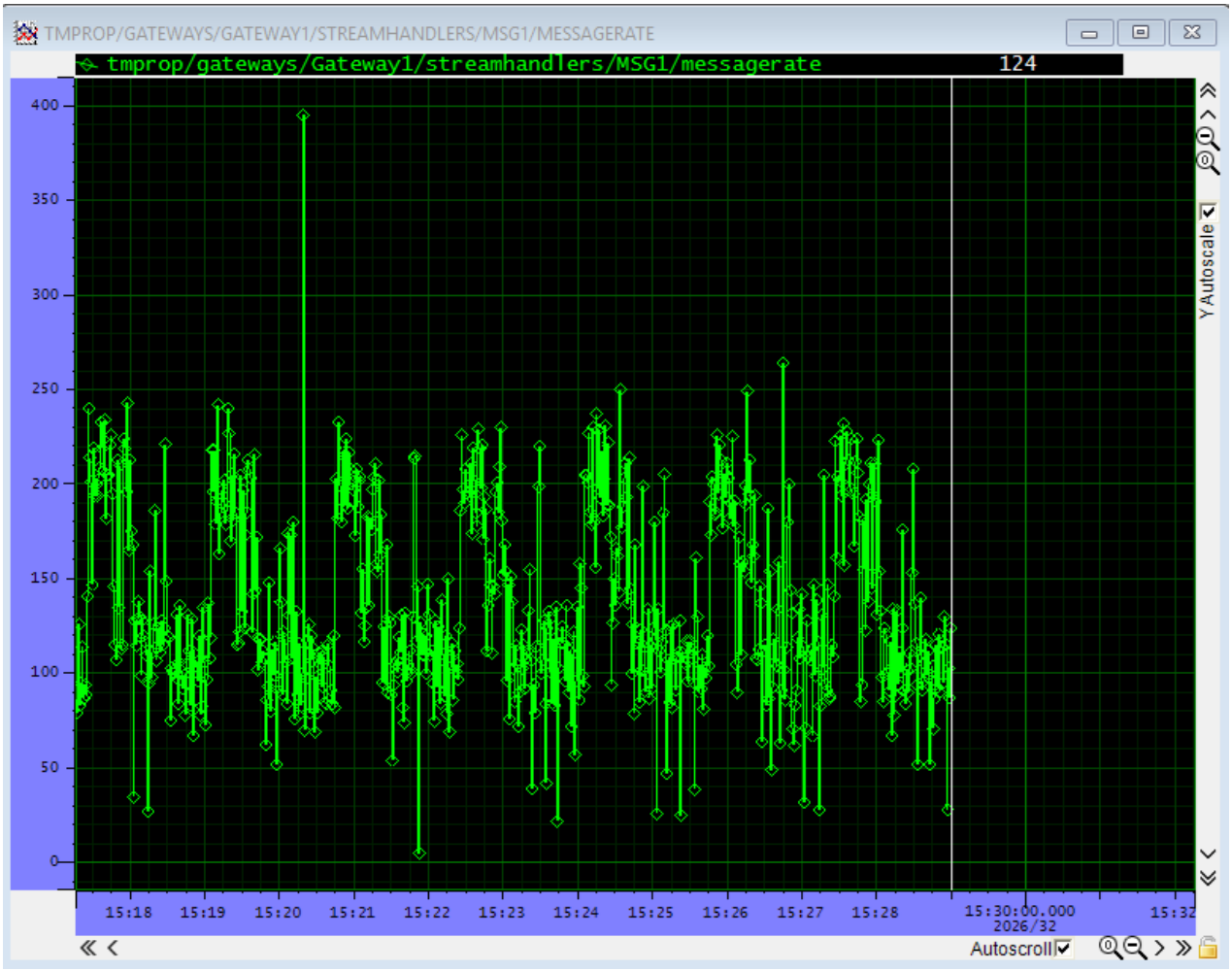
A typical alphanumeric monitoring page

A description of the selected parameter is shown at the bottom of the page.

The parameters are grouped according to function (e.g. Configuratiom or Statistics).

4.2.3 Line-Plot

A sub-set of parameters can be plotted against time. In general, parameters which are useful for monitoring performance can be plotted, such as message rates, queue sizes, CPU and memory usage. In the case of CPU and memory usage, the line plots show the usage for all process running on the machine, not just the TMPropagator processes.



Time plot of the message rate of a gateway stream handler.

4.3 Playback

Most of the TMProp monitoring parameters are not stored in the archive, only those which can be plotted, and it is not possible to use the usual playback controls which are hidden when the TMProp entity is selected. However, the line plots can be scrolled backwards into the past using the usual scroll functions. The alphanumeric displays always show the current values.

4.4 Archive Duration

The implementation stores the history of a subset of parameters in a sub-directory of the server root folder called “monitoringArchive”. This is a different format to the normal archiving, and each parameter is stored in a separate sub-directory.

The oldest files in the archive are automatically purged after a configurable time, to avoid the archive growing indefinitely. The default duration is 10 days, but this can be configured by setting the *monitoringArchiveMaximumAge* attribute of the *Server* element in the config file:

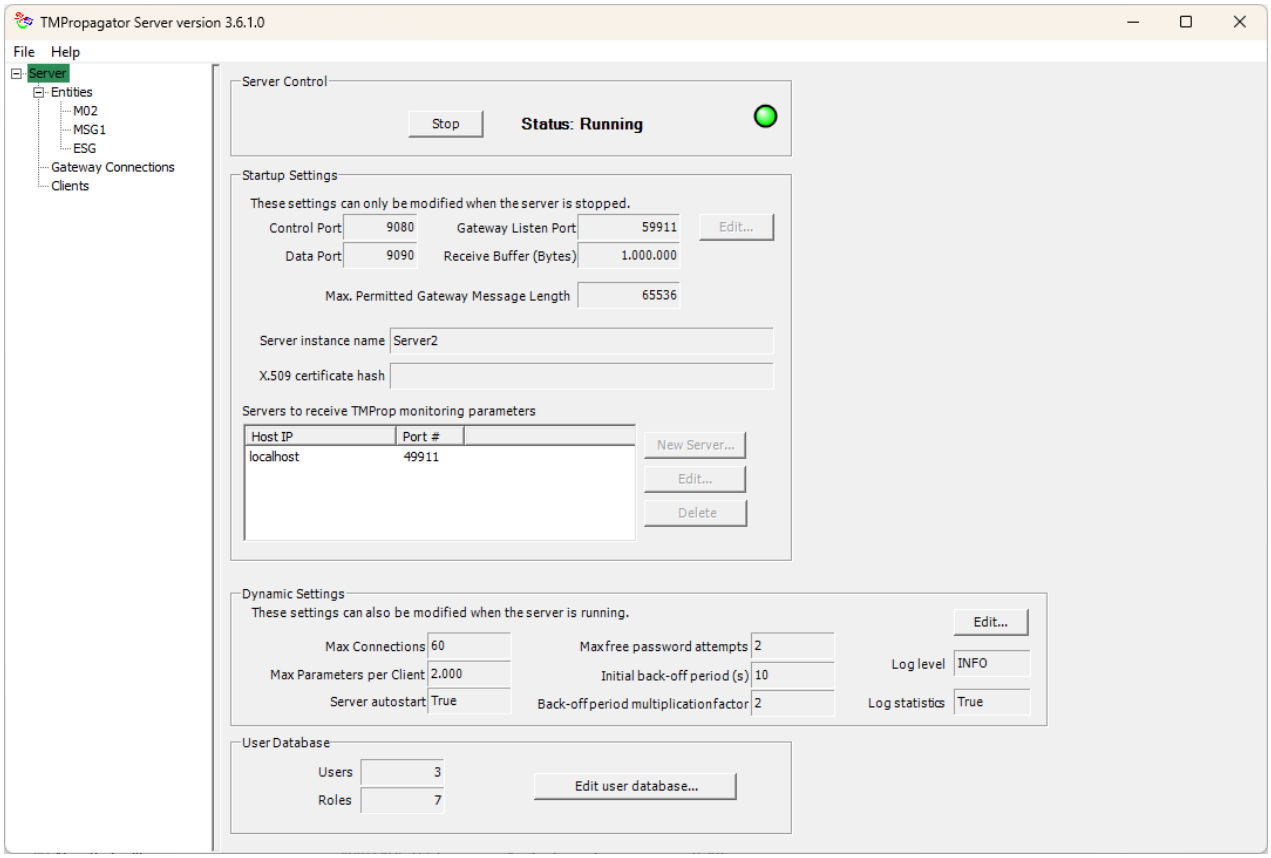
```
<ORSFServerConfig monitoringArchiveMaximumAge="86400" …
```

The archive age should be specified in seconds.

4.5 Configuration

If only a single server is in use, then no further configuration is necessary. However, in the case of multiple servers, then a small amount of extra configuration is necessary, so that each server sends its own monitoring parameters to the other server.

The figure below shows a server configured to send monitoring data to another server instance running on the local host and listening on port 49911. Similarly, the other server instance should have an entry pointing back to this instance.



IMPORTANT!

In addition to the cross-connecting of the servers, it is also necessary to give each server and gateway instance a unique name. If this is not done, then it is not possible to distinguish monitoring samples coming from the different instances. For two gateways and two servers, they could be called Gateway1, Gateway2, Server1 and Server2 for example.

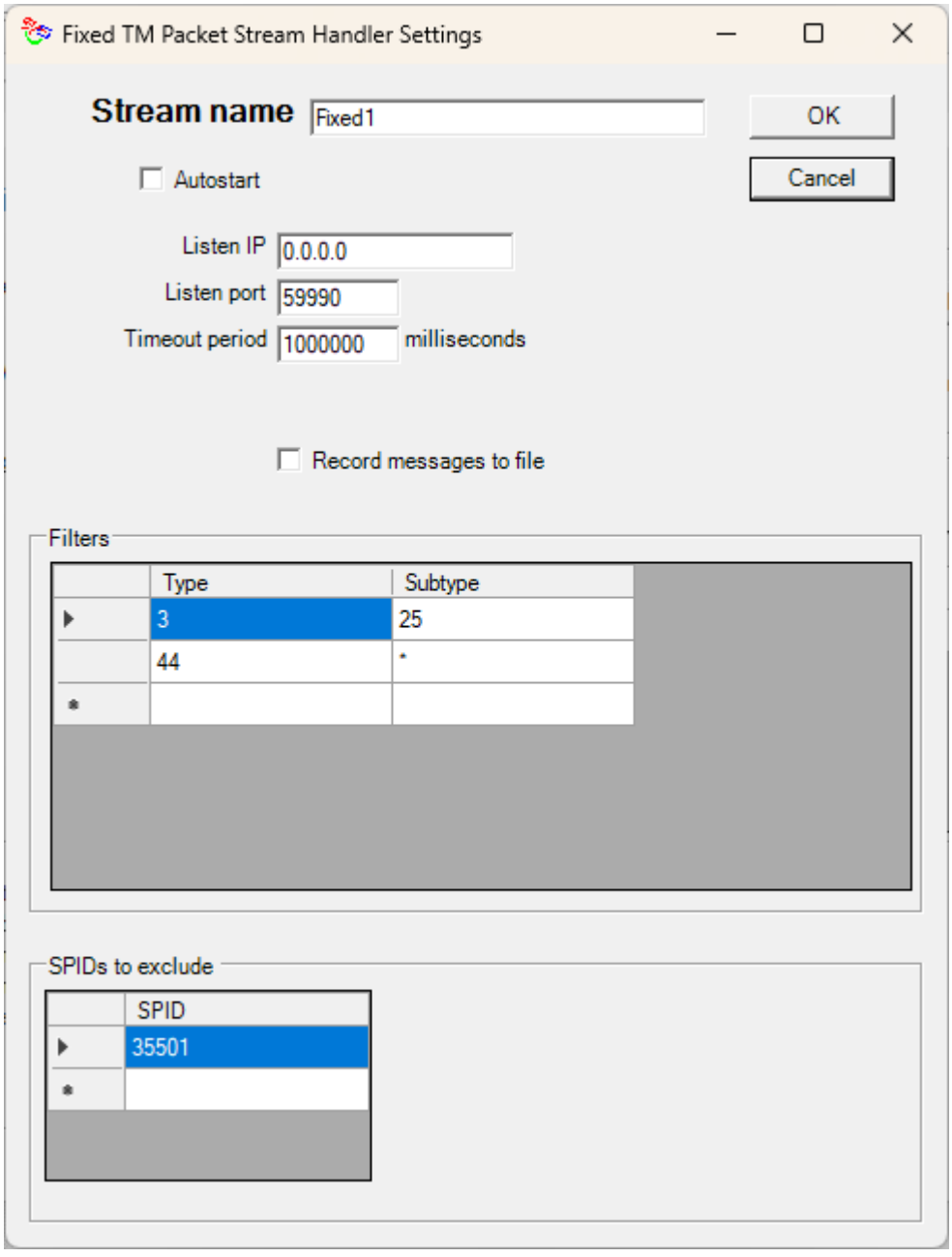
4.6 Legacy Statistics Logging

The statistics logging was implemented as an interim solution to allow the time history of parameters to be displayed in external tools such as Excel. This functionality is still present, but as it is now possible to plot parameters directly in the TMProp client, it is recommended to disable this feature using the server GUI. This will reduce the load on the server. This is done by setting the “Log statistics” option in the dynamic server settings to false.

5 FILTERING OF FIXED PACKETS ON GATEWAY USING SPID (TMPROP-575)

A filter mechanism has been added to the Gateway which allows fixed TM packets to be filtered out on the basis of their SPID. Filtered packets are received from the MCS, but are discarded by the Gateway and not forwarded to the Server for archiving and distribution to the clients.

The Gateway GUI has been updated to allow a set of SPIDs to be defined, as shown below:


Fixed TM Packet Stream Handler Settings
Stream name
Fixed1
OK
Cancel
Autostart
Listen IP
0.0.0.0
Listen port
59990
Timeout period
1000000
milliseconds
Record messages to file
Filters
Type
Subtype
3
25
44
*
SPIDs to exclude
SPID
35501

6 BUILD PROCEDURE AND IPR MESSAGE IN SOURCE FILES (TMPROP-559)

6.1 Source Code Header

A header has been added to every source code file for which EUMETSAT owns the IPR. The header looks like this:

```
// Copyright(©) 2026 EUMETSAT
// License: Proprietary
```

6.2 Provision of Source Code

In this and future releases, all the source code for which EUMETSAT owns the IPR is provided as a single GIT repository. This has the advantage that the change history of the source code is also available.

The source code repository is provided in the *Source_Repository* directory of the distribution.

6.3 Provision of Compiled Binaries for COTS Modules

Many of the applications depend upon modules for which EUMETSAT does not own the IPR. These are referred to as the COTS modules and they are provided already compiled in the *COTS_binaries* directory.

6.4 Required Build Tools

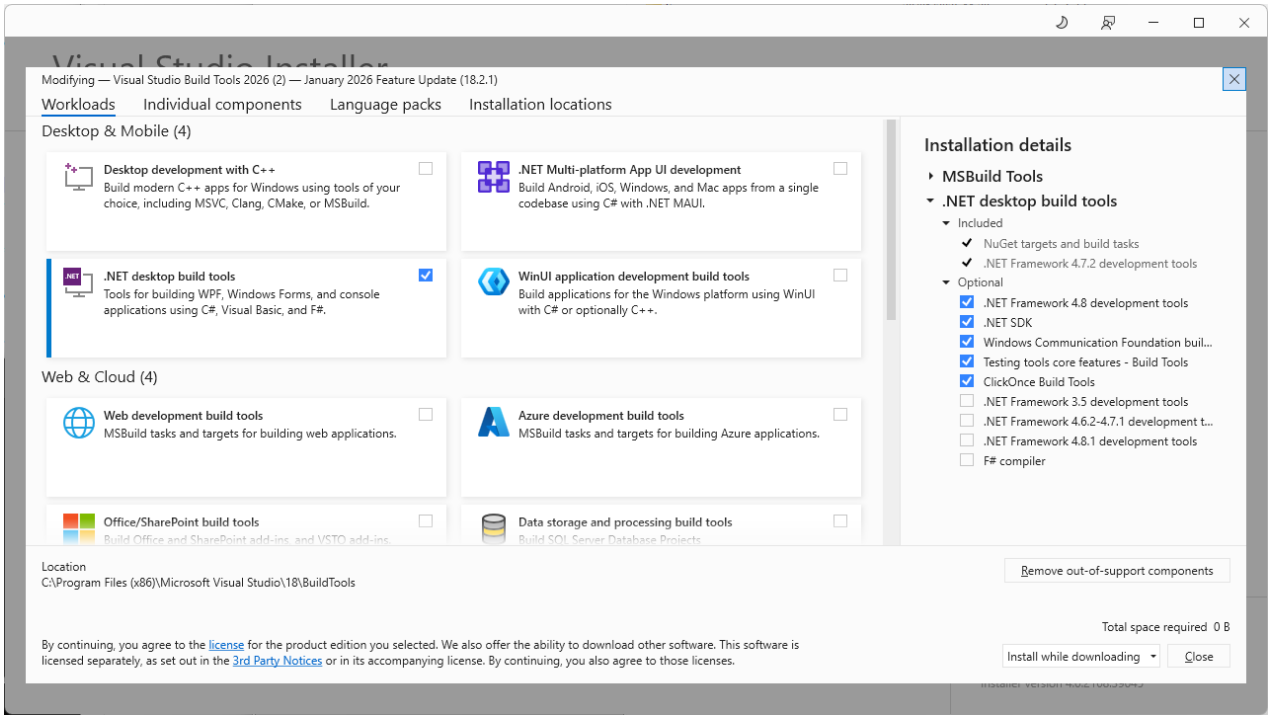
6.4.1 Microsoft Build Tools for VisualStudio 2026

All the TMPropagator applications can be built with the free and open source Microsoft Build tools for Visual Studio 2026. These can be downloaded from here:

https://visualstudio.microsoft.com/downloads/

The installer for these tools is also provided in the release root directory as *vs_BuildTools.exe*

When running the installer, make sure that the ".Net Desktop Development build tools" option is selected as shown below:



The other options are not needed to compile the TMPropagator source code.

6.4.2 Advanced Installer

To generate the TMPropagator client application installer, the free version of the Advanced Installer is required. It can be downloaded from here:

```
https://www.advancedinstaller.com/download.html
```

6.5 Building the TMPropagator Applications

To make the build process easier, a build script has been provided which builds all the applications at once and copies the necessary default configuration files.

If the Microsoft Build Tools have been correctly installed, then there should be a new folder added to the Windows Start menu called *Visual Studio*. Click on the item in this folder called *Developer Powershell for VS*. A new shell window will be opened with the environment variables and PATH correctly set for building the applications.

In the new shell window, change the directory to the *Source_Repository* directory of the release.

Start the build process by typing this command:

```
.\BuildAll.bat
```

The build process should run and produce some warnings, but no errors. The source files located in the *Source_Repository* directory should have been compiled and each TMPropagator application should be in its own sub-directory in a newly created *BuildOutput* directory which is itself a sub-directory of *Source_Repository*.

6.6 Running the Compiled Applications

The build script creates a set of startup scripts for the most important applications. These are all stored in the *BuildOutput* directory and are called *StartGateway.bat*, *StartServer.bat* etc. They can all be run from a console window, or simply by clicking on the script in Windows Explorer.

6.7 Generating the Client Installer

The client installer package requires an additional step to generate it from the source code, as it can only be created using the *Advanced Installer* application. After having run the build script described in the previous steps, open the *Advanced Installer* application and then load the file .*\Source_Repository\ClientInstaller\ClientInstaller.aip* using the File/Open menu. Finally click on the Build button in the application header to generate the installer which will be placed in the *ClientInstaller-SetupFiles* sub-directory.

6.8 Provision of Pre-compiled Applications

Pre-compiled versions of the applications are provided as usual, in addition to the source code files. These can be used directly and it is not necessary to compile the source files to install the system. These pre-compiled applications can be found in the usual sub-directories (e.g. *Server*, *Gateway* and *Client*)

7 IMPROVEMENTS AND NEW FEATURES UNRELATED TO NON-CONFORMANCES

7.1 Gateway Outgoing Message Queue Configuration

The gateway uses a queue for outgoing messages to each server. The maximum queue size was previously hard coded to 100,000 messages. If the maximum size is reached, then messages are discarded by the gateway to avoid a memory overflow. This has been seen to happen on some projects (see issue TMPROP-601), and results in missing data in the server archive.

The maximum queue size has now been made configurable, via the Gateway GUI.

7.2 Test Server Tool

The test server tool which generates telemetry updates for testing has been improved, and it is now possible to generate values for a test parameter either from a fixed value which is entered manually or using a simple sine function to produce a constantly updating value. Furthermore, the state flags can be manually changed, as well as the period between generated samples.

This feature can be used, for example, to generate a very high frequency updating parameter.

8 INSTALLATION INSTRUCTIONS

Please see the document *Configuration Manual for Gateway, Server and Dev Machine* for full installation instructions

The logging level specified in the logging.config.xml file in the config directory of both Gateway and Server is no longer used. The log level is now set via the GUI and the setting is stored in the normal gateway and server config files. Nonetheless, the logging.config.xml files are still required for other logging settings.

This release has been compiled for the .Net Framework 4.8, and may require this to be installed on the target machines. However, most recent versions of Windows already have this pre-installed.

9 COMPATIBILITY TO PREVIOUS VERSIONS

Due to a change in the technology used for the client server interface, neither the client nor the server are compatible with previous versions, so the client included in this release must be deployed together with the new server.

The Gateway to Server interface has not changed, but it is still recommended to deploy the new Gateway version.

10 SYSTEM REQUIREMENTS

10.1.1 Operating System

The TMPropagator Gateway and Server require Windows Server 2019 or higher as the operating system.

For the clients, Windows 10 or higher is required.

10.1.2 .NET Framework 4.8

All machines where any module of the TMPropagator software is installed requires the .NET Framework 4.8 (or higher) as a prerequisite.

Download and install the latest Microsoft .NET Framework version (4.8 at time of writing) for the machines operating system from the Microsoft web site. It is not necessary, or desirable, to uninstall the previous .Net versions first. This step might not be necessary if .NET 4.8 or higher is already installed. This can be verified by following the procedure given here;

https://docs.microsoft.com/en-us/dotnet/framework/migration-guide/how-to-determine-which-versions-are-installed

10.1.3 Hardware

A minimum of 8 GB memory is recommended for the Gateway and Server machines. This should be increased if many projects are to share the same hardware, or if they have particularly high data rates.

The disk usage depends almost entirely on the telemetry data rate and the desired length of archive. It is suggested to compare the anticipated data rates and archive duration of new projects to older projects when estimating disk space requirements.

The new stream types have also been found to generate considerable quantities of data which increases the size of the archive.

For the clients, any modern PC hardware should be sufficient. A minimum 4 GB memory is recommended.