GEUTEBRÜCK

G-Connect

User guide

Author: Ewenz Michael

Version history

Date	Author	Reviewed by	Description
09.03.2018	Michael Ewenz		Initial creation
22.03.2018	Michael Ewenz	Martin Hauk	Supported data types added (Section 9.4)
26.03.2018	Michael Ewenz	Martin Hauk	Spelling fixes and further description
16.05.2018	Michael Ewenz		New plug-in for reading XML files XPath expressions can contain member references
10.08.2018 (1.1.1.27)	Michael Ewenz		Telnet plug-in now also accepts JSON strings
17.08.2018 (1.1.1.30ff)	Michael Ewenz		Improvement: A worker can do additional work on reaction to received XML/JSON data synchronously or asynchronously. The accepted XML/JSON data size can be configured in the Telnet plug-in. Logging can be configured for the socket connections in Telnet plug-in.
22.08.2018 (1.1.1.33ff)	Michael Ewenz		Type TMediaChannelID implemented
23.08.2018 (1.1.1.35ff)	Michael Ewenz		The Telnet plug-in can be configured to accept strings matching a defined regular expression. (Section 8.3)
07.09.2018 (1.1.1.37ff.)	Michael Ewenz		With help of a rules-engine incoming XML documents can be operated to generate multiple G-Core actions according to the contents of the document. See section 6.
12.09.2018 (1.1.1.37ff.)	Michael Ewenz		New plug-in for WebSocket communication. (Removed in version 4.0.0.49).
20.09.2018 (1.1.1.38ff.)	Michael Ewenz		Passwords can be stored AES256 encrypted in configuration files.
27.09.2018 (1.1.1.39ff.)	Michael Ewenz		Improvement in WebSocket plug-in
15.10.2018 (1.1.1.42ff.)	Michael Ewenz		Websocket sub protocol V10.PCP.SAP.COM implemented
29.10.2018 (1.1.1.44ff.)	Michael Ewenz		Duplicate action suppression and additional action field type TEventTypeID implemented
16.11.2018 (1.1.1.48ff)	Michael Ewenz		Predefined members values may contain placeholders for inserting other member values. For controlling worker plug-ins rulesets can be defined An additional plug-in for Web API connection to CyberExtruder face recognition application was created. An additional worker plug-in for copying files was created
08.01.2019 (1.1.1.50ff)	Michael Ewenz		G-Core Events can be configured to trigger actions inside the service
14.01.2019 (1.1.1.53ff)	Michael Ewenz		If- and Else-Clauses can point to different respons actions.

			Special names in predefined elements ca be used for simple calculations like add, sub, mul and div.
14.01.2019 (1.1.1.53ff)			The File-Watcher plug-in can handle xml, csv and json formatted files.
11.03.2019 (1.1.1.57ff)	Michael Ewenz		New XMLConnectFileWriter plug-in. Integrated new licence model. GCoreXMLConnectWebAPI plug-in exchanged against XMLConnectFaceRec plug-in.
14.03.2019 (1.1.1.59ff.)	Michael Ewenz		The FileCopy plug-in may clear selected directories at start up depending on configuration.
20.03.2019 (1.1.1.61ff)	Michael Ewenz		While installation the components can be selected. Creating G-Core actions allows using the parameter type “Enum”. The additional attribute “EnumType” declares the used value.
05.04.2019 (1.1.1.66ff)	Michael Ewenz		New plug-in for CarmenGo LPR New token for predefined members
09.05.2019	Michael Ewenz		New plug-in for extracting pictures from a received XML document In Face Recognition plug-in the received data can be operated by regular expression
03.06.2019 (1.1.1.75ff)	Michael Ewenz		New plug-in for connecting to vehicle management system CargoClix via a RESTful Web API client
04.06.2019 (1.1.1.75)	Michael Ewenz		Object VehicleStatus has been changed in CargoClix plug-in
13.06.2018 (1.1.1.80ff)	Michael Ewenz		In CargoClix plug-in REST API communication can be SSL encrypted
18.06.2019 (1.1.1.83ff)	Michael Ewenz		Addes Ruckus plug-in
24.06.2019 (1.1.1.83ff)	Michael Ewenz		Added codepage configuration in face recognition plug-in
05.09.2019 (1.1.1.94ff)	Michael Ewenz		New plug-in providing a RESTful API
21.10.2019	Michael Ewenz		Convert received date/time in action to ISO 8601 format New plug-in for connection to Setel Petronas App
22.11.2019 (1.1.1.100)	Michael Ewenz		Plug-in for connection to Briefcam added
28.11.2019 (1.1.1.102ff.)	Michael Ewenz		Plug-in for connection to LivAssured NightWatch Placeholders in “Predefined” region can also stand for values determined by rules.
06.12.2019 (1.1.1.104ff.)	Michael Ewenz		XmlFileCopy plug-in enabled to read files from a shared network folder (with authentication) and write to another shared network folder (also with authentication)

			It can delete the contents of a shared network folder as well. XmlConnectRESTService plug-in may operate POST requests of typ multipart/form-data containing types application/json and/or image/jpeg. A received image is written into a file locally or into a shared network folder.
16.12.2019 (1.1.1.107ff.)	Michael Ewenz		The request mapping entry may contain a reference to a ruleset. New plug-in interfacing Mercareon vehicle management system
28.01.2020 (2.0.0.0ff.)	Michael Ewenz		Multiple G-Core servers can be accessed simultaneously. Files to be read and written may be located in shared network folders.
14.02.2020 (2.0.0.5ff)	Michael Ewenz		New plug-in for integrating Kuando Busylight
25.03.2020 (3.0.0.1)	Michael Ewenz		New architecture for XMLConnect. G-Core connection is now just a plug-in like all others. This affects most configurations. Therefore there is a huge update of almost all chapters. The solution is renamed to **G-Connect**.
11.05.2020 (3.0.0.8)	Michael Ewenz		REST Client plug-in added
23.06.2020 (3.0.0.9)	Michael Ewenz		New features added to read device and server infos
29.07.2020 (3.0.0.13)	Michael Ewenz		Linked documents reference to the pdf version
17.08.2020 (3.0.0.16)	Michael Ewenz		In the FileProviderPlugin file content can be cached locally
21.08.2020 (3.0.0.18)	Michael Ewenz		Predifined element in XMLConnectConfig.xml extended by #Trim command
31.08.2020 (3.0.0.20)	Michael Ewenz		Plug-in for connection to IMRON UnityIS (TCP client) added
15.09.2020 (3.0.0.24)	Michael Ewenz		Plug-in for connection to VideoGuard (SIRA) added plus some necessary base features
05.10.2020 (3.0.0.26)	Michael Ewenz		Added value #RegEx to Predefined members
02.11.2020 (3.0.0.28)	Michael Ewenz		Plug-in for connection to FLIR PSR radar server
03.11.2020 (3.0.0.29)	Michael Ewenz		GCore plug-in: Added feature to send JPEG images to DirectImageFeed channel. FileWatcher plug-in: JPEG files can be read
25.01.2021 (3.0.0.34)	Michael Ewenz		Plug-in for connection to OnGuard 7.6 and 8.0 via OpenAccess interface
10.02.2021 (4.0.0.3)	Michael Ewenz		New Licensetypes and renaming
15.03.2021 (3.0.0.40)	Michael Ewenz		TCPClient tests network connection with regular Ping requests.
31.03.2021 (4.0.0.16)	Michael Ewenz		In Predefined element a conversion of UTC timestamp to local date and time can be configured

16.04.2021 (4.0.0.20)	Michael Ewenz		New plug-in for starting external processes. New plug-in for notifying changes in an SQL Database
11.05.2021 (4.0.0.30)	Michael Ewenz		New plug-in presenting an NTP client. New plug-in for regularly checking the harddisk status. G-Core plug-in sends messages if a G-Core server connects or disconnects and if a camera is detected or detection failed.
17.05.2021 (4.0.0.32)	Michael Ewenz		New plug-in for retrieving web content from a given URL
27.05.2021 (4.0.0.33)	Michael Ewenz		Application based on new SDK 6.1.0.226
18.06.2021 (4.0.0.39)	Michael Ewenz		Closer description of Flir Radar plug-in mapping. Parameters of predefined functions with multiple params must be enclosed in brackets separately. (see 10.3.3)
29.07.2021 (4.0.0.41)	Michael Ewenz		In the G-Connect configuration a list of lookup tables for different source values can be defined to add multiple destination values for one source value
10.08.2021 (4.0.0.44)	Michael Ewenz		Plug-ins renamed to unify the naming. Configuration file parameter added to the G-Connect configuration to allow loading multiple instances of a plug-in with different configurations.
19.08.2021 (4.0.0.46)	Michael Ewenz		Message blocker plug-in added. Function for replacing characters in a string field added.
27.08.2021 (4.0.0.48)	Michael Ewenz		Message combine plug-in added
30.08.2021 (4.0.0.49)	Michael Ewenz		Description of WebSocketClient plug-in, Ruckus plug-in and Setel plug-in removed
01.09.2021 (4.0.0.51)	Michael Ewenz		XPath2 query implemented for ability to use extended XPath 2.0 functions. CarmenGo plug-in improved to work with CarmenGo 1.4.0
09.09.2021 (4.0.0.53)	Michael Ewenz		With help of the G-Core plug-in it is possible to read event images and forward them to other plug-ins
17.09.2021 (4.0.0.57)	Michael Ewenz		A generic REST client is now implemented in the core part of G-Connect. Plug-ins may use this. There's no need to implement it separately in the plug-in.
19.09.2021 (4.0.0.58)	Torsten Krügel		Plug-in for NEDAP AEOS 2021.2
06.10.2021 (4.0.0.63)	Michael Ewenz		Based on G-Core SDK version 6.0.0.49 (Current official release).
15.11.2021 (4.0.0.66ff)	Michael Ewenz		Based on G-Core SDK version 6.1.0.365 because of new license key for Nedap plug-in

26.11.2021 (4.0.0.188)	Michael Ewenz		Multiple actions triggering events can be configured
17.12.2021 (4.0.0.191)	Michael Ewenz		The restful client plug-ins are now supporting digest authentication
07.01.2022 (4.0.0.193)	Michael Ewenz		Password encryption basis changed
19.01.2022 (4.0.0.195)	Michael Ewenz		New predefined function `#LocalHostname` retrieves the hostname of the local machine.
20.01.2022 (4.0.0.196)	Michael Ewenz		Behavior of the VideoGuard plug-in changed to meet the requirements.
14.03.2022 (4.0.0.201)	Michael Ewenz		Changes in the kernel and in the SQL plug-in to assure correct receiving if a bigger amount of datasets is written simultaneously.
17.03.2022 (4.0.0.202)	Michael Ewenz		The plug-ins FileWatcher and FileProvider are able to read JPEG images from a file.
04.05.2022 (4.0.0.204)	Michael Ewenz		RESTService plug-in implements SSL and basic authentication
16.05.2022 (4.0.0.210)	Michael Ewenz		REST client urls may contain variables in the configuration file. The body content of a PUT or POST command may be JSON, XML or plain text formatted. A REST API call returns 204 – NoContent if it is accepted and operated correctly.
02.06.2022 (4.0.0.211)	Michael Ewenz		In the REST client plug-in POST and PUT bodies may contain single integer, Boolean or text values if the format type is “text”. The PostRepetitionInterval now contains an attribute for the number of repetitions.
17.06.2022 (4.0.0.212)	Michael Ewenz		The RunProcesses plug-in is now able to run a windows process triggered by an external event. The argument may contain variables.
01.07.2022 (4.0.0.213)	Michael Ewenz		G-Core plug-in: All actions containing a TMediChannelID parameter can now correctly be created.
15.07.2022 (4.0.0.215)	Michael Ewenz		Example configurations for Code Blue and Entry Pass P1 added to the installer Http content type x-www-form-urlencoded is processed as string.
18.07.2022 (4.1.0.3)	Michael Ewenz		Version officially released
10.08.2022 (4.1.0.4)	Michael Ewenz		GConnectFileWatcher plug-in is able to register Mpeg4 files for further operation.
01.09.2022 (4.1.0.5)	Michael Ewenz		G-Core plug-in can handle TResourceID parameters.
13.09.2022 (4.1.0.6)	Michael Ewenz		GConnectSAPSOAPClient plug-in adapted with new wsdl because of adding YDID field
09.11.2022 (4.1.0.8)	Michael Ewenz		New plug-ins GConnectSQLitePlugin and GConnectProcessXmlFiles. XML files containing namespaces can be handled.

			New predefined function for conversion of DateTime values to Unix timestamp implemented.
25.01.2023 (4.1.0.16)	Torsten Krügel		Plugin for CIAS IB server
25.01.2023 (4.1.0.17)	Torsten Krügel		Plugin for Fibersensys APU
31.01.2023 (4.1.0.22)	Michael Ewenz		LoadWebContent plugin extended by authentication possibility
15.05.2023 (5.0.0.0)	Torsten Krügel		GConnect Failover / Backup
15.05.2023 (5.0.0.0)	Torsten Krügel		Plugin for Platesmart
13.06.2023 (5.0.0.7)	Torsten Krügel		Plugin for Bastille
23.06.2023 (5.0.0.8)	Torsten Krügel		Plugin for Southwest Microwave RPM II
07.07.2023 (5.0.0.13)	Torsten Krügel		Plugin for Logflows
14.07.2023 (5.0.0.15)	Torsten Krügel		Add information about Failover capability
19.07.2023 (5.0.0.16)	Torsten Krügel		Added information about Failover
02.08.2023 (5.0.0.19)	Torsten Krügel		Plugin for CCure9000
19.09.2023 (5.0.0.22)	Torsten Krügel		Plugin for CCure9000 Version 2.9
19.09.2023 (5.0.0.23)	Torsten Krügel		Plugin for Southwest Microwave POE
12.12.2023 (5.0.0.67)	Torsten Krügel		TCP client upgrade
21.02.2024 (5.0.0.74)	Torsten Krügel		DMP Plugin
20.03.2024 (x.0.0.x)	Torsten Krügel		MBSecure Plugin

Contents

1 General definitions 12
2 Introduction 12
2.1 Purpose 12
2.2 Examples 13
2.3 Use cases 13
3 Architecture 16
4 Installation 17
5 Uninstallation 19
6 Rule engine 21
7 Password encryption 23
8 Plug-ins 23
8.1 GCore Plug-in 23
8.2 File Watcher Plug-in 24
8.3 Telnet Plug-in 25
8.4 File Provider Plug-in 25
8.5 Webservice File Save Plug-in 26
8.6 Face Recognition Plug-in 26
8.7 File Copy Plug-in 26
8.8 File Writer Plug-in 27
8.9 CarmenGo Plug-in 27
8.10 Extract Picture Plug-in 27
8.11 CargoClix Plug-in 28
8.12 RESTfull API Plug-in 28
8.13 Briefcam Plug-in 29
8.14 NightWatch Plug-in 29
8.15 Mercareon Plug-in 31
8.16 Kuando Busylight Plug-in 32
8.17 REST Client Plug-in 34
8.18 TCP Client (UnityIS) Plug-in 35
8.19 VideoGuard Plug-in 36
8.20 FLIR PSR Server Plug-in 36
8.21 OnGuard Plug-in 37
8.22 Run external processes Plug-in 38
8.23 SQL database notification Plug-in 38
8.24 NTP client Plug-in 39
8.25 Health Plug-in 40
8.26 Load web content Plug-in 40

8.27 Message blocker plug-in 41
8.28 Message combine plug-in 41
8.29 NEDAP AEOS plug-in 41
8.30 SAP SOAP Client 43
8.31 SQLite database 43
8.32 Process XML-files 43
8.33 CIAS IB server plug-in 44
8.34 Fibersensys APU plug-in 45
8.35 Platesmart plug-in 47
8.36 Bastille plug-in 53
8.37 Southwest Microwave RPMII plug-in 54
8.38 Logflows 57
8.39 CCure9000 62
8.40 Southwest Microwave POE plug-in 65
8.41 DMP Alarm Panel plug-in 69
8.42 MBSecure plug-in 71
9 Licensing 75
10 Configuration 75
10.1 Start and stop the service 75
10.2 Application configuration 76
10.3 Main G-Connect configuration 77
10.3.1 Element ThirdPartyConnection 77
10.3.2 Element XPathExpressionList 77
10.3.3 Element Predefined 78
10.3.4 Element MappingList 79
10.3.5 Element Stylesheets 80
10.3.6 Element XMLFiles 80
10.3.7 Element Rulesets 80
10.3.8 Element LookupTables 80
10.3.9 Namespaces list 80
10.3.10 Configuration examples 80
10.3.11 Supported data types and formats 82
10.4 RESTful Client configuration 83
10.5 Plug-ins 84
10.5.1 G-Core Plug-in 84
10.5.2 File Watcher Plug-in 89
10.5.3 Telnet Plug-in 89
10.5.4 File Provider Plug-in 90

10.5.5 Webservice File Save Plug-in 90
10.5.6 Face Recognition Plug-in 90
10.5.7 File Copy Plug-in 90
10.5.8 CarmenGo Plug-in 91
10.5.9 CargoClix Plug-in 91
10.5.10 RESTfull API Plug-in 92
10.5.11 Briefcam Plug-in 92
10.5.12 NightWatch Plug-in 92
10.5.13 Mercareon Plug-in 92
10.5.14 REST Client Plug-in 93
10.5.15 TCP Client (UnityIS) Plug-in 94
10.5.16 VideoGuard Plug-in 94
10.5.17 FLIR Plug-in 94
10.5.18 OnGuard Plug-in 94
10.5.19 Start external processes Plug-in 95
10.5.20 SQL Server notification Plug-in 96
10.5.21 NTP client Plug-in 96
10.5.22 Health Plug-in 97
10.5.23 LoadWebContent Plug-in 97
10.5.24 Message blocker plug-in 97
10.5.25 Message combine plug-in 97
10.5.26 SAP SOAP client 97
10.5.27 SQLite database 98
10.5.28 Process XML files 98
11 Used Licenses 98
12 Failover / Backup functionality 99
12.1 Overview 99
12.2 Failover capable plugins 100
12.3 Configuration file 100
12.4 Note 103
12.5 G-Connet Main workflow 103
12.6 G-Connect Backup workflow 103
12.7 Licenseing 104

Figures

Figure 1: Third party triggered 12
Figure 2: G-Core action triggered 13
Figure 3: Use case action triggered 14
Figure 4: Use case 3rd party triggered 14
Figure 5: Use case 3rd Party driven file request and action creation 15
Figure 6: Architecture 16
Figure 7: Recommended installation path 17
Figure 8: Select the components 18
Figure 9: Installation information 18
Figure 10: Setup progress bar 19
Figure 11: Completion screen 19
Figure 12: Setup options menu 20
Figure 13: Uninstallation success message 20
Figure 14: Announcement structure 30
Figure 15: Message structure 30
Figure 16: clientactive message 30
Figure 17: seizurealarm message 30
Figure 18: techalarm message 31
Figure 19: dailystatus message 31
Figure 20: NightWatch Web interface 31
Figure 21: Windows services 76
Figure 22: G-Connect.exe.config 76
Figure 23: Reference to ThirdPartyConnection ID in Mapping 81
Figure 24: Reference to XPathExpressionSets and Rulsets in Mapping 81
Figure 25: References between XPathExpressions 81
Figure 26: References between XPathExpressions and Rules 82
Figure 27: References between Predefined values and XPathExpressions 82
Figure 28: Refrence between XPath expression and Namespaces list 82

1 General definitions

- G-Core – Main server software for recording and maintaining video material by Geutebrück GmbH.
- G-Scope – Digital recording device by Geutebrück GmbH.
- XML – Extensible markup language.
- JSON – JavaScrip Object Notation
- SOAP – Simple Object Access Protocol
- SQL – Structured Query Language

2 Introduction

2.1 Purpose

G-Connect is a Windows service to connect G-Core to 3rd Party applications regardless of communication types. The service devides into 2 parts, the basic module to process XML documents and a plug-in mechanism to handle connections to a wide range of different 3rd Party applications.
In general the service covers two main scenarios:

- A third party source triggers processing the data by the service and creating a G-Core action containing the particular data.

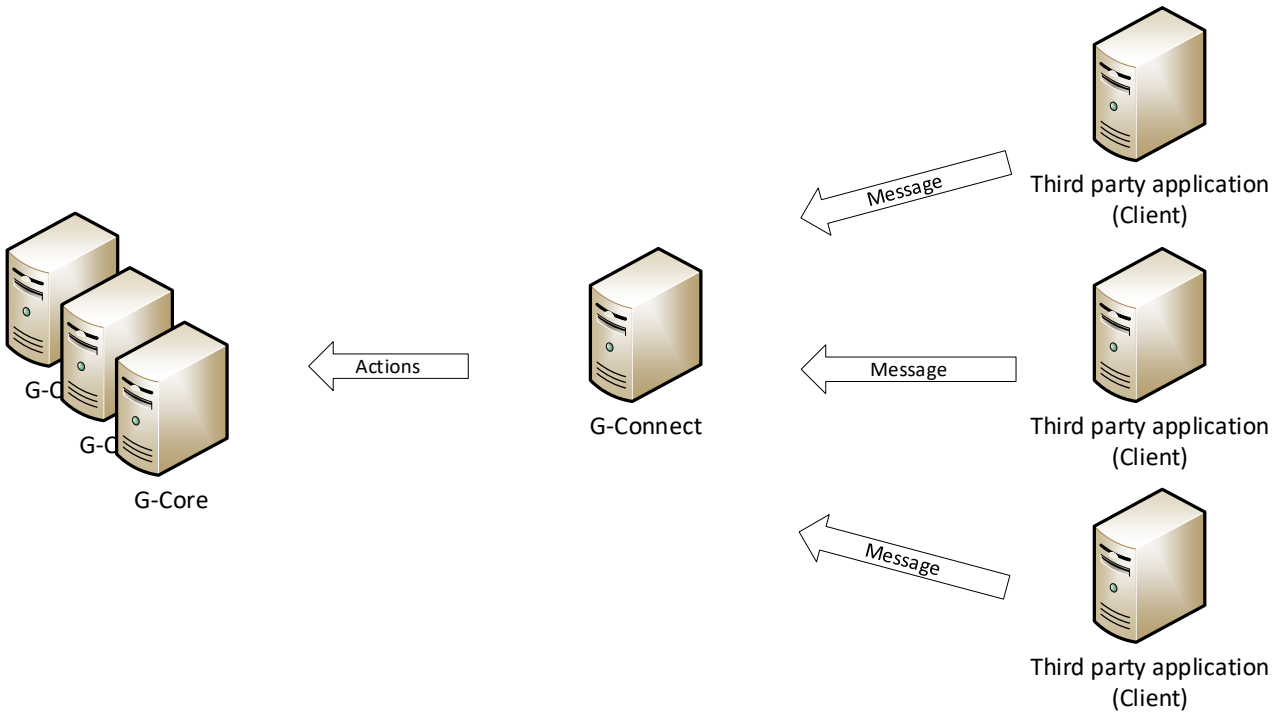

Figure 1: Third party triggered

- A G-Core action triggers requesting data from the XML source. The data is processed and written into a G-Core answer action.

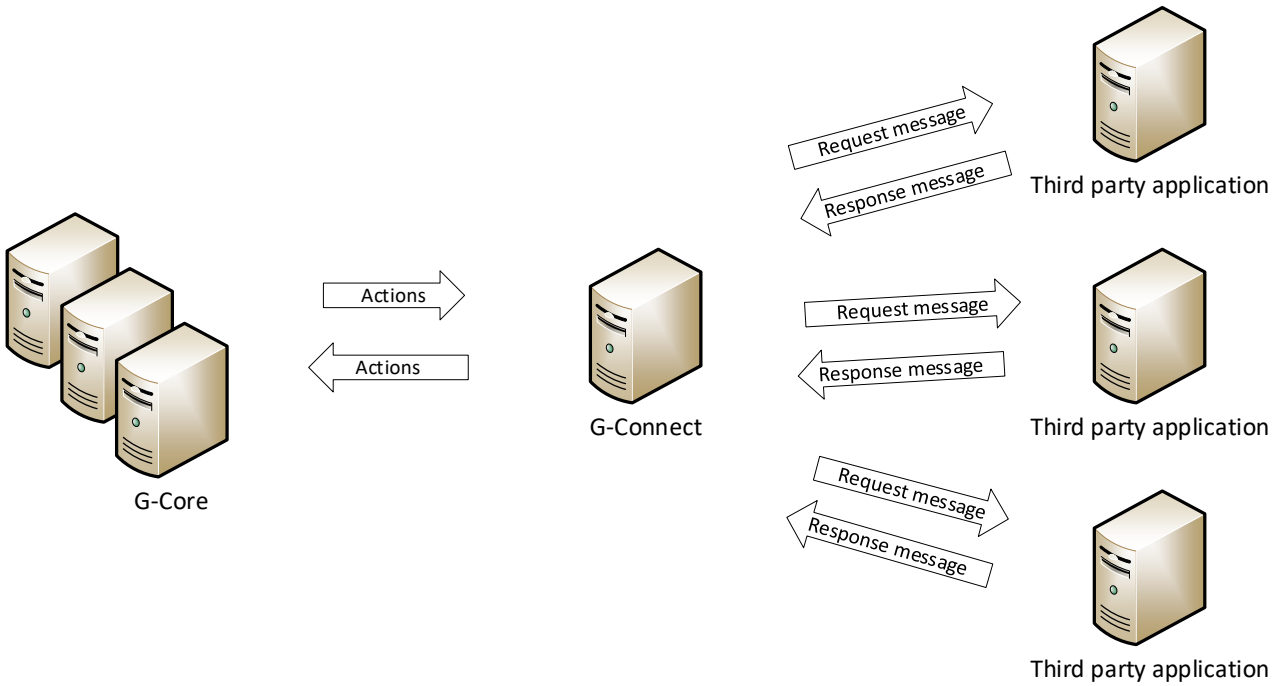

Figure 2: G-Core action triggered

2.2 Examples

A third party application analyses videos for special events. These shall be sent to a video recorder like G-Core to fire an alarm to an operator.
An ANPR camera connected to a video recorder like G-Core detects numberplates of vehicles. Requesting data from a third party application it is to be decided whether the vehicle is allowed to enter a company area. In addition recognized vehicles shall be signalized to the third party application.
A face recognition application analyses videos for dedicated people. If a person is detected, an operator shall be informed about the possible interests of this person.

2.3 Use cases

A truck arrives at a company. To enter the company ground, it has to pass a physical barrier. Its numberplate is scanned by an ANPR camera connected to G-Core. The created G-Core ANPR action is received by the G-Core-Plug-in of **G-Connect** service which creates a request message via a plug-in to the connected 3rd party application. This returns a response message containing the information whether the truck is allowed to pass or not. The service creates a G-Core action to inform an operator or a connected service about the result.

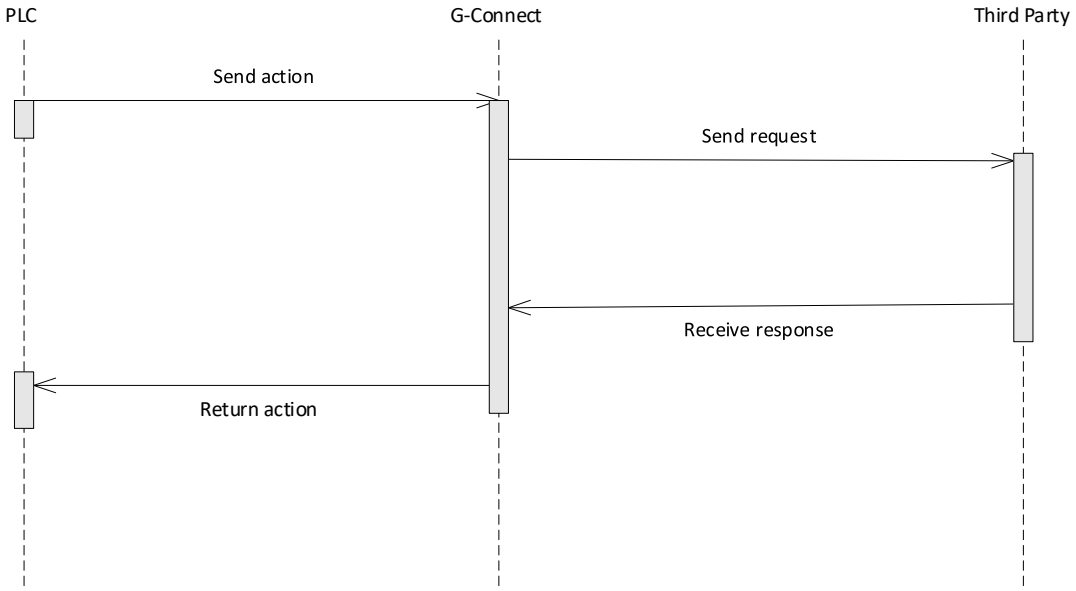

Figure 3: Use case action triggered

A 3rd party application connected to the **G-Connect** service sends a message (as file or via TCP/IP). The service extracts relevant data and creates a G-Core action containing it.

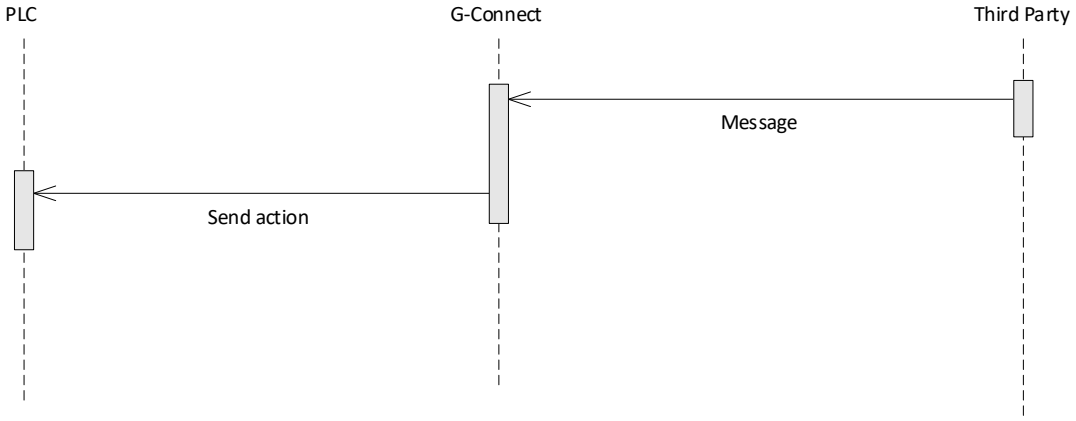

Figure 4: Use case 3rd party triggered

A 3rd party application connected to the **G-Connect** service sends a message (as file or via TCP/IP). The service extracts relevant data, requests additional data from the 3rd party application, does some additional work (e.g. saving data to a file) and creates a G-Core action containing it. The additional data request can be operated synchronously or asynchronously.

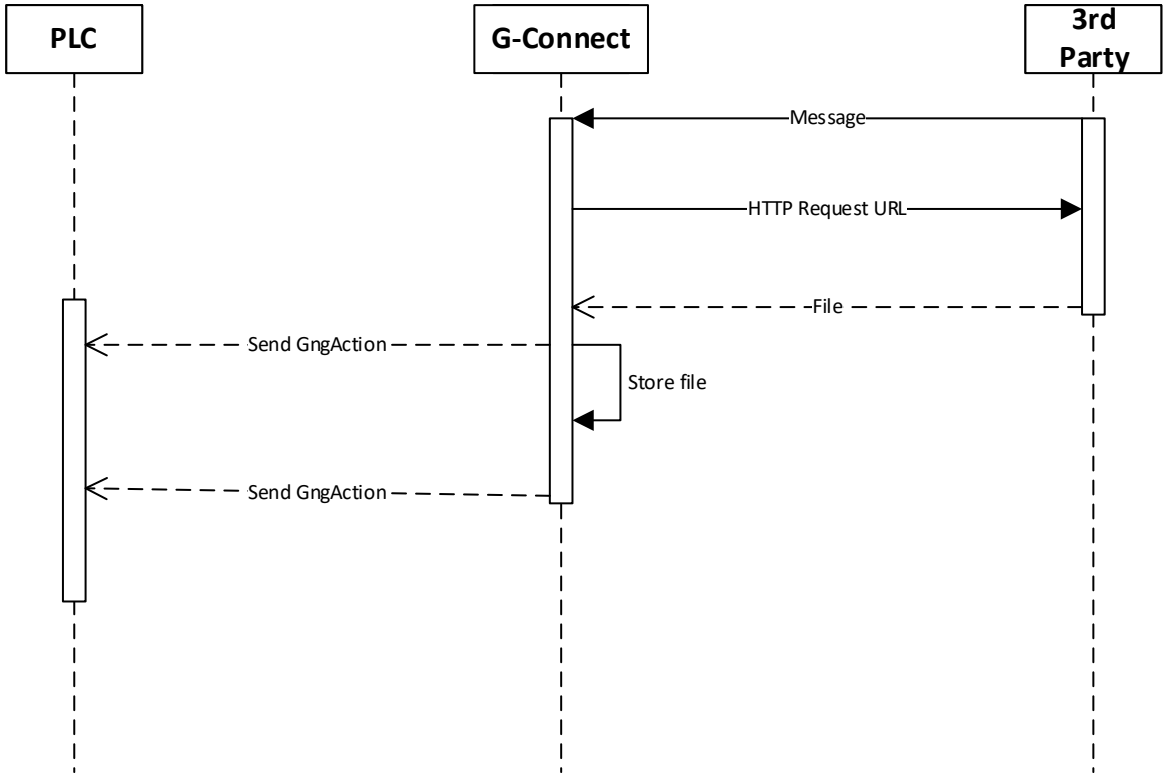

Figure 5: Use case 3rd Party driven file request and action creation

3 Architecture

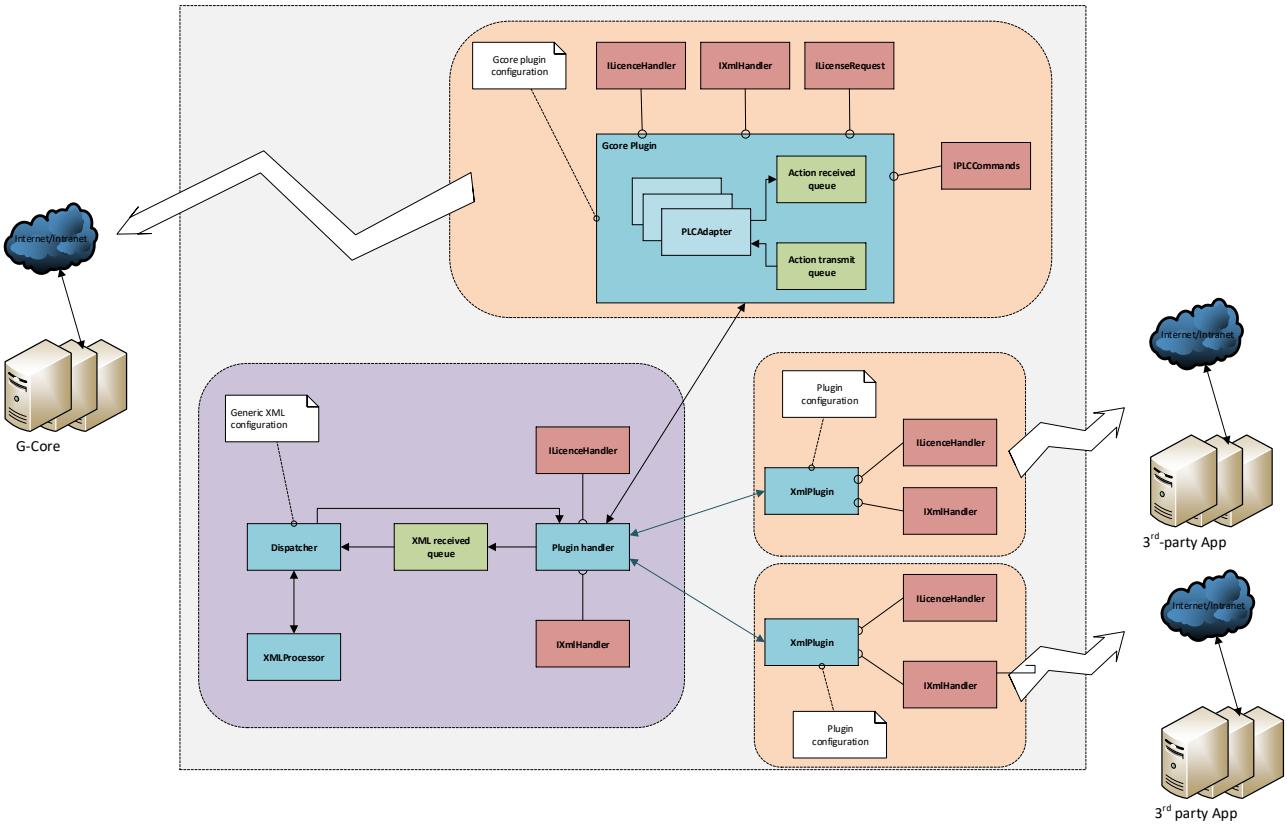

Figure 6: Architecture

The basic part in the architecture is the dispatcher. It organises the data transfer between the plug-ins. It also is responsible for processing the XML documents using the XML processor. Incoming data from plug-ins are queued and processed in related threads. Outgoing data is sent to the plug-ins where it is converted to the dedicated format and forwarded to the 3rd party application.. The system is highly configurable. Each plug-in has its own configuration containing connection details to the 3rd party applications and definition for converting the data into the inter-communication-protocol. A central configuration controls data transfer from one plug-in to another.

4 Installation

System requirements:

- Windows 8.1 64 bit or higher or
- Windows Server 2012 R2 64 bit or higher
- Microsoft .Net Framework 4.6.1 (contained in the installer package)

To start the installation process the file “G-Connect Service-x.x.x.x.exe” must be started with administrator rights.

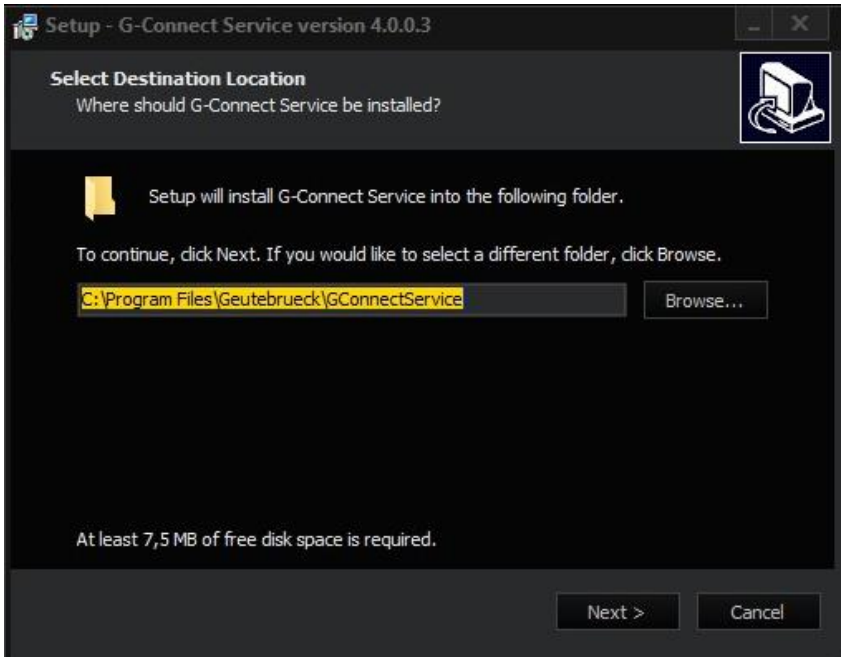

Figure 7: Recommended installation path

To keep the recommended installation path, just click „Next” to proceed with the next step. Otherwise change the path and then click “Next”.

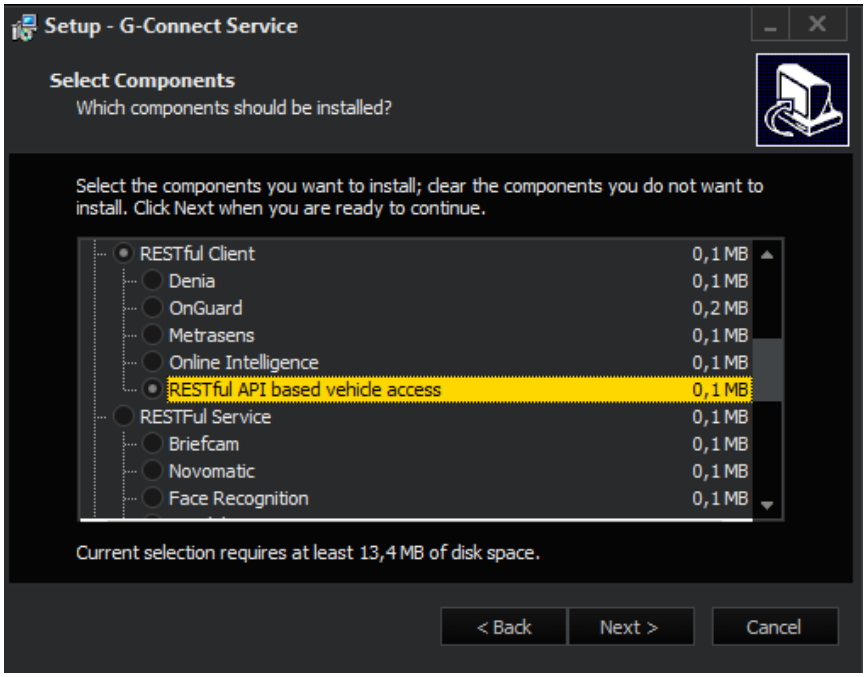

Figure 8: Select the components

You can select, for which project you want to install **G-Connect** service. The binaries will be installed anyway but only the special configuration files will be copied. After selection click “Next” to preceed.

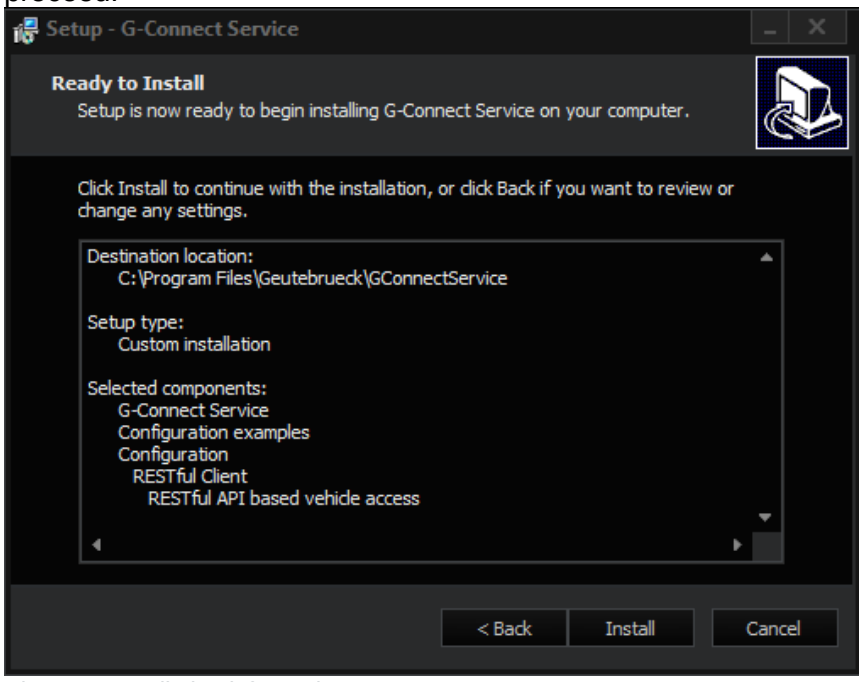

Figure 9: Installation information

To continue with installation just click “Install”.

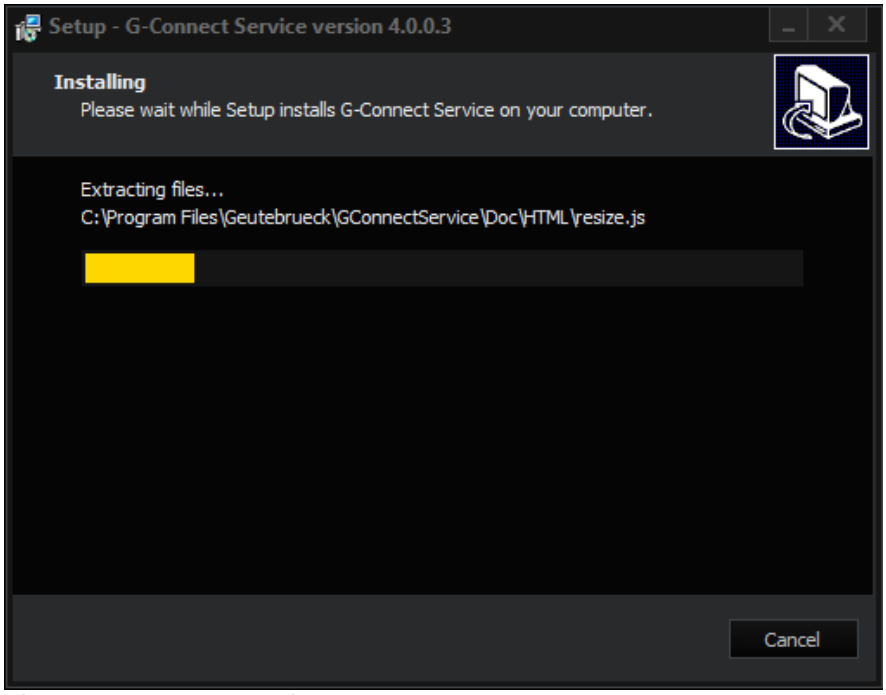

Figure 10: Setup progress bar

A progress bar indicates the status of installation.

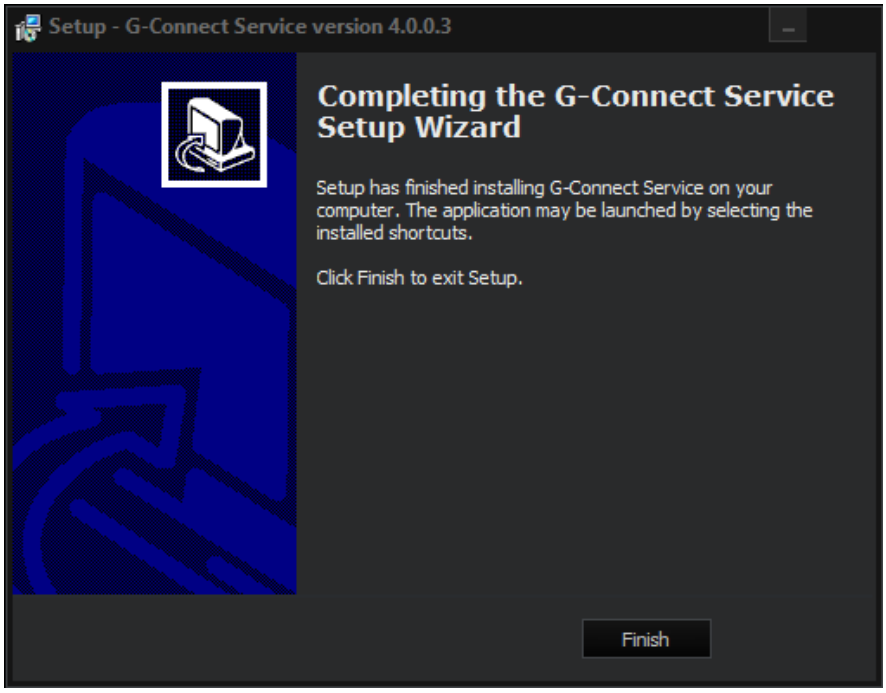

Figure 11: Completion screen

After installation has completed, click “Finish” to exit the installer.

5 Uninstallation

If the product is already installed, it can be re-installed or removed from the device.

Select an option and proceed.

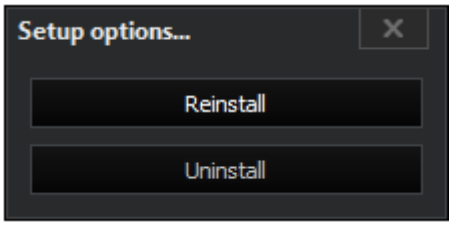

Figure 12: Setup options menu

If “Uninstall” is chosen, the service is completely removed from the device. A process bar shows up to indicate the state of uninstallation. After successful remove the following screen is displayed.

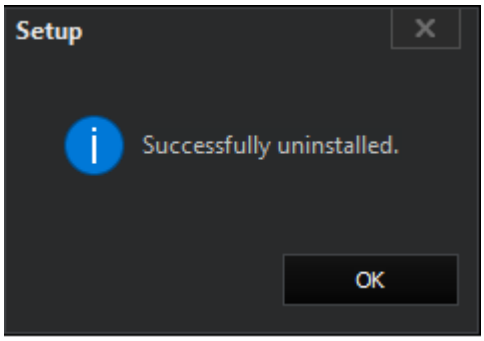

Figure 13: Uninstallation success message

The configuration will not be deleted nor overwritten by reinstallation.

6 Rule engine

The **G-Connect** service contains a rule engine for processing incoming messages from 3rd party applications. With help of rules the service calculates which message to be sent dependant on the content of the message data fields. A message field can be filled with varying data controlled by the content of the related data fields.
Rules are organized in rulesets to be linked to the mapping. Each rule results data to be sent by a message. A rule consists of *if* and *else* clauses, *AND* and *OR* conjunctions and condition phrases. Condition phrases compare two values of Int, String or DateTime types to each other.

⚠ As of version 1.1.1.53 it is possible to set different response message IDs to the *if* clause and the corresponding *else* clause. The response message ID defined in the rule has priority. But if it is set to 0, the response message defined in the *if* clause will be used when the clause calculates to true, otherwise, if defined, the response message linked to the *else* clause will be sent.
The following example may show the power of the rules engine:

An XML dataset contains the fields Y1, Y2, Y3 and Y4. Depending on the content of Y4 either Y2 or Y3 shall be filled into the ZonID field of the G-Core PPZoneAlarm action. The State field of the action shell be set to 1 if Y1 contains “MO” else to 0.

```
<Ruleset ID="1">
  <Rule MessageID ="PPZoneAlarm"> points to PPZoneAlarm action
    <IfClause>  starts an If clause. If clauses cannot contain additional If clauses
      <Conjunction> starts a conjunction of AND or OR phrases
        in this case only one condition has to be fulfilled
        <Condition Member="Y1" Operand="equal" Value="MO"/>
      </Conjunction>
      resulting member if condition is true
      <Member Name="ZoneState" Value="1" Type="Int32"/>
    </IfClause>
    <ElseClause> starts an else clause
      If and Else clauses can be nested without restriction
      <IfClause>
        <Conjunction>
          <Condition Member="Y1" Operand="equal" Value="CL"/>
        </Conjunction>
        <Member Name="ZoneState" Value="0" Type="Int32"/>
      </IfClause>
    </ElseClause>
  </Rule>
  <Rule MessageId="PPZoneAlarm">
    <IfClause>
      <Conjunction>
        <AND> Conjunctions can be nested without restriction
          <OR> if nested, the conjunction has to be written first
            <Condition Member="Y1" Value="MO" Operand="equal"/>
            <Condition Member="Y1" Value="CL" Operand="equal"/>
          </OR>
          <Condition Member="Y4" Value="CC" Operand="equal"/>
        </AND>
      </Conjunction>
      multiple members can be set simultaneously
      <Member Name="Subcell" Value="1" Type="Int32"/>
      <Member Name="ZoneID" Value="{Y3}" Type="String"/>
    </IfClause>
    <ElseClause>
      <IfClause>
        <Conjunction>
          <AND>
            <OR>
              <Condition Member="Y1" Value="MO" Operand="equal"/>
              <Condition Member="Y1" Value="CL" Operand="equal"/>
            </OR>
            <Condition Member="Y4" Value="  " Operand="equal"/>
          </AND>
        </Conjunction>
        <Member Name="Subcell" Value="0" Type="Int32"/>
        <Member Name="ZoneID" Value="{Y2}" Type="String"/>
      </IfClause>
    </ElseClause>
  </Rule>
</Ruleset>
```

In the XML configuration for the **G-Connect** service this will look like this:
Restrictions:

- *Else* clauses must be placed below *If* clauses.
- *If* clauses must not contain additional *If* clauses.
- *Else* clauses may contain additional *If* clauses.
- Resulting members must be placed below conjunctions.
- In *Else* clauses resulting members must also be placed below an additional *If*/*Else* construct.
- Conjunctions may contain single conditions or *OR* or *AND* conjunctions but only one out of these.
- *OR* and *AND* conjunctions also may contain multiple conditions and/or further *OR* or *AND* conjunctions. But an *AND* conjunction can contain only conditions and/or *OR* conjunctions. An *OR* conjunction can only contain conditions and/or *AND* conjunctions.
- Commutative law:
  - A *AND* B = B *AND* A
  - A *OR* B = B *OR* A
- Distributive law
  - A *AND* (B *OR* C) = (A *AND* B) *OR* (A *AND* C)
  - A *OR* (B *AND* C) = (A *OR* B) *AND* (A *OR* C)
- Associative law
  - A *OR* (B *OR* C) = (A *OR* B) *OR* C = A *OR* B *OR* C
  - A *AND* (B *AND* C) = (A *AND* B) *AND* C = A *AND* B *AND* C

⚠ As of version 1.1.1.106 request mapping entries may contain a ruleset id as well. This allows operating values received by an action or event with help of predefined methods and rules.

```
<?xml version="1.0" encoding="UTF-8"?>
<XMLConnectConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="XmlConnectConfig.xsd">
    <ThirdPartyConnection ID="XmlGCorePlugin" PluginPath="Plugins\XMLConnectGCorePlugin.dll"/>
    <ThirdPartyConnection ID="XmlMercareonPlugin" PluginPath="Plugins\XMLConnectMercareonPlugin.dll"/>
    <XPathExpressionList>
        <!-- Mercareon Expressions for searching in booking list -->
        <XPathExpressionSet ID="1">...</XPathExpressionSet>
        <XPathExpressionSet ID="2">
            <XPathExpression Expression="DictionaryProxyOfStringObject/KeysAndValues[Key='PlateTag']/Value" Member="PlateTag"></XPathExpression>
            <XPathExpression Expression="DictionaryProxyOfStringObject/KeysAndValues[Key='CountryCode']/Value" Member="CountryCode"></XPathExpression>
            <XPathExpression Expression="DictionaryProxyOfStringObject/KeysAndValues[Key='MediaChannel']/Value" Member="MediaChannel"></XPathExpression>
            <XPathExpression Expression="DictionaryProxyOfStringObject/KeysAndValues[Key='EventDateTime']/Value" Member="EventDateTime"></XPathExpression>
            <XPathExpression Expression="DictionaryProxyOfStringObject/KeysAndValues[Key='Template']/Value" Member="Template"></XPathExpression>
        </XPathExpressionSet>
        <XPathExpressionSet ID="3">...</XPathExpressionSet>
    </XPathExpressionList>
    <Predefined>
        <!-- Mercareon -->
        <Member Name="DateTimeNow" Value="#dt" Type="String"/>
        <!-- #dt will be exchanged by the current timestamp -->
        <Member Name="SeparatorDEU" Value="-" Type="String"/>
        <Member Name="SeparatorCZ" Value="  " Type="String"/>
        <Member Name="SeparatorDE" Value=" " Type="String"/>
        <Member Name="NormalisedPlateTag" Value="#NPRTemplate({PlateTag}, {Template}, {Separator})" Type="String"/>
    </Predefined>
    <Mappinglist>...</Mappinglist>
    <Rulesets>
        <!-- Evaluate for which reason the vehicle access is granted or denied -->
        <Ruleset ID="1">...</Ruleset>
        <!-- Evaluate which separator is used for normalizing the license plate -->
        <Ruleset ID="2">
            <Rule>
                <IfClause>
                    <Conjunction>
                        <Condition Member="CountryCode" Operand="equal" Value="DEU"/>
                    </Conjunction>
                    <Member Name="Separator" Value="{SeparatorDEU}" Type="String"/>
                </IfClause>
                <ElseClause>
                    <IfClause>
                        <Conjunction>
                            <Condition Member="CountryCode" Operand="equal" Value="CZ"/>
                        </Conjunction>
                        <Member Name="Separator" Value="{SeparatorCZ}" Type="String"/>
                    </IfClause>
                    <ElseClause>
                        <IfClause>
                            <Conjunction>
                                <Condition Member="CountryCode" Operand="equal" Value="DE"/>
                            </Conjunction>
                            <Member Name="Separator" Value="{SeparatorDE}" Type="String"/>
                        </IfClause>
                        <ElseClause>
                            <Member Name="Separator" Value="" Type="String"/>
                        </ElseClause>
                    </ElseClause>
                </ElseClause>
            </Rule>
        </Ruleset>
    </Rulesets>
</XMLConnectConfig>
```

The example shows the configuration for calculating a numberplate string from the received NPR event. The event fields contain the number plate without any spaces and special characters. The type field contains the evaluation mask for the plate tag. With help of a rule and a predefined method (#NPRTemplate) the field “NormalisedPlateTag” will be calculated containing spaces and special characters. E.g: NRGB123 with template “XX:XXZZZ” and countrycode “DEU” will result in “NR-GB 123”.

7 Password encryption

Passwords stored in the configuration files can be encrypted by the AES256 algorithm. The tool ***AesPasswordEncrypter.exe*** generates the AES256 encrypted string from the given password. It is essential to encrypt the password on the very same machine on which **G-Connect** is hosted.

8 Plug-ins

G-Connect uses a plug-in architecture to connect to various 3rd party applications. The basic service concentrates on processing XML documents while plug-ins are responsible for the communication channels. There are a lot of different communication protocols, proprietary or generally accepted. To meet requirements of a special communication channel a plug-in may be developed by Geutebrueck or even by a 3rd party provider.
The current installation contains some ready to use plug-ins.

⚠ As of version 4.0.0.44 the name of the configuration file can be added to the “ThirdPartyConnection” entry in the G-Connect configuration. Hence it is no longer necessary to stick to the hardcoded filename.

```
<ThirdPartyConnection ID="GCorePlugin" PluginPath="Plugins\GConnectGCorePlugin.dll" ConfigPath="Configuration\GConnectGCorePluginConfig.xml"/>
<ThirdPartyConnection ID="CargoClixPlugin" PluginPath="Plugins\GConnectCargoClixClientPlugin.dll" ConfigPath="Configuration\GConnectCargoClixPluginConfig.xml
```

Anyway if no configuration filename is added, the hardcoded filename will be used.
Additionally it is now possible to load multiple instances of a plug-in with different configurations.

8.1 GCore Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins

The GCore plug-in was introduced in version 3.0.0.1. It connects the **G-Connect** service to one or multiple G-Cores. It generates G-Core actions out of the transmitted data and converts action and event data into the internal transmission protocol and forwards it to the basic **G-Connect** service. It also provides access to the G-Core SAM service to let other plug-ins consume licenses. The GCore Plug-in configuration contains connection details for the G-Core servers and defines data fields to be converted.
To signalize the service being alive, it repeatedly sends a G-Core *CustomAction* with *Int* value 0 and *String* value “XML Service: live check” (⚠ As of version 4.0.0.6 it is “G-Connect Service: live check”). The repeat interval is set to 10 seconds by default and can be configured (See 10.5.1).

⚠ As of version 3.0.0.29 JPEG images can be sent to the G-Core server to a dedicated media channel. G-Core version must be 5.1.0 or higher. The media channels must be of type *DirectImageFeed*.

As of version 4.0.0.30 the plug-in retrieves the following additional data from a connected G-Core server:

- IP address
- MAC address
- Server GUID
- Firmware version
- Number of channels
- Number of inputs
- Number of outputs

The G-Core plug-in needs the configuration file "**GConnectGCorePluginConfig.xml**". The library filename is **GConnectGCorePlugin.dll**.

As of version 4.0.0.53 event images can be retrieved with an incoming event. These can be sent to other plug-ins. Therefore the attribute "LoadImage" of the element "GngEventCfg" must be set to *true*.

As of version 4.0.0.188 it is possible to define multiple action triggering an event.

As of version 4.0.0.213 for actions containing a TMediaChannelID parameter the internal enumeration "ParameterView" is considered. Hence such actions are created correctly.

As of version 4.1.0.5 the plug-in can create G-Core actions containing TResourceID typed parameters.

8.2 File Watcher Plug-in

Info: Not FailOver capable.
See chapter 12. ## Failover capable plugins.

This plug-in observes a configured path. Each time a new XML file is copied to this folder, the plug-in reads the contents, creates an XML document and provides it to the basic service. The processed file is deleted afterwards. At startup the plug-in operates all files located in this directory.

As of version 1.1.1.54 this plug-in can also process csv and json formatted files.

As of version 2.0.0.0 this plug-in can access shared network folders.

As of version 3.0.0.29 this plug-in can also read JPEG files. It converts a JPEG file into a base64 string and encapsulates it in an XML document below the elements **RootElementName** and **ItemElementName**. The image can be forwarded to another plug-in as e.g. GCore plug-in to send it via DirectImageFeed function to a G-Core channel.

As of version 4.0.0.202 an extension filter can be set to read only dedicated files. E.g. **.xml*, **.csv* or **.jpg*

As of version 4.1.0.4 the plug-in is able to register mpeg4 files for further operation. It sends path, filename and extension in a hardcoded xml structure to the core service.

Example:

```
<root>
    <item>
        <Fullpath>d:\Temp\mp4ToH264\Source\Recording_001.mp4</Fullpath>
        <Path>d:\Temp\mp4ToH264\Source</Path>
        <Name>Recording_001</Name>
        <Extension>.mp4</Extension>
    </item>
</root>
```

The file watcher plug-in needs the configuration file "**GConnectFileWatcherPluginConfig.xml**".
The library filename is **GConnectFileWatcherPlugin.dll**.

8.3 Telnet Plug-in

Info: Not FailOver capable.
See chapter 12. ## Failover capable plugins.

This plug-in provides a TCP/IP server for multiple client connections. It awaits XML, JSON or strings matching a defined regular expression at the configured port, creates XML documents out of them and provides these to the basic service. To access the provided values XPath expressions must be used.
In case the regular expression filter is used, the resulting XML document to be parsed has got the following format

```
<?xml version="1.0" encoding="utf-16"?>
<DictionaryProxyOfStringString xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <KeysAndValues>
                    <Key>0</Key>
                    <Value>Here always the whole matching string is contained</Value>
          </KeysAndValues>
          <KeysAndValues>
                    <Key>The group name given in the regular expression</Key>
                    <Value>The value matching the group in the regular expression</Value>
          </KeysAndValues>
          …Further entries for each group
</DictionaryProxyOfStringString>
```

An XPath expression for querying a dedicated value looks as follows:

```
DictionaryProxyOfStringString/KeysAndValues[Key='group name']/Value
```

The Telnet plug-in needs the configuration file "**GConnectTelnetPluginConfig.xml**".
The library filename is **GConnectTelnetPlugin.dll**.

8.4 File Provider Plug-in

Info: Not FailOver capable.
See chapter 12. ## Failover capable plugins.

The plug-in provides a configured file to be read by the main process. It can process xml, csv and json formatted files.

As of version 2.0.0.0 the file may be located in a shared network folder. Therefore domain. host user and password have to be configured.

As of version 3.0.0.16 the file content can be read into a local cache. Whenever the file changes the cache is synchronized.

As of version 4.0.0.202 the plug-in can read JPEG files as well. An extension filter can be set to read only dedicated files. E.g. **.xml*, **.csv* or **.jpg*

The file provider plug-in needs the configuration file "**GConnectFileProviderPluginConfig.xml**".
The library filename is **GConnectFileProviderPlugin.dll**.

8.5 Webservice File Save Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

This plug-in saves a provided file from an URL to a dedicated path. URL and destination paths are sent to the plug-in using a special XML file (see 10.5.5).

As of version 2.0.0.0 the destination path of the file may be located in a shared network folder.

The Webservice file save plug-in doesn't need any additional configuration file.
The library filename is **GConnectWebServiceToFilePlugin.dll**.

8.6 Face Recognition Plug-in

Info: Not FailOver capable.
See chapter 12. ## Failover capable plugins.

The plug-in provides a Web API self hosted server. Currently it only provides access to the url *http://<G-Connect-IP:9000>/FaceRec* and accepts POST messages in `x-www-form-urlencoded` format. The messages must contain at least two key/value pairs: Channel and XMLData. Splitting the data transported in the request body is done by operating a regular expression. Contained text is UTF-8 formatted and can be converted using the Codepage setting.

The face recognition client plug-in needs the configuration file "**GConnectFaceRecPluginConfig.xml**".
The library filename is **GConnectFaceRecPlugin.dll**.

8.7 File Copy Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

This plug-in copies files from source to destination which is controllable by an XML file. If the destination file already exists, it will be deleted before copying. In case of failure the plug-in tries to repeat deletion up to 10 times. By configuration it can be defined whether the source file shall be deleted after copying. The attempt to delete the file will be repeated up to 10 times in case of failure. It also can be configured to force clearing selected directories at start up of the service.

The plug-in does not consume any licence by itself but checks whether enough licences are consumed by the calling plug-in. For each different source/destination directory pair one licence must be consumed.

 As of version 1.1.1.104 source and destination directories may also be located on other machines. Such directories must be shared in the network. The configuration has to contain the sharing credentials.

 The file copy plug-in needs the configuration file “**GConnectFileCopyConfig.xml**”.
The library filename is **GConnectFileCopyPlugin.dll**.

8.8 File Writer Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

This plug-in writes received G-Core Action data to a csv type file. If the file doesn’t exist, it is created and a title row is added. The data is appended with each Action as long as the file still exists.

 As of version 2.0.0.0 the destination file may be located in a shared network folder.

 The file writer plug-in doesn’t need any additional configuration file.
The library filename is **GConnectFileWriterPlugin.dll**.

8.9 CarmenGo Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

CarmenGo is a simple to use Licence Plate Recognition application. Detection results can be read via a REST interface. The application provides a RESTful API server where a client can get data out of the database via GET requests.
The plug-in first reads all current entries of the database and calculates the highest dataset id. By polling it requests the next incoming dataset. If a new plate is recognized by CarmenGo, the plug-in reads the metadata of this event.
As of CarmenGo version 1.1.15 the behavior of the queries has changed. The URI *http://<hostname or IP>/api/database?limit=1* returns the most recent entry in the database. To check whether a new vehicle has arrived, this URI must be sent frequently.

 The CarmenGo plug-in needs the configuration file “**GConnectCarmenGoPluginConfig.xml**”.
The library filename is **GConnectCarmenGoPlugin.dll**.
As of CarmenGo version 1.4.0.9244 the interface has been changed. A login request with username and password must be called before data can be retrieved. The login returns a token which has to be sent with each data request. Therefore the behavior of the CarmenGo plug-in has been improved as of G-Connect 4.0.0.51.

8.10 Extract Picture Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

An XML file or document may contain one or multiple pictures compressed in base64 format. This plug-in extracts the picture and writes it into a file. The picture and the destination file path are provided with help of the CopyFiles.xml definition.

 As of version 2.0.0.0 the destination file may be located in a shared network folder. CopyFiles.xml must contain domain, host, user and password then. The password must be entered encrypted (see 7)

 The extract picture plug-in doesn't need any additional configuration file.
The library filename is **GConnectExtractPicturePlugin.dll**.

8.11 CargoClix Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

As a RESTful Web API client the service regularly requests a Vehiclelist from CargoClix and returns registered vehicles. Registrations are cached to bridge a lost connection.
The vehiclelist is requested by a GET call to *http://<CC-IP:CC-Port>/<Vehicles path>*. The body contained in the answer is structured like described in Linked Documents/Vehicle.pdf.

Registered vehicles are sent to CargoClix with a POST call to *http://<CC-IP:CC-Port>/Registration* containing the body structure described in Linked Documents/VehicleStatus.pdf.

 The CargoClix plug-in needs the configuration files "**GConnectCargoClixPluginConfig.xml**" and "**RESTAPI_ClientConfig.xml**".

 As of version 4.0.0.44 the plug-in configuration file contains a parameter for the additional configuration file name. Hence it is no longer necessary to stick to the hardcoded filename.
The library filename is **GConnectCargoClixClientPlugin.dll**.

8.12 RESTfull API Plug-in

Info: Not FailOver capable.
See chapter 12. ## Failover capable plugins.

This plug-in provides a RESTfull API currently containing only a POST method. The client has to call the URI *http(s)://<IP:Port>/api*. The body may contain XML or JSON formatted data. The format type must be set to either *application/xml* or *application/json*. In case of JSON formatted body it is converted to XML by the plug-in to operate it inside the G-Connect service. Therefore the elements JsonRoot and JsonItem are necessary.

 As of version 1.1.1.104 the plug-in also can operate POST requests of type *multipart/form-data*. Such a request is a bundle of different type data. This plug-in can handle *application/json* and *image/jpeg*. Images will be written into files located locally or in a shared network folder. The configuration has to contain the destination directory.

As of version 4.0.0.204 the plug-in also supports SSL and basic authentication.

As of version 4.0.0.210 a POST call returns 204 – NoContent if it was accepted and operated correctly.

As of version 4.0.0.215 the plug-in handles the content type x-www-form-urlencoded as string.

The RESTfull API plug-in needs the configuration file "**GConnectRESTServicePluginConfig.xml**".
The library filename is **GConnectRESTServicePlugin.dll**.

8.13 Briefcam Plug-in

Info: Not FailOver capable.
See chapter 12. ## Failover capable plugins.

Briefcam developed an application loading live or recorded videos to analyse the content for special events or objects such as people, cars or even dedicated faces. For alarm signalization to third party applications it uses a simple RESTful interface sending an HTTP Post message. The message has got the following format:
POST *http(s)://<ip:port>/api* with body

```
{
        "CameraName":"<camera name>",
        "AlertTime":"<timestamp of event occurrancy in ISO 8601 format>",
        "RuleName":"<rule name defined in Briefcam>",
        "ThumbnailUrl":"<url to download the picture>",
        "VideoUrl":"<url to download the video snippet>",
        "FaceIdentities":[{
                "ID":"<person id defined in Briefcam>",
                "Name":"<person name registered in Briefcam>",
                "ThumbnailUrl":"<url to download the picture>",
                "WatchListID":"<id of the defined watch list>",
                "WatchListName":"<name of the defined watch list>"
                }
        ]
}
```

The "FaceIdentities" field is only set when a face recognition rule is used.

The Briefcam plug-in needs the configuration file "**GConnectRESTServicePluginConfig.xml**".
The library filename is **GConnectBriefcamPlugin.dll**.

8.14 NightWatch Plug-in

Info: Not FailOver capable.
See chapter 12. ## Failover capable plugins.

LivAssured NightWatch is a medical analysis system for detecting epileptic seizure. The interface plug-in receives messages from the NightWatch device and creates CustomActionEx actions to be

used for event and alarm configuration in G-Core. The NightWatch API connects to the TCP socket provided by the **G-Connect** plug-in. It sends an announcement combined with a message. This is acknowledged by the plug-in. The message contains some data transported by an XML formatted body. See description below.

```
struct announcement
{
   char     magic[16];     // Communication Identifier
   char     clientId[16];  // Client MAC address
   uint32_t version;       // API version
   uint32_t flags;         // Flags (supported items)
   uint64_t timestamp;     // Epoch Timestamp
}; // Total 16 bytes
```

Figure 14: Announcement structure

```
struct message {
{
   uint32_t type;        // Message Type
   uint32_t size;        // Size of body, multiple of 8
   uint8_t  body[];      // message body
}; // Total 8+size bytes
```

Figure 15: Message structure

The body contains ether

```
<clientalive>
  <timestamp>20180810_223011</timestamp>
  <epochtime>156234637</epochtime>
  <macid>AABB1122CCDD</macid>
</clientalive>
```

Figure 16: clientactive message

or

```
<seizurealarm>
  <timestamp>20180810_223011</timestamp>
  <epochtime>156234637</epochtime>
  <macid>AABB1122CCDD</macid>
  <code>8</code>
  <desc>HartslagVerhoging</desc>
</seizurealarm>
```

Figure 17: seizurealarm message

or

```
<techalarm>
  <timestamp>20180810_223011</timestamp>
  <epochtime>156234637</epochtime>
  <macid>AABB1122CCDD</macid>
  <code>2048</code>
  <desc>LowBattery</desc>
</techalarm>
```

Figure 18: techalarm message

or

```
<dailystatus>
  <timestamp>20180810_223011</timestamp>
  <epochtime>156234637</epochtime>
  <macid>AABB1122CCDD</macid>
  <desc>ToBeDefined</desc>
</ dailystatus >
```

Figure 19: dailystatus message

The NightWatch API must be activated by the NightWach Web interface.

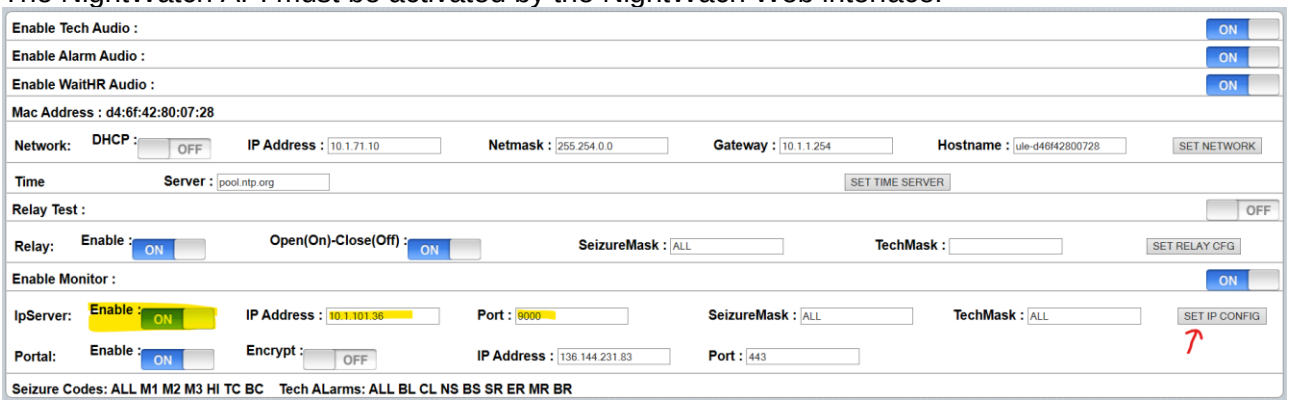

Figure 20: NightWatch Web interface

The used port must be defined in the plug-in configuration (see 10.5.12).
A complete API and user documentation is also available.

The Nightwatch plug-in needs the configuration file "**GConnectNightwatchPluginConfig.xml**".
The library filename is **GConnectNightwatchPlugin.dll**.

8.15 Mercareon Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

Mercareon by Transporeon GmbH is a logistic tool to control warehouse flow. For allowing third party applications to retrieve data from the system it provides XML files on an FTP server. The files contain - below others - a booking id, a vehicle plate tag and a status of operation. For each booking and each status a separate file is provided.
A single file looks like the following example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<booking>
   <booking_id>1107838842</booking_id>
   <booking_status_datetime>2019-12-10 05:50:33</booking_status_datetime>
   <location_id>232479012</location_id>
   <location_name>Koblenz</location_name>
   <group_id>232479651</group_id>
   <group_name>145</group_name>
   <license_plate>TF MB 93</license_plate>
   <dispatch>
      <status>
         <index>1</index>
         <datetime>2019-12-10 05:33:30</datetime>
      </status>
      <status>
         <index>2</index>
         <datetime>2019-12-10 05:50:32</datetime>
      </status>
      <status>
         <index>3</index>
         <datetime></datetime>
      </status>
      <status>
         <index>4</index>
         <datetime></datetime>
      </status>
   </dispatch>
</booking>
```

status/index 1 is set when a new vehicle is booked
status/index 2 is set when the vehicle can be operated. For vehicle access management applications this means a vehicle may get access to the site.
status/index 3 is set when operation of the vehicle is finished.
status/index 4 is not relevant.

The **G-Connect** Mercareon plug-in operates the files in the following manner:
At start it copies all files of the last n days (n can be configured) from the FTP folder to a local directory. This may also be a shared folder in the local network. If the local folder already contains files, all files older than the last n days will be deleted. It then operates file by file and fills a list sorted by the booking id containing the relevant booking data filtered by the status.
Every time a new file is created in the FTP directory it is copied to the local path an operated by the plug-in. It actualizes the list. If a booking reaches the status 3 (operation finished) it is removed from the list.
When a vehicle is recognized by the NPR camera, the **G-Connect** service checks its booking status and grants or denies access to the site.

The Mercareon plug-in needs the configuration file "**GConnectMercareonPluginConfig.xml**".
The library filename is **GConnectMercareonPlugin.dll**.

8.16 Kuando Busylight Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

Kuando Busylight is a small device for visual and acoustical signaling. It contains a light which can be coloured in the full RGB range and a speaker to play predefine sounds. Multiple devices may be connected to one machine.
The plug-in is designed to make use of the Busylight SDK version 3.0.0.8.

The plug-in provides the following commands with described parameters:

Command	Parameters	Description
Alert	Colour, Soundclip, Volume	The light blinks with the given colour, the sound clip is repeatedly replayed with the given volume. Acts on all connected Busylights
Alert2	Colour, Soundclip, Volume, [Devicenumber]	Same as Alert but plays the clip only once. RGB colour is not supported. If the device number is lower than the number of connected devices, the command only acts on the called device.
AlertAndReturn	Colour, Soundclip, Volume	Similar to Alert. RGB coulour is not supported. The command acts on all connected devices.
AlertSingleBL	Colour, Soundclip, Volume, Devicenumber	Same as Alert but acts only on a single device.
Blink	Colour, Ontime, Offtime	Turns the light on and off with the given colour for the given timeranges. The values for On- and Offtime are devided by 10 in seconds. E.g. an Ontime value of 10 keeps the light on for one second.
BlinkSingleBL	Colour, Ontime, Offtime, Devicenumber	Same as Blink but acts only on the dedicated device
ColorWithFlash	Colour, Flash	Switches between the two given colours repeatedly
ColorWithFlashSingleBL	Colour, Flash, Devicenumber	Same as ColorWithFlash but acts on the dedicated device
Jingle	Colour, Jingleclip, Volume	Plays the jingle once and blinks in the given colour
JingleSingleBL	Colour, Jingleclip, Volume, Devicenumber	Same as Jingle but acts only on the dedicated device
Light	Colour	Turns the light on with the given colour
LightSingleBL	Colour, Devicenumber	Same as Light but acts only on the dedicated device
Pulse	Colour	Pulse the light in the given colour frequently
PulseSingleBL	Colour, Devicenumber	Same as Pulse but acts only on the dedicated device
Terminate	none	Terminates all action of the Busylight devices. Hereafter the XMLConnect service has to be restarted to act on the Busylights again.

Colour values applies also to Flash

Value	
“Red”	red
“Green”	green
“Blue”	blue
“Yellow”	yellow
“Blank”	off
“RGB”	Red 0 - 255 Green 0 - 255 Blue 0 - 255

Soundclip values

Clip name
FairyTale
Funky
IM1
IM2
KuandoTrain
OpenOffice
TelephoneNordic
TelephoneOriginal
TelephonePickuMeUp
Quiet

Jingleclip values

Clip name
IM1
IM2

The XML document to be used for influencing Busylights needs the structure described in Linked Documents/Busylight.pdf.

The Kuando Busylight plug-in doesn’t need any additional configuration file.
The library filename is **GConnectBusyLightPlugin.dll**.

8.17 REST Client Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

The REST Client plug-in sends messages to a REST server using the URI format `http(s)://<SRV-IP or name>:<SRV-Port>/<command path>` and a JSON formatted body.

The REST client plug-in needs the configuration files “**GConnectRESTClientPluginConfig.xml**” and “**RESTAPI_ClientConfig.xml**”.
The library filename is **GConnectRESTClientPlugin.dll**.

As of version 4.0.0.191 this plug-in supports digest authentication.

8.18 TCP Client (UnityIS) Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

This TCP client plug-in is designed for connection to **IMRON UnityIS** ACS software. This software expects the following protocol:

Receive a command:
<command>(<user>) <message id> <parameter1> <parameter2>CR
Answer:
<4 byte length>(<message id>)%[<data>] or
<4 byte length>(<message id>)! <Error message> or
<4 byte length>(<message id>)% Processing… -> means data will follow

Unsolicited message form UnityIS:
<4 byte length>(E):<data> -> means message has a payload (in XML format (?))
<4 byte length>(S):<status message>
A **G-Core** action may contain a **UnityIS** command and some parameters. **G-Connect** adds the message id and the user and sends the request to **UnityIS**. If the answer contains data, it will be returned to **G-Core** by an action.
If **UnityIS** sends an unsolicited message (alarm or event), **G-Connect** forwards it to **G-Core** as an action.
The plug-in exposes the following message types:

- TCPClientMessageRequest – Request to UnityIS
- TCPClientMessageAnswer – Answer from UnityIS
- TCPClientMessageEvent – Event from UnityIS marked with “E”
- TCPClientMessageMisc – Miscellaneous message from UnityIS marked with “S” or other single characters

 As of version 3.0.0.40 (not yet in 4.0.0.x) a network disconnection may be recognized by sending a “Ping” message regularly.

 The TCP client plug-in needs the configuration file “**GConnectTCPClientPluginConfig.xml**”. The library filename is **GConnectTCPClientPlugin.dll**.

The TCP client plugin can handle following XMl settings

```
<?xml version="1.0" encoding="utf-8"?>
<GConnectTCPClientPluginConfig>
  <TCPHost Host="Host1" Port="1" User="User1" Password="Password1"
RcvBufferSize="4096" ReconnectTimeout="10000" CheckConnectionStatusInterval="100"
UsePing="true" />
  <CommandID>Command</CommandID>
  <Params>
    <ParamID>ParamID1</ParamID>
    <ParamID>ParamID2</ParamID>
    <ParamID>ParamID3</ParamID>
  </Params>
  <ResponseTimeout>1000</ResponseTimeout>
  <RegularExpression>RegularExpression1</RegularExpression>
</GConnectTCPClientPluginConfig>
```

8.19 VideoGuard Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

VideoGuard is a tool for collecting informations about VMS. It is introduced by SIRA (Security Industry Regulatory Agency) a governmental agency of Arabic Emirates.
To certify for bringing VMS devices into this market a connection to VideoGuard is mendatory.
The VideoGuard interface consists of a WCF service providing some basic methods for sending recorder information and alarms and retrieving connection data to a NTP server.
The WCF service is described in Linked Documents\Web-Service-Integration-for-Recorder-Alarms.pdf.
The plug-in uses the following hardcoded message types:

- SendAlarm – to plug-in, for sending web service alarms except recording server status.
- GetNTPServerInfo – to plug-in, for requesting NTP server information.
- GetNTPServerInfoResponse – from plug-in, transport NTP server information.
- SendRecorderInfo – to plug-in, for sending recording server information.
- SendHeartBeat – to plug-in, for sending a heartbeat message.
- ServerStatus – to plug-in, for sending a web service alarm containing the recording server status.

The VideoGuard plug-in needs the configuration file "**GConnectVideoGuardPluginConfig.xml**".
The library filename is **GConnectVideoGuardPlugin.dll**.

8.20 FLIR PSR Server Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

FLIR is a manufacturer of radar systems. The plug-in connects to the radar server. The plug-in implements a TCP client with an XML protocol. At startup it sends a request to retrieve the zone configuration. With help of the zone definition and the GPS coordinates of a detected target it can calculate in which zone the target is located. A target then can be identified by the sensor id, the zone id and the target id. Additionally the plug-in provides the target position as WGS84 coordinates, its distance from the sensor, its speed and its course.
Messages from the FLIR plug-in are sent with the ID “FlirMessage”. A complete structure description of the XML sent by the Flir radar server can be found in Linked Documents\AXMLInterface.pdf. G-Connect operates the element AXmlTarget as follows:

AXmlTarget	G-Connect / Flir plug-in	data type	range
id	TargetId	String	With Tracs: T<TracsId>.<TracsTargetId>, <RadarId>.<TargetId> Example: T1.1, 3.1 W/O Tracs: <RadarId>.<TargetId> Example: 32.123
status	TargetState	String	1 = detected, 0 = not detected (costing)
lt	Latitude	double	<Degree>.<Fract degree> Example: 50.585097

			positive values are located on the north side, negative values on the south side of the globe. 0° is the equator
lg	Longitude	double	<Degree>.<Fract degree> Example: 7.320658 positive values are located on the east side, negative values on the west side of the globe. 0° is Greewich, GB
elev	Elevation	Integer	Hight of the target in meters
range	Range	double	Distance from the sensor position in meters
bearing	Bearing	double	Azimuth angle related to north (0° to 360°)
elev_angle	ElevationAngle	double	Angle related to the horizon (0° to 90°)
course	Course	double	course vector angle related to north (0° to 360°)
speed	Speed	Integer	in km/h
quality	Quality	Integer	in percent
depth			
width			
snr			
rcs			
dop_speed			
dop_spread			
class	TargetClass	long	Type of the object
pclass	TargetPClass	long	Probability of type detection
pclass_data	TargetPClassData	String	Probability data of type detection
ts64	Timestamp	DateTime	Local date and time

Additionally data from Element AXmlZonesConfigReport is used:

AXmlZonesConfigReport	G-Connect / Flir plug-in	data type	range
id	SensorId	String	
AXmlZone.zone_index	ZoneId	long	

The FLIR plug-in needs the configuration file "**GConnectFlirPluginConfig.xml**".
The library filename is **GConnectFlirPlugin.dll**.

8.21 OnGuard Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

OnGuard is an access control system (ACS) manufactured by **Lenel**. Via the so called OpenAccess interface which is a RESTful and SignalR interface it offers the possibility to receive events and alarms, acknowledge event and send commands to influence the connected hardware. **G-Connect** offers the following features:

- Receiving hardware events,
- Acknowledging events,
- Sending generic events,
- Sending hardware commands and receiving responses (not used).

Received hardware events are stored in a local (MS SQL Lite) database. If an event shall be acknowledged by **G-Core**, it just sends the serial number of the event. The other data is taken from the database. The particular dataset is removed afterwards. For keeping the database small, each 12 hours it is checked for datasets older than 24 hours. In case such are removed. This means that after 24 hours an event cannot be acknowledged by **G-Core** anymore.
OnGuard needs a special Subscription-License for OpenAccess with a dedicated Application-Id.
The plug-in first retrieves the version information of **OnGuard** and afterwards requests all directory information configured in **OnGuard**.
The plug-in sends an authentication request and if succeeding it receives a token to be used by all further requests. If the plug-in successfully logged in, it subscribes to all hardware events. If any hardware event is recognized by **OnGuard**, it is automatically pushed to the **G-Connect OnGuard plug-in**. For certification purpose the plug-in implements an event counter to show whether any event may be lost.

 The OnGuard plug-in needs the configuration files "**GConnectOnGuardPluginConfig.xml**" and "**RESTAPI_ClientConfig.xml**".

 As of version 4.0.0.44 the plug-in configuration file contains a property for the additional configuration file name. It is no longer necessary to stick to the hardcoded filename.
The library filename is **GConnectOnGuardPlugin.dll**.

8.22 Run external processes Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

With help of this plug-in external processes can be started in the context of a user.

 The run external processes plug-in needs the configuration file "**GConnectRunProcessesPluginConfig.xml**".
The library filename is **GConnectRunProcessesPlugin.dll**.

 As of version 4.0.0.212 this plug-in is able to start a process triggered by an external event. The argument may contain variables closed in {}.

8.23 SQL database notification Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

If a row in a MS SQL Server database is changed or inserted, this plug-in may notify to create a message to another plug-in.
The database server needs the following settings:

- The Service Broker needs to be enabled.
- ANSI_NULLS ON
- ANSI_PADDING ON
- ANSI_WARNINGS ON
- ARITHABORT ON

- CONCAT_NULL_YIELDS_NULL ON
- QUOTED_IDENTIFIER ON

The SQL database notification plug-in needs the configuration file "**GConnectSQLPluginConfig.xml**".
The library filename is **GConnectSQLPlugin.dll**.

8.24 NTP client Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

The NTP client plug-in acts like an NTP client. Via the XML interface to the G-Connect service it gets the necessary data to request the current date and time from an NTP server:

- NTPServerHost
- NTPServerPort
- TimeZone
- UpdateInterval
- Deviation

TimeZone is an index into the following time zone table:

Details of Time Zone	Time Zone
UTC-12:00	1
UTC-11:00	2
UTC-10:00	3
UTC-09:30	4
UTC-09:00	5
UTC-08:00	6
UTC-07:00	7
UTC-06:00	8
UTC-05:00	9
UTC-04:30	10
UTC-04:00	11
UTC-03:30	12
UTC-03:00	13
UTC-02:00	14
UTC-01:00	15
UTC+00:00	16
UTC+01:00	17
UTC+02:00	18
UTC+03:00	19
UTC+03:30	20
UTC+04:00	21
UTC+04:30	22
UTC+05:00	23
UTC+05:30	24
UTC+05:45	25
UTC+06:00	26
UTC+06:30	27

UTC+07:00	28
UTC+08:00	29
UTC+08:30	30
UTC+08:45	31
UTC+09:00	32
UTC+09:30	33
UTC+10:00	34
UTC+10:30	35
UTC+11:00	36
UTC+12:00	37
UTC+12:45	38
UTC+13:00	39
UTC+14:00	40

The **UpdateInterval** defines the frequence to retrieve the current time and to synchronize the local clock.
Deviation determines the minimum difference in seconds between network time and local time to update the local clock.
Messages sent to the NTP client plug-in use the message id `SetNTPClientParams`.

The NTP client plug-in needs the configuration file "**GConnectNTPClientPluginConfig.xml**".
The library filename is **GConnectNTPClientPlugin.dll**.

8.25 Health Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

The health plug-in checks the status of locally connected hard drives frequently. It requests the following values for each hard drive from the system:

Value	Description
Index	Internal number of the hard drive starting with 0.
DeviceId	Internal name of the hard drive. E.g. \\.\PHYSICALDRIVE0
Model	Model name of the device
Signature	Signature of the device

The plug-in signals a fail (`HarddiskFailedMessage`) or an exchange of a hard disk (`HarddiskExchangedMessage`). A message also contains the hostname of the machine and the timestamp of occurrence.
The interval for checking the harddisks can be set by configuration.

The health plug-in needs the configuration file "**GConnectHealthPluginConfig.xml**".
The library filename is **GConnectHealthPlugin.dll**.

8.26 Load web content Plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

The “LoadWebContent plug-in” loads data from a given URL into memory and sends it to the basic service. It also gets meta data from the basic service and adds it to the received web content.

The load web content plug-in needs the configuration file "**GConnectLoadWebContentPluginConfig.xml**".
The library filename is **GConnectLoadWebContentPlugin.dll**.

As of version 4.1.0.22 the access to a UNC path can be authorized by authentication. If the url starts with ‘//’ the following text until the next ‘/’ is interpreted as hostname. If a username is defined in the configuration, the plugin tries to authenticate the user at the host. The password must be encrypted.

8.27 Message blocker plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

This plug-in checks incoming messages for special conditions. If these are fulfilled, the message is blocked. Otherwise it sent right back to the core.
Example: A number plate recognition system sends a recognized number plate several times as long as the car is in the detection area. To avoid calling a following action multiple times this plug-in only returns the first occurrence and blocks all following occurrences for a defined time.

8.28 Message combine plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

This plug-in combines messages with each other in groups. Incoming messages are sorted into the belonging group defined by the configuration. The plug-in waits until a group is completed an sends a message back containing data of all messages in the group. Afterwards the group is cleared and waits for the next set of messages.
Example:
An NPR recognition solution shall scan both the front and the rear plate of a vehicle. Only if both are recognized a message containing both plates is sent back.

A wait timeout delays sending of the back message for some time to be able to receive the messages multiple times even if all messages of the group are received at least once. This improves the accuracy of the recognition. A receive timeout causes sending a back message even if at least one message of the group isn’t received yet. This avoids waiting eternally if a message is never received. The waiting timeout is prioritized higher, i.e. if the receive timeout is expired but the waiting timeout is not, no back message is sent until the waiting timeout is expired as well. If the receive timeout is set to 0, the plug-in will wait until all messages of the group are received.

8.29 NEDAP AEOS plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

This connection is done with the web socket option from NEDAP AEOS 2021.2.
The AEOS server opens a port (8035) and the G-CONNECT NEDAP plugin connects to it.

This plug-in can handle all messages coming from AEOS like events, status, response, exceptions and infos.
This plugin can send commands to AEOS server.

General events from AEOS:
AEOS general events are separated into following parameters:
The corresponding NEDAP plugin parameters are below in bold font.

<Sequence nr> : Sequence number. Numeric field, varies from 1 .. 999
SeqNum
<AEOS> : Text is 'AEOS', in case event is generated by AEOS.
If text in event is not AEOS, event will be discarded.

<BC Type> : Behavior component type.
BehaviourComponentType

<Service key> : Combination of host name and the component where this event comes from.
ServiceKey

<Event number> : Numeric value, identifies the event. For entire list, see "aeos_generic-events.pdf" from NEDAP.
EventNumber

<date> : date in format: yyyymmdd
<time> : Time in format: hhmmss
combined in **EventTime** format: yyyy/mm/dd hh:mm:ss

<badge info> : Optional, contains badge info for certain events
BadgeInfo

<Carrier info> : Optional, contains carrier info for certain events
CarrierInfo

<Department name> : Optional, contains department name for certain events
Departmentname

<Direction info> : Optional, 0=No direction;1=In;2=OUT
Direction

<Reserved> : Optional, Reserved for future use.
AEOS13

And
ACSDATA which is a combination of **EventNumber** + **BadgeInfo**

Event type 1015 will generate a MessageId="ACSAccessGranted".
All events with BadgeNoAccessEvent will generate a MessageId = "ACSAccessDenied".
All other events will generate a MessageId = "ACSRawDataPush".

Other messages from AEOS:
status, response, exception and info, coming in string format to the variable **NEDAPDATA.**

MessageId = "CustomAction";

Commands to AEOS:
This plugin can as well send commands to NEDAP AEOS.
With MessageId="CustomActionEx" you can use the parameter “Command” to send a complete string to AEOS.
The syntax must be like executeArgs(singleShot, [true, 1000], "aepu01:RO1");
See NEDAP AEOS documentation aeos_socketinterface.pdf.

The document ConfigurationNedapGCore.pdf describes the handshake between NEDAP and GCORE.

The Nedap plug-in needs one G-Connect Nedap license (8.32003) per connection to a G-Core.

8.30 SAP SOAP Client

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

This plug-in connects to an SAP SOAP service by Leogistics. The data to be transported consists of 3 data fields:
`TIMESTAMP, LOADINGPOINTID, STATUS.`

As of version 4.1.0.6 the field `YDID` was added.

8.31 SQLite database

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

The SQLite database plug-in handles SQLite databases. Multiple databases can be created and managed. Each database has its own set of insert, select and delete commands configured in the dedicated configuration files. Additionally a cleanup sequence can be defined to maintain the database and delete obsolete datasets.

8.32 Process XML-files

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

This plug-in works similar as the FileWatcher. It watches a dedicated path and reads a newly created XML file into the memory. Other than the FileWatcher it processes XML files with configured XPath expressions and creates a message per queried dataset. If namespaces are

used in the XML file, they can be declared in the configuration, so that abbriviations can be used in the XPath expressions.

8.33 CIAS IB server plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

CIAS IB server collect data from different perimeter protection devices.
CIAS plug-in is connected to one IB servers. This server supports up to 128 (from 0 to 127) IBDevices. IBDevice refers to any device connected to IBServer.
Several CIAS plug-in can be installed at one G-Connect server.
Main duty of CIAS plug-in is to transfer all kinds of alarms generated by IBDevices. Alarm types include PREALARM, ALARM, TAMPER, FAULT, NO_ANSWER.

Basically IBServer is polling each IBDevice periodically. And this polling information is read, parsed and decrypted by CIAS plug-in.

CIAS reporting following events:
PREALARM
ALARM
FAULT
TAMPER
NO_ANSWER

The events are mapped in XML configuration:
PREALARM / ALARM -> PPDeviceAlarm (InterfaceID: "172.28.38.186", DeviceAddress: 4, Sensor: 4, State: 1)
FAULT / TAMPER -> PPDeviceAlarm (InterfaceID: "172.28.38.186", DeviceAddress: 4, Sensor: 2, State: 1)
NO_ANSWER -> PPDeviceOffline (InterfaceID: "192.147.168.129", DeviceAddress: 4)

InterfaceID: IP of IB server
DeviceAddress: Number of sensor
Sensor: 0= Cable Fault A; 2 = tamper; 4= sensor
State: 0= normal; 1= alarm

If connection to IP server is not possible an action in send to GCORE VMS:
SystemError (Source: 22, Message: 3, WindowsError: 0, Description: "CIAS description", XMLInfo: "CIAS")

If license for G-Connect CIAS plug-in is not available an actions is send:
SystemError (Source: 22, Message: 2, Description: "Not licensed", XMLInfo: "CIAS")

The plugin uses a configuration file to set the CIAS IB servers and the maximum attempts to connect to servers in case of communication failure.
The configuration files is named `GConnectCiasPluginConfig.xml`.
In this file there are four sections: `TCPMainHost`, `TCPBackupHost`, `RetryOnFail` and `TimeToWaitOnFailSeconds`.

`TCPMainHost` and `TCPBackupHost` are used to set up the main CIAS IB server and the backup (if needed), compound of Host and Port respectively.
- `TCPMainHost`: The main CIAS IB server.
        <TCPMainHost Host="192.168.10.102" Port="10000" />
- `TCPBackupHost`: The backup CIAS IB server.
        <TCPBackupHost Host="192.168.10.103" Port="10000" />
```
- `RetryOnFail`: The maximum number of retry connections in case of communication failure. This will be used only at plugin start up. Once the plugin starts and connects successfully once every further communication failure it will be try only one time to connect main or backup server.
- <RetryOnFail>3</RetryOnFail>
```
- `TimeToWaitOnFailSeconds` : The time in seconds to wait, when a notification of server disconnected is received before trying to connect to the other.
- <TimeToWaitOnFailSeconds>1</TimeToWaitOnFailSeconds>

If main IB server gets disconnected the plug-in immediately (network timeout) switch to backup server.
The plugin-in will switch back to main server if it is available again automaticly.

8.34 Fibersensys APU plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

Fibersensys plug-in is an input driver which is designed to take events/alarm from Fiber SenSys alarm processing units(APU). The driver works using a TCP connection and this connection provides fast communication with other drivers. The plug-in has up to 128 channels and each of these channels support one APU module. All of the modules works independent from the other and has own parameters to connect to device.

Fibersensys APU has been developed and tested with FD3xx-IP driver.

There are 3 types of events in Fibersensys;
• Intrusion or alarm
• Fault (broken fiber or hardware malfunction)
• Tamper (a switch which indicates whether the device has been opened).

Events that are coming from FiberSenSys APU’s are named like; DEVICENAME_DEVICESTATE, where DEVICENAME is the channel name of driver.
To give an example:
Devicename is APU1:
APU1.CHb_ALARM
APU1.CHb_FAULT
APU1.Cha_TAMPER

In XML configuration file these events are mapped to actions:
ALARM -> PPDeviceAlarm (InterfaceID: "172.28.38.186:10000", DeviceAddress: 4, Sensor: 4, State: 1)
FAULT-> PPDeviceAlarm (InterfaceID: "172.28.38.186:10000", DeviceAddress: 4, Sensor: 0, State: 1)

TAMPER -> PPDeviceAlarm (InterfaceID: "172.28.38.186:10000", DeviceAddress: 4, Sensor: 2, State: 1)

InterfaceID: IP and port of APU
DeviceAddress: Number of channel of an APU
Sensor: 0= Cable Fault A; 2 = tamper; 4= sensor
State: 0= normal; 1= alarm

IF APU is not reachable at all an action is cend to GCORE VMS:
PPDeviceOffline (InterfaceID: "172.28.38.186", DeviceAddress: 4)

If connection to APU is not possible an action in send to GCORE VMS:
SystemError (Source: 22, Message: 3, WindowsError: 0, Description: " Fibersensys description", XMLInfo: " Fibersensys")

If license for G-Connect Fibersensys plug-in is not available an actions is send:
SystemError (Source: 22, Message: 2, Description: "Not licensed", XMLInfo: "Fibersensys")

The plug-in using the following dlls: `FSI.DeviceSDK.dll` v2.3.0.22255, `FSIDeviceCommunicationLibrary.dll` v2.2.0.18366 and `FSIUtilitiesLibrary.dll` v2.0.0.0.

The Fibersensys GConnect plugin uses a xml configuration file named GConnectFibersensysPluginConfig.xml to retreive the configuration and behave according to that. This section describes the configuration variables that can be tunned in the configuration file.

- `RetryOnFail`: Value that states how many times to retry when trying to connect to APU devices.
<RetryOnFail>3</RetryOnFail>
```
- `ReconectWaitInLoopSeconds`: After plugin start a background task is triggered to verify that all APU devices are correctly connected. This value specifies how many seconds the task should sleep before run again.
<ReconectWaitInLoopSeconds>3</ReconectWaitInLoopSeconds>
```
- `ReconectDelaySeconds`: After each failure connection with a particular APU device, this variable specifies the delay in seconds before connect to the broken APU device. This also prevents that 3rd party application recognize the plugin behavior as a DoS attack.
<ReconectDelaySeconds>3</ReconectDelaySeconds>
```
- `TimeoutAPUConnectSeconds`: Amount of time in seconds before end the connection attempt to APU device for timeout.
<TimeoutAPUConnectSeconds>5</TimeoutAPUConnectSeconds>
```
- `TCPHostList`: This encapsulates the list of APU devices that the plugin should connect to. It can be specified as much devices are needed up to 128. Each device is specified by the tag `TCPHost` where Ip address is entered.

```
  <TCPHostList>
                <TCPHost Host="192.168.10.102"/>
                <TCPHost Host="192.168.10.103"/>
  </TCPHostList>
```

8.35 Platesmart plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

Overview
GConnectPlatesmartPlugin is a plugin to integrate Platesmart into G-Connect.
Platesmart provide information about licence plate of cars. As addition they do have access to a national database for licence plates. This database can provide information about the owner of the car in terms of criminal violations.

Configuration files
GConnectPlatesmartConfig.xml should be inside Configuration folder, inside the G-CONNECT installation folder. This file will have essentially the configuration of the endpoints that will be set in the Platesmart Server for the publishing of the ALPR Events (Standard LPR Recognition) and Watchlist Alert Event (Extended LPR Recognition)

GConnectPlatesmartConfig.xml
```xml
<?xml version="1.0" encoding="UTF-8"?>
<GConnectPlatesmartPluginConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

	xsi:noNamespaceSchemaLocation="GConnectPlatesmartPluginConfig.xml">

	<!-- Added RestClient configuration for images download-->
	<AdditionalConfigurationFile>Configuration\RESTAPI_ClientConfig.xml</AdditionalConfigurationFile>

	<ImageTransferenceConfiguration>
		<NameOfRootElement>root</NameOfRootElement>
		<NameOfItemElement>item</NameOfItemElement>
	</ImageTransferenceConfiguration>

	<!-- Endpoints configuration -->
	<!-- If Username is provided then the authorization header should be sent -->
	<EventReceiverService>
		<CredentialsConfiguration>
			<!-- Username and Password used for endpoint authentication/authorization process  -->
			<User>username</User>
			<Password>password</Password>
			<IsEncryptedPassword>false</IsEncryptedPassword>
		</CredentialsConfiguration>
		<EndpointConfiguration>
			<!-- Because this endpoint should be accessed remotely,
			the IP Address should be used and never can be
			LOCALHOST or 127.0.0.1  -->
			<Host>Host_IP_Address</Host>
			<!-- A free port should be specified  -->
			<Port>Port</Port>
			<Route>api/{controller}/{action}/{id}</Route>
			<!-- Set this value to true if a secure connection would be used  -->
```

```
                        <UserSSL>false</UserSSL>
                </EndpointConfiguration>
        </EventReceiverService>
</GConnectPlatesmartPluginConfig>
```

Configuration elements descriptions
- **AdditionalConfigurationFile:** It is used for specifying the path for the RestApiClient config file.
- **ImageTransferenceConfiguration:** It is the structure names that will contain a downloaded image inside a XmlDocument for being transferred to GCore
- **EventReceiverService:** It will contain the endpoint configuration where Platesmart Server will publish the Events and the Alerts.

  - **CredentialsConfiguration:** If the parameters of this section are set, them the endpoint will demand a authentication process to every request made to him.
    - Username: Username needed for endpoint authentication
    - Password: Password needed for endpoint authentication
  - **EndpointConfiguration**
    - Host: G-Connect host IP
    - Port: Port where will be published the endpoint.
    - Route: Configuration route for the endpoint
    - UserSSL: Define if SSL will be used to access the endpoint.

The platesmartplugin endpoint routes looks like this:
- http://{ip}:{port}/api/v1.0/platesmart/event
- http://{ip}:{port}/api/v1.0/platesmart/alert

Http Client
As a piece of all the plugin configuration, in the same folder should exists the RESTAPI_ClientConfig.xml file. The HttpClient configuration is needed for the downloading process of the images related to the evens.

RESTAPI_ClientConfig.xml
```xml
<?xml version="1.0" encoding="UTF-8"?>
<REST_API_ClientConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="RESTAPIClient.xsd">
        <!-- Name or IP address of the Platesmart Server -->
        <ServerIP>Platesmart_Image_Storage_Server_IP</ServerIP>
        <!-- Port of the server. In case the default port 80
        shall be used, this element maybe deleted -->
        <ServerPort>Port</ServerPort>
        <!-- For basic authentication a username must be added here.
        If no authentication is needed, this element maybe left blank or deleted -->
        <ServerUser>sysadmin</ServerUser>
        <!-- If needed the password must be encrypted.
        Use the AesPasswordEncrypter.exe tool. If not needed,
        this element maybe left blank or deleted -->
        <ServerPassword>masterkey</ServerPassword>
        <!-- Set to true if SSL encrypted transport shall be used.
        Default value is false. If no SSL encrypted transport
        is needed, the element maybe deleted -->
        <UseSSL>false</UseSSL>
```

```
        <!-- List of destination URLs. The ID references the command set in the
XMLConnect...PluginConfig.xml -->
        <URLList>
                <URL ID="Snapshot" URLString="api/v1.0/image/durable/camera-
preview/{cameraId}" Command="GET" ContentType="image/jpg"/>
                <URL ID="Source" URLString="api/v1.0/image/transient/alpr/{sourceId}/{imageId}"
Command="GET" ContentType="image/jpg"/>
                <URL ID="Cropped" URLString="api/v1.0/image/transient/alpr-
plate/{sourceId}/{eventId}" Command="GET" ContentType="image/jpg"/>
        </URLList>
        <!-- List of accepted content types. Defines the data formats to be accepted by the client -->
        <AcceptedContentTypes>
                <AcceptedContentType>application/json</AcceptedContentType>
        </AcceptedContentTypes>
</REST_API_ClientConfig>
```

G-ConnectConfig.xml

```xml
<Predefined>
   <Member Name="GCoreAdapter" Value="Local" Type="String"/>
   <Member Name="DIF1" Value="20" Type="Int64"/>
   <Member Name="DIF2" Value="43" Type="Int64"/>
   <Member Name="DIF3" Value="44" Type="Int64"/>
   <Member Name="Int" Value="0" Type="Int32"/>
</Predefined>
```

It is necessary to pay attention to the Predefined section. **DIF1**, **DIF2** and **DIF3** are the channels IDs for the Direct Image Feed. The Media Channel ID defined in GCore must by set in the **Value** property.

Notes

It is necessary to execute G-Connect as administrator rights to allow the creating of health service endpoint

GConnectPlatesmartPlugin workflow

1. The process begins when Platesmart Server sends an Event (LPR Standard Recognition) or an Alert (LPR Extended Recognition) to the endpoint published by the plugin.
2. GConnectPlatesmartPlugin sends a PlateSmartAlert () or PlateSmartEvent () action into the system
3. GConnectPlatesmartPlugin downloads the tree images corresponding to the received event
4. Sends each image to the corresponding DIF channel

Example:
PlateSmartAlert (PlateNo: "ABC123", Make: "BMW", Type: "Sedan", Alert: "Stolen", Color: 6, MediaChannelID: 11)

PlateSmartEvent (PlateNo: "ABC123", Make: "Mercedes", Type: "Truck", Color: 3, MediaChannelID: 9)

The Platesmart Plugin uses custom define actions.
The file PLC-test.xml is needed for all clients who access the GCORE server.

This file must be located at “C:\Program Files\Geutebrueck\Gcore” also on a remote machine. If G-Connect is not installed on the same PC where G-Core is installed then the file PLC-test.xml has to be copied to that location and the G-Core server needs to be restarted after.

From Platesmart expected JSON:

Sending an EVENT (LPRStandardRecognition)

{{url}}/api/v1.0/platesmart/event/


```
{
    "id": "99b880ffc8a84bd1b4655ceba9eb9247",
    "type": "alpr",
    "version": "1.0",
    "timestamp": 1707818685000,
    "image": {
        "id": "4dc9b3e194fc4059b5ba9473d1801945",
        "width": 1920,
        "height": 1080
    },
    "device": {
        "id": "6ea565734ea8",
        "type": "example_vms_2019v2"
    },
    "source": {
        "id": "3b596866d23340069b3cd7e46d06a1f5",
        "type": "alpr_processor",
        "name": "2 Backdoor"
    },
    "plate": {
        "tag": "ABC123",
        "code": "US-FL",
        "region": {
            "x": 50,
            "y": 50,
            "width": 200,
            "height": 100
        }
    },
    "vehicle": {
        "bearing": 0.0,
        "occlusion": 0.0,
        "color": {
            "code": "black"
        },
        "make": {
            "name": "Mercedes-Benz",
            "code": "mercedes_benz"
        },
        "orientation": {
            "name": "Rear",
            "code": "rear"
```

```
        },
        "type": {
            "name": "Sedan",
            "code": "sedan"
        }
    },
    "location": {
        "latitude": 0,
        "longitude": 0
    }
}
```


Sending an ALERT (LPRExtendedRecognition)

{{url}}/api/v1.0/platesmart/alert/


```
{
    "id": "721601d90f8f4f9194a0d3aaa62cc5db",
    "timestamp": 1707898739,
    "type": "watchlist",
    "version": "2.0",
    "expectedPlateCode": "US-FL",
    "expectedPlateTag": "ABC123",
    "description": "STOLEN VEHICLE",
    "policyId": "b066a8e6ad3d462c81a36ca816c0df18",
    "policyName": "Example Watchlist Name",
    "tags": [],
    "source": {
        "id": "api_60f1ccb87c6d4adfbb44612751cce8e0",
        "description": "Example Data Connection Description",
        "name": "Example Data Connection Name"
    },
    "metadata": {
        "bls": "cf-crime",
        "bld": "ASST.CRIM"
    },
    "event": {
        "id": "99b880ffc8a84bd1b4655ceba9eb9247",
        "type": "alpr",
        "version": "1.0",
        "timestamp": 946645199000,
        "image": {
            "id": "4dc9b3e194fc4059b5ba9473d1801945",
            "width": 1920,
            "height": 1080
        },
        "device": {
            "id": "6ea565734ea8",
            "type": "example_vms_2019v2"
        },
        "source": {
```

```
            "id": "3b596866d23340069b3cd7e46d06a1f5",
            "type": "alpr_processor",
            "name": "1 Example Camera Name"
        },
        "plate": {
            "tag": "ABC123",
            "code": "US-FL",
            "region": {
                "x": 50,
                "y": 50,
                "width": 200,
                "height": 100
            }
        },
        "vehicle": {
            "bearing": 0.0,
            "occlusion": 0.0,
            "color": {
                "code": "BROWN"
            },
            "make": {
                "name": "Mercedes-Benz",
                "code": "mercedes_benz"
            },
            "orientation": {
                "name": "Rear",
                "code": "rear"
            },
            "type": {
                "name": "Sedan",
                "code": "sedan"
            }
        },
        "location": {
            "latitude": 0,
            "longitude": 0
        }
    }
}
```

8.36 Bastille plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

Overview
GConnectBastillePlugin is a plugin to integrate Bastille into G-Connect.
Bastille's mission is to continue to expand the capabilities of the leading Wireless Threat Intelligence Platform. Bastille makes invisible connections and threats visible.

Configuration files
GConnectBastilleConfig.xml should be inside Configuration folder, inside the G-CONNECT installation folder. This file will have essentially the configuration of the endpoints that will be receive the streaming of the device events sent by the webhooks configured in the Bastille Enterprise Admin Console (such as device detections, geofence breaches).

GConnectBastilleConfig.xml
```xml
<?xml version="1.0" encoding="UTF-8"?>
<GConnectBastillePluginConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="G-ConnectBastillePluginConfig.xsd">

  <!-- Plugin Configuration  -->
  <BastilleServerConfig>
    <!-- Interval Value -->
    <!-- The interval value is the time to wait for the next check of time elapsed from the last http
request or heartbeat   -->
    <!-- Interval is expressed in seconds   -->
    <Interval>10</Interval>
  </BastilleServerConfig>

  <WebApiService>
    <CredentialsConfiguration>
      <!-- Username and Password used for endpoint authentication/authorization process  -->
      <User>username</User>
      <Password>password</Password>
      <IsEncryptedPassword>false</IsEncryptedPassword>
    </CredentialsConfiguration>
    <EndpointConfiguration>
      <!-- Because this endpoint should be accessed remotely, the IP Address should be used and
never could be LOCALHOST or 127.0.0.1  -->
      <Host>Host.IP.Adress</Host>
      <!-- A free port should be specified  -->
      <Port>Port</Port>
      <Route>api/{controller}/{action}/{id}</Route>
      <!-- Set this value to true if a secure connection would be used  -->
      <UseSSL>false</UseSSL>
    </EndpointConfiguration>
  </WebApiService>

</GConnectBastillePluginConfig>
```

```
### Configuration elements descriptions

- **BastilleServerConfig:**
  - Interval: The interval value is the time to wait for the next check of time elapsed from the last http request or heartbeat.  Interval is expressed in seconds
- **WebApiService:** It will contain the endpoint configuration where Bastille Server will stream the events detected.

  - **CredentialsConfiguration:** If the parameters of this section are set, them the endpoint will demand a authentication process to every request made to him.
    - Username: Username needed for endpoint authentication
    - Password: Password needed for endpoint authentication

  - **EndpointConfiguration**
    - Host: G-Connect host IP
    - Port: Port where will be published the endpoint.
    - Route: Configuration route for the endpoint
    - UserSSL: Define if SSL will be used to access the endpoint.

### Actually the GConnectBastillePlugin endpoint routes looks like this:
- http://{ip}:{port}/api/heartbeat
- http://{ip}:{port}/api/sendalarm

## Notes

It is necessary to execute G-Connect as administrator rights to allow the creating of web api endpoints

## GConnectBastillePlugin workflow

1. The process begins when Bastille Enterprise Server streams the events (heartbeat, events, etc.) to the endpoints published by the plugin.
2. If the event is a heartbeat, a log message is sent to GCONNECT console.
3. If the event is a device offline, a log message is sent to GCONNECT console and a PPDeviceOffline action is sent to GCore
4. If the event is a zone detection, a log message is sent to GCONNECT console and a PPZoneAlert action is sent to GCore
5. If the elapsed time between the last http request and the check time is bigger than the Interval defined by configuration, a log message is sent to GCONNECT console and a PPDeviceOffline action is sent to Gcore
```

8.37 Southwest Microwave RPMII plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

Overview

GConnectSWMicrowaveRPMPlugin is a plugin to integrate the Remote Polling Module II (RPM) Southwest Microwave into G-Connect.

The Remote Polling Module II (RPM) developed by Southwest Microwave Inc is an application layer
protocol that provides a method for customer software to query status/send commands to remote devices
attached to an INTREPID Remote Polling Module II (RPM II) via TCP/IP socket layer.

Supported devices:

- RPMII
- PMII
- ROM II-16
- ROM II-8
- AIM II
- MTP II
- MODEL 330

Configuration files

GConnectSWMicrowaveRPMPluginConfig.xml should be inside Configuration folder, inside the G-CONNECT installation folder.

GConnectSWMicrowaveRPMPluginConfig.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<GConnectSWMicrowaveRPMPluginConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="G-ConnectSWMicrowaveRPMPluginConfig.xsd">

  <ConnectionConfiguration>
    <!-- Pull Interval in seconds -->
    <Interval>5</Interval>
    <!-- INTREPID RPM II module IpAddress v4  -->
    <Host>localhost</Host>
    <!-- Connection port  -->
    <Port>50001</Port>
  </ConnectionConfiguration>

  <CredentialsConfiguration>
    <!-- Username for credentials -->
    <Username>sysadmin</Username>
    <!-- Password for credentials -->
    <Password>masterkey</Password>
  </CredentialsConfiguration>

  <ReconnectionConfig>
    <!-- Reconnection Interval in seconds-->
    <Interval>20</Interval>
    <!-- Maximum number of reconnection attempts  -->
    <!-- After the maximum of reconnection attempts gets reached, the plugin will not try to reconnect automatically any more -->
    <Attempts>100</Attempts>
  </ReconnectionConfig>
```

```
  <DateTimeConfiguration>
    <!-- Allows to set the Date/Time to the RPM -->
    <!-- Default value is FALSE -->
    <SetDateTime>false</SetDateTime>
  </DateTimeConfiguration>

  <TestingMode>
    <!-- TestingMode is for the plugin to simulate a connection to RPM extracting information of devices and alarms     -->
    <!-- and sending the proper actions th GCore server -->
    <!-- When it is TRUE: The plugin will avoid to connect to the RPM module and the login process too. It will be focused on the interpretation of some autogenerated test data of devices and alarms, and send to GCore the proper alarms and notifications -->
    <!-- When it is FALSE: The plugin will try to connect to the RPM module address defined in the configuration file ConnectionConfiguration section -->
    <!-- Default value is FALSE -->
    <Activated>false</Activated>
  </TestingMode>

</GConnectSWMicrowaveRPMPluginConfig>
```

Configuration elements descriptions

- **ConnectionConfiguration:**
  - Interval: The interval value is the time to wait for the next check of time elapsed from the last request or communication. Interval is expressed in seconds. Default value is 5.
  - Host: The IP Address of the RPM. Default value is 192.168.1.4.
  - Port: Port defined in the RPM for allowing incoming connections. Default value is 50001.
- **CredentialsConfiguration:**
    - Username: Username needed for the login process
    - Password: Password needed for the login process
- **ReconnectionConfig:**
  - Interval: The interval value is the time to wait for the next check of time elapsed from the last request or communication. Interval is expressed in seconds. Default value is 20.
  - Attempts: Maximum number of reconnection attempts. Default value is 100.
- **DateTimeConfiguration:**
  - SetDateTime: Allows to set the Date/Time to the RPM. Default value is FALSE.
- **TestingMode:**
  - TestingMode is for the plugin to simulate a connection to RPM extracting information of devices and alarms and sending the proper actions th GCore server. When it is TRUE: The plugin will avoid to connect to the RPM module and the login process too. It will be focused on the interpretation of some autogenerated test data of devices and alarms, and send to GCore the proper alarms and notifications. When it is FALSE: The plugin will try to connect to the RPM module address defined in the configuration file ConnectionConfiguration section. Default value is FALSE.

GConnectSWMicrowaveRPMPlugin workflow

1. The plugin start the communication sending the GetLoginInformation request.

2. If the GetLoginInformation response status value is 0, the plugin start the next step. If the response status value is bigger than 0, it indicates and error and the Reconnection process will start.
3. The plugin will send the SendLoginCredentials command.
4. If the SendLoginCredentials response status value is 0, the plugin start the next step. If the response status value is bigger than 0, it indicates and error and the Reconnection process will start.
5. Optional: If defined in the configuration, the plugin will send the SetDateTime command.
6. If the SetDateTime response status value is 0, the plugin start the next step. If the response status value is bigger than 0, it indicates and error and the Reconnection process will start.
7. The plugin will send the GetRPMIIAttachedDevices command.
8. If the GetRPMIIAttachedDevices response status value is 0, the plugin start the next step. If the response status value is bigger than 0, it indicates and error and the Reconnection process will start.
9. The plugin will send the GetAlarmRecordOfAllDevices command.
10. If the GetAlarmRecordOfAllDevices response status value is 0, the plugin start the next step. If the response status value is bigger than 0, it indicates and error and the Reconnection process will start.
11. The plugin will translate all the device alarms into GCore actions and send them to the GCore server.
12. All the process will be repeated from step 7.

Actions send to GCore

PPDeviceOffline (InterfaceID: " IP Adress of the plugin")
PPDeviceAlarm (InterfaceID: " IP Adress of the plugin", DeviceAddress: 33, Sensor: 6, State: 1)
PPZoneAlarm (InterfaceID: "192.168.1.44", DeviceAddress: 2, Cable: 0, Subcell: 444, State: 1)

Actions send to RMPII

The plugin should utilize MSGID 0x17 to perform Set/Clear Relays operation upon receiving the action with the parameters PPSetDeviceOutput (InterfaceID: "192.168.1.24", DeviceAddress: 4, Output: 1, State: 0)

8.38 Logflows

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

#Overview

In the Cingleot Warehouse project G-Connect will be used for connecting the different systems to the VMS. The following systems will be utilized:

1. Logflows
   A yard management system processing logistic company subscriptions for transport orders. It manages white, black and booking-lists and dock-assignments to vehicles as well.
2. Barrier system (via G-Matrix middleware)
   A system containing barriers and vehicle detection cameras at the main entrance. A login terminal allows short-term login to the yard management system.

3. Video analytics (via G-Matrix middleware)
   At the ramp and docks on each floor video analytics detect vehicles and obstacles. The analytic data is required for navigation and dock assignments.
4. Navigation displays (via G-Matrix middleware)
   The navigation displays show the drivers where to conduct their vehicles.

The Logflows plugin do handle the communication between these endpoints.

Prerequisites
- Microsoft NetFramework 4.8 runtime. https://go.microsoft.com/fwlink/?LinkId=2085155
- Microsoft Sql Server 2019 LocalDB or superior. https://go.microsoft.com/fwlink/?LinkID=866658

Configuration files
- G-ConnectConfig.xml
- GConnectLogflowPluginConfig.xml
- Logflow_ClientConfig.xml
- GMatrix_ClientConfig.xml
- LogflowLocal_ClientConfig.xml
 This files should be inside the Configuration folder, in the G-CONNECT installation folder.

GConnectLogflowPluginConfig.xml
```xml
<?xml version="1.0" encoding="UTF-8"?>
<GConnectLogflowPluginConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="G-ConnectLogflowPluginConfig.xsd">

  <!-- Additional configurations needed for http clients connections -->
  <LogflowConfigPath>Configuration\Logflow_ClientConfig.xml</LogflowConfigPath>
  <GMatrixConfigPath>Configuration\GMatrix_ClientConfig.xml</GMatrixConfigPath>

<LogflowLocalConfigPath>Configuration\LogflowLocal_ClientConfig.xml</LogflowLocalConfigPath
>

  <!-- Endpoints configuration -->
  <!-- If Username is provided then the authorization header should be sent -->
  <WebApiService>
    <CredentialsConfiguration>
      <!-- Username and Password used for endpoint authentication/authorization process  -->
      <User>username</User>
      <Password>passwork</Password>
      <IsEncryptedPassword>false</IsEncryptedPassword>
    </CredentialsConfiguration>
    <EndpointConfiguration>
      <!-- Because this endpoint should be accessed remotely, the IP Address should be used and
never could be LOCALHOST or 127.0.0.1  -->
      <Host>IP Address</Host>
      <!-- A free port should be specified  -->
      <Port>Port</Port>
      <Route>api/{controller}/{action}/{id}</Route>
      <!-- Set this value to true if a secure connection would be used  -->
      <UseSSL>false</UseSSL>
    </EndpointConfiguration>
```

```
  </WebApiService>
</GConnectLogflowPluginConfig>
```

Configuration elements descriptions
- LogflowConfigPath: The path to the Http Client configuration for Logflow API connection. Actually to a Postman Mock Server.
- GMatrixConfigPath: The path to the Http Client configuration for GMatrix API connection. Actually to a Postman Mock Server.
- LogflowLocalConfigPath: The path to the Http Client configuration for Logflow API connection published locally. Due to a miss function of Postman, was needed to Mock some APIs locally like **DockAssign**.
- **WebApiService:** It will contain the endpoint configuration published by the plugin.

  - **CredentialsConfiguration:** If the parameters of this section are set, them the endpoint will demand a authentication process to every request made to him.
    - Username: Username needed for endpoint authentication
    - Password: Password needed for endpoint authentication
  - **EndpointConfiguration**
    - Host: G-Connect host IP Address
    - Port: Port where will be published the endpoint.
    - Route: Configuration route for the endpoint
    - UserSSL: Define if SSL will be used to access the endpoint.

Logflow_ClientConfig.xml
```xml
<?xml version="1.0" encoding="UTF-8"?>
<REST_API_ClientConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="RESTAPIClient.xsd">
  <!-- Name or IP address of the server -->
  <ServerIP>4b8587af-c63e-41de-b387-27976503ccc7.mock.pstmn.io</ServerIP>
  <!-- Port of the server. In case the default port 80 shall be used, this element maybe deleted -->
  <ServerPort>80</ServerPort>
  <!-- For basic authentication a username must be added here. If no authentication is needed, this
element maybe left blank or deleted -->
  <ServerUser></ServerUser>
  <!-- If needed the password must be encrypted. Use the AesPasswordEncrypter.exe tool. If not
needed, this element maybe left blank or deleted -->
  <ServerPassword></ServerPassword>
  <!-- Set to true if SSL encrypted transport shall be used. Default value is false. If no SSL
encrypted transport is needed, the element maybe deleted -->
  <UseSSL>true</UseSSL>

  <!-- Timeout definition for every Http requests expressed in seconds. Default value is 5 -->
  <Timeout>10</Timeout>

  <!-- List of destination URLs. The ID references the command set in the
XMLConnect...PluginConfig.xml -->
  <URLList>
    <!-- Deprecated? -->
```

```
    <URL ID="dockrequest" URLString="dockrequest" Command="POST"
ContentType="application/json"/>
    <!-- -->

    <URL ID="dockassign" URLString="dockassign" Command="POST"
ContentType="application/json"/>
    <URL ID="whitelist" URLString="whitelist" Command="GET" ContentType="application/json"/>
    <URL ID="blacklist" URLString="blacklist" Command="GET" ContentType="application/json"/>
    <URL ID="bookinglist" URLString="bookinglist" Command="GET"
ContentType="application/json"/>
  </URLList>
  <!-- List of accepted content types. Defines the data formats to be accepted by the client -->
  <AcceptedContentTypes>
    <AcceptedContentType>application/json</AcceptedContentType>
  </AcceptedContentTypes>
</REST_API_ClientConfig>
```

Configuration elements descriptions

- ServerIP: Logflow Server IP Address. The Server Name can be used too.
- ServerPort: Port of the server used for publishing the APIs. In case the default port 80 shall be used, this element maybe deleted.
- ServerUser: For basic authentication a username must be added here. If no authentication is needed, this element maybe left blank or deleted.
- ServerPassword: If needed the password must be encrypted. Use the AesPasswordEncrypter.exe tool. If not needed, this element maybe left blank or deleted.
- UseSSL: Set to true if SSL encrypted transport shall be used. Default value is false. If no SSL encrypted transport is needed, the element maybe deleted.
- Timeout: Timeout definition for every Http requests expressed in seconds. Default value is 5.
- URLList: Will contain the APIs URLs needed to consume from the defined server.

GMatrix_ClientConfig.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<REST_API_ClientConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="RESTAPIClient.xsd">
  <!-- Name or IP address of the server -->
  <ServerIP>013e4746-7153-4ea0-b2fb-0fdcb54bff5a.mock.pstmn.io</ServerIP>
  <!-- Port of the server. In case the default port 80 shall be used, this element maybe deleted -->
  <ServerPort>80</ServerPort>
  <!-- For basic authentication a username must be added here. If no authentication is needed, this
element maybe left blank or deleted -->
  <ServerUser></ServerUser>
  <!-- If needed the password must be encrypted. Use the AesPasswordEncrypter.exe tool. If not
needed, this element maybe left blank or deleted -->
  <ServerPassword></ServerPassword>
  <!-- Set to true if SSL encrypted transport shall be used. Default value is false. If no SSL
encrypted transport is needed, the element maybe deleted -->
  <UseSSL>true</UseSSL>
```

```
  <!-- Timeout definition for every Http requests expressed in seconds. Default value is 5 -->
  <Timeout>10</Timeout>

  <!-- List of destination URLs. The ID references the command set in the
XMLConnect...PluginConfig.xml -->
  <URLList>
    <URL ID="opengate" URLString="opengate" Command="POST"
ContentType="application/json"/>
    <URL ID="toparking" URLString="toparking" Command="POST"
ContentType="application/json"/>
  </URLList>
  <!-- List of accepted content types. Defines the data formats to be accepted by the client -->
  <AcceptedContentTypes>
    <AcceptedContentType>application/json</AcceptedContentType>
  </AcceptedContentTypes>
</REST_API_ClientConfig>
```

Configuration elements descriptions
- ServerIP: Server IP Address. The Server Name can be used too.
- ServerPort: Port of the server used for publishing the APIs. In case the default port 80 shall be used, this element maybe deleted.
- ServerUser: For basic authentication a username must be added here. If no authentication is needed, this element maybe left blank or deleted.
- ServerPassword: If needed the password must be encrypted. Use the AesPasswordEncrypter.exe tool. If not needed, this element maybe left blank or deleted.
- UseSSL: Set to true if SSL encrypted transport shall be used. Default value is false. If no SSL encrypted transport is needed, the element maybe deleted.
- Timeout: Timeout definition for every Http requests expressed in seconds. Default value is 5.
- URLList: Will contain the APIs URLs needed to consume from the defined server

LogflowLocal_ClientConfig.xml
```xml
<REST_API_ClientConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="RESTAPIClient.xsd">
  <!-- Name or IP address of the server -->
  <ServerIP>Local IP Address</ServerIP>
  <!-- Port of the server. In case the default port 80 shall be used, this element maybe deleted -->
  <ServerPort>Port</ServerPort>
  <!-- For basic authentication a username must be added here. If no authentication is needed, this
element maybe left blank or deleted -->
  <ServerUser></ServerUser>
  <!-- If needed the password must be encrypted. Use the AesPasswordEncrypter.exe tool. If not
needed, this element maybe left blank or deleted -->
  <ServerPassword></ServerPassword>
  <!-- Set to true if SSL encrypted transport shall be used. Default value is false. If no SSL
encrypted transport is needed, the element maybe deleted -->
  <UseSSL>false</UseSSL>

  <!-- Timeout definition for every Http requests expressed in seconds. Default value is 5 -->
  <Timeout>20</Timeout>
```

```
  <!-- List of destination URLs. The ID references the command set in the
XMLConnect...PluginConfig.xml -->
  <URLList>
    <URL ID="dockassign" URLString="api/dockassign" Command="POST"
ContentType="application/json"/>
  </URLList>
  <!-- List of accepted content types. Defines the data formats to be accepted by the client -->
  <AcceptedContentTypes>
    <AcceptedContentType>application/json</AcceptedContentType>
  </AcceptedContentTypes>
</REST_API_ClientConfig>
```

Configuration elements descriptions
- ServerIP: IP Address of the host where the plugin is deployed.
- ServerPort: Port of the server used for publishing the APIs. In case the default port 80 shall be used, this element maybe deleted.
- ServerUser: For basic authentication a username must be added here. If no authentication is needed, this element maybe left blank or deleted.
- ServerPassword: If needed the password must be encrypted. Use the AesPasswordEncrypter.exe tool. If not needed, this element maybe left blank or deleted.
- UseSSL: Set to true if SSL encrypted transport shall be used. Default value is false. If no SSL encrypted transport is needed, the element maybe deleted.
- Timeout: Timeout definition for every Http requests expressed in seconds. Default value is 5.
- URLList: Will contain the APIs URLs needed to consume from the defined server

Actually the GConnectLogflowPlugin endpoint routes looks like this:
- http://{ip}:{port}/api/dockassign

8.39 CCure9000

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

#Overview

The CCure9000 plugin connects to the Victor service platform of CCure 9000 Version 2.9 and 3.0. The CCure victor webservice has to be installed and a licence is needed on the CCure system.

The CCure9000 plugin is able to receive door events, I/O information and card reader information. It also receive events from the CCure system, tamper etc.
With the CCure9000 plugin, GCore can send door unlock and door lock commands to the CCure system. GCore can also set specific outputs on the CCure system.

The specifc doors and output do have readable names like “Door1Floor1” or “Output30”.
These objects (doors/cardreader/output/inputs) needs to be named/renamed in CCure hardware application.

The configuration files are:

- G-ConnectConfig.xml
- GConnectGCorePluginConfig.xml
- RESTAPI_ClientConfig.xml

The configuration file which must be adjusted to the system is RESTAPI_ClientConfig.xml.

Configuration elements descriptions

- ServerIP: Name or IP address of the victor webserver
- ServerPort: Publication Port from the victor webserver. Standard 80
- ServerUser: Username needed for the login process into CCure/Victor webservice
  It depends on how the CCure system was installed.
  It could be the windows credentials
  <ServerUser>DESKTOP-77GGHGQ\CCureUser1</ServerUser>
  “DESKTOP-77GGHGQ” is here the server name of CCure installation
  “CCureUser1” is here the username
- Password: Password needed for the login process into CCure/Victor webservice
  Password used along with the username
- ClientID: This is the GUID provided by CCure and must not be changed
- ClientVersion: The correct version of your CCure Client has to be set. (2.9 or 3.0).
- SignalRUrl: SignalR Url in the HTTP Server. Do not change
- SignalRHubName: SignalR Hub Name. Do not change
- ReconnectionInterval: Elapsed time for the next connection try. Default value is 10.
- ReconnectionAttemps: Maximum number of reconnection attempts. Default value is 5.
  After that number of reconnect tries the plugin will be disabled

GConnectCCurePlugin workflow

1. The plugin start the communication requesting to Login and obtaining the SessionId/Token.
2. If the Login response is success true, the plugin start the next step. If the response is false or the communication with the server fails, it indicates and error as a Gcore action SMSSystemInformation and the Reconnection process will start.
3. The plugin will start the KeepAlive service, since if the server detects an inactivity time greater than 90 s it implies the connection closure.
4. The plugin will subscribe to the SignalR hub to listening for notifications and events

Actions created in Gcore system

Door events are sent to GCore as follows:

FORCE event
SMSDoorEvent(2, 0, "Devicename", ObjectId, "Server Name/Ip")

OPEN event
SMSDoorEvent(4, 0, "Devicename", ObjectId, "Server Name/Ip")

CLOSE event
SMSDoorEvent(5, 0, "Devicename", ObjectId, "Server Name/Ip")

HOLD event
SMSDoorEvent(3, 0, "Devicename", ObjectId, "Server Name/Ip")

Input/Outputs events are sent to GCore as follows
For Input Devices:
- ARM/ACTIVATE event
SMSIONotification(1, 0, "Devicename", ObjectId, "Server Name/Ip")

- DISARM event
SMSIONotification(0, 0, "Devicename", ObjectId, "Server Name/Ip")

For Output Devices:
- ACTIVATE event
SMSIONotification(1, 0, "Devicename", ObjectId, "Server Name/Ip")

- DEACTIVATE event
SMSIONotification(0, 0, "Devicename", ObjectId, "Server Name/Ip")

Admit/Denied cards are sent to GCore as follows:
ADMIT event
SMSCardReaderNotification(0, 0, CardNumber, "Devicename", ObjectId, "Server Name/Ip")

REJECT event
SMSCardReaderNotification(1, 0, CardNumber, "Devicename", ObjectId, "Server Name/Ip")

All other events are sent to GCore as follows:
SMSIONotification

Example:
SMSIONotification (Status: 1, RequestType: 0, DeviceName: "Tamper-TestApc1", DeviceID: 5451, ServerHost: "sdkgcore.geutebrueck.com", ClientAccount: "GEUTEBRUECK\Kruegel0211", ClientHost: "AN16-005104.geutebrueck.de", ClientType: 5, GeneralProcessingTimeStamp: "2023/08/02 09:33:17,652 GMT+02:00", User1: "sysadmin")

Actions from GCore to GConnectCCurePlugin for executing some actions in CCure:

LOCK Door request:
SMSDoorEvent(5, 1, "Devicename", "Server Name/Ip")
Example:
SMSDoorEvent(5, 1, "Door2Floor1", " DESKTOP-77GGHGQ ")

UNLOCK Door request:
SMSDoorEvent(4, 1, "Devicename", "Server Name/Ip")
Example:
SMSDoorEvent(4, 1, "Door2Floor1", " DESKTOP-77GGHGQ ")

ARM/ACTIVATE OUTPUT request:
SMSIONotification(Status: 0, RequestType: 1, "Devicename", "Server Name/Ip is optional")
Example:

SMSIONotification (Status: 0, RequestType: 1, DeviceName: "istar output1", ServerHost: "WIN10-REMOTE-UP")

DISARM/DEACTIVATE OUTPUT request:
SMSIONotification(Status: 1, RequestType: 1, "Devicename", "Server Name/Ip is optional")

Needed GCore license:
Option G-Connect/CCure9000 8.32010

8.40 Southwest Microwave POE plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

Overview

GConnectSWMicrowavePOEPlugin is a plugin to integrate the INTREPID Polling Protocol II (IPP II) from Southwest Microwave Inc (SMI) into G-Connect.
INTREPID Polling Protocol II (IPP II) developed by Southwest Microwave Inc is an application layer protocol that provides Master/Slave communications between control equipment and SMI INTREPID Polling Protocol II sensors & peripherals.

Supported devices:
Model 316T-POE-S Transmitter (316-POE-S)
Model 316R-POE-S Receiver (316-POE-S)
Model 336T-POE-S Transmitter (336-POE-S)
Model 336R-POE-S Receiver (336-POE-S)
Model 334T-POE-S Transmitter (334-POE-S)
Model 334R-POE-S Receiver (334-POE-S)
Processor Module-POE-S (PM-POE-S)
Model 390 Transceiver (390)
Model 395 Transceiver (395)
Relay Output Module-POE-S (ROM-POE-S)
Alarm Input Module-POE-8S (AIM-POE-8S)

Configuration files

GConnectSWMicrowavePOEPluginConfig.xml should be inside Configuration folder, inside the G-CONNECT installation folder.

GConnectSWMicrowavePOEPluginConfig.xml
```xml
<?xml version="1.0" encoding="UTF-8"?>
<GConnectSWMicrowavePOEPluginConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="G-ConnectSWMicrowavePOEPluginConfig.xsd">
```

```
  <ConnectionConfiguration>
    <!-- Polling Interval in milliseconds-->
    <PollInterval>1000</PollInterval>
    <!-- Polling Agent Service Port minVal = 49152  maxVal = 65535 -->
    <PollingAgentPort>49152</PollingAgentPort>
    <!-- Connection Mode Active/Passive -->
    <ConnectionMode Type="Active" />
    <!-- Controller Slot to connect Primary/Secondary/Any -->
    <ControllerSlot Type="Any"/>
    <!-- Conflict Action Override/Fail -->
    <ConflictAction Type="Overwrite" />
    <!-- Secure Connection true/false -->
    <SecurityConnection>false</SecurityConnection>
    <!-- Polling Agent Client Authentication Required true/false -->
    <PollingAgentClientAuthentication>false</PollingAgentClientAuthentication>
  </ConnectionConfiguration>

  <!-- List of INTREPID devices --> Up to 32 devices
  <DeviceList>
    <DeviceConfig IpAddress="Device.IP.Address.1" Port="DevicePort1"/>
    <DeviceConfig IpAddress="Device.IP.Address.2" Port="DevicePort2"/>
    <DeviceConfig IpAddress="Device.IP.Address.3" Port="DevicePort3"/>
    <DeviceConfig IpAddress="Device.IP.Address.4" Port="DevicePort4"/>
    <DeviceConfig IpAddress="Device.IP.Address.5" Port="DevicePort5"/>
    <DeviceConfig IpAddress="Device.IP.Address.6" Port="DevicePort6"/>
    <DeviceConfig IpAddress="Device.IP.Address.7" Port="DevicePort7"/>
  </DeviceList>

  <CredentialsConfiguration>
    <!-- Password for credentials -->
    <Password>Password#0</Password>
  </CredentialsConfiguration>

  <ReconnectionConfig>
    <!-- Reconnection Interval in seconds-->
    <Interval>5</Interval>
  </ReconnectionConfig>

</GConnectSWMicrowavePOEPluginConfig>
```

Configuration elements descriptions

- **ConnectionConfiguration:**
  - PollingInterval: The polling interval is the waiting time between one poll and another. PollingInterval is expressed in milliseconds. Default value is 1000.
  - PollingAgentPort: Port defined for allowing incoming connections. Default value is 50001.
  - ConnectionMode: Defines the kind of connection to use. Default value is Active.
  - ControllerSlot: Defines the preferred controller slot to connect. Default value is Any.
  - ConflictAction: Defines the value for conflict action. Default value is Overwrite.
  - SecurityConnection: Enable the SSL-encrypted communication. Default value is false.
  - PollingAgentClientAuthentication: Enable the polling agent client authentication. Default value is false.

- **CredentialsConfiguration:**
  - Password: Password needed for the login process
- **ReconnectionConfig:**
  - Interval: The interval value is the time to wait for the next check of time elapsed from the last request or communication. Interval is expressed in seconds. Default value is 5.

GConnectSWMicrowavePOEPlugin workflow

1. The plugin start the communication with every device defined by configuration.
2. If the connection is successful, the plugin start the next step. If the connection fails, it indicates and error and the Reconnection process will start.
3. The plugin will start the polling process.
4. If the polling process is successful, the plugin start the next step. If the polling process fails, it indicates and error and the Reconnection process will start.
5. The plugin will translate all the device alarms into GCore actions and send them to the GCore server.
6. All the process will be repeated from step 3.

Incoming actions from GCore
If is needed to activate/deactivate a relay in a POE ROM Device from GCore, then the PPSetDeviceOutput can be used following this syntax:

```cmd
PPSetDeviceOutput (InterfaceID: "192.168.35.14:50091", DeviceAddress: 14, Output: 0, State: 0)
```
- InterfaceID: It is the IP Address and Port to connect to the POE ROM Device
- DeviceAddress: It is the last segment of the IP Address assigned to the device
- Output: It is the number of the Relay to activate/deactivate starting from 0
- State: 0 means CLEAR, 1 means SET

Actions sent to GCore
When the Discovering or Polling process fails
```cmd
PPDeviceOffline (InterfaceID: "192.168.35.14:50091")
```
- InterfaceID: It is the IP Address and Port to connect to the POE Device

For alarm sensor type: CableDisturbance (PM Poll type)
```cmd
PPZoneAlarm (InterfaceID: "192.168.35.14:50091", DeviceAddress: 14, Cable: 0, Subcell: 3, State: 1)
```
- DeviceAddress: Number indicating the last segment of the device ip address
- Cable: Indicates the cable containing the subcells reporting the alarm
- Subcell: Indicate the subcell reporting the alarm
- State: 0 Normal. 1 Alarm.

For alarm sensor type: SupervisedInput or ExternalInput
```cmd
PPDeviceInput (InterfaceID: "192.168.35.14:50091", DeviceAddress: 14, Input: 2, State: 1)
```
- Input: Indicates the Input reporting the alarm

For alarm sensor type: Alarm
```cmd
PPDeviceAlarm (InterfaceID: "192.168.35.14:50091", DeviceAddress: 14, Sensor: 4, State: 1)
```
- Sensor: 4 means "Sensor"

For alarm sensor type: CableFault
```cmd
PPDeviceAlarm (InterfaceID: "192.168.35.14:50091", DeviceAddress: 14, Sensor: 0, State: 1)
```
- Sensor: Value 0 means "Cable A Fault". Value 1 means "Cable B Fault"

For alarm sensor type: Communication
```cmd
PPDeviceAlarm (InterfaceID: "192.168.35.14:50091", DeviceAddress: 14, Sensor: 6, State: 1)
```

```
- Sensor: Value 6 means "Aux"

### For alarm sensor type: CertificateExpired, MicrowaveDisturbance, ServiceTool, Tamper
```cmd
PPDeviceAlarm (InterfaceID: "192.168.35.14:50091", DeviceAddress: 14, Sensor: 2, State: 1)
```
- Sensor: Value 2 means "Tamper"

For alarm sensor type: PathAlignment
```cmd
PPDeviceAlarm (InterfaceID: "192.168.35.14:50091", DeviceAddress: 14, Sensor: 5, State: 1)
```
- Sensor: Value 5 means "Path"
```

8.41 DMP Alarm Panel plug-in

Info: NOT FailOver capable.
See chapter 12. ## Failover capable plugins.

Overview

DMP is a plugin to integrate the Alarm Panel from DMP into G-Connect.
https://www.dmp.com/products/control-panels

These alarm panels are able to connect to G-Connect open port via TCP/IP.
They transfer all messages from the panel to G-Connect where they are sperated into different G-Core actions.

The port where the plugin can be reached by the DMP alarm panel is defined in the config file “GConnectTelnetPluginConfig.xml”. <Port>2001</Port>

Mapped actions:
Alarms related to “Zones” will be transferred to “PPDeviceAlarm”.

Messages from DMP “Door Access 1-12345 Zj\045\t "DA\v 001\u 00001"WILLIAM SMITH \”
will be transferred to “SMSDoorEvent”

All other messages from DMP alarm panel will be transferred to “SMSSystemInformation” or ”CustomActionEx”.

G-Core actions will show the original message from the DMP panel.
e.g.“12345 Za\037\t "FI\z 001"OFFICE SMOKE DET\ee"RI\"”

12345 is the account number which is configured in the DMP panel.

“Za” (Zone Alarm) is the classification for the message and can be found in the document “LT-0872.pdf” which is part of this installation.

Examples:

##STXD8DF 12345 Zs\014\t 087\##CR
SMSSystemInformation (Information: "12345 Zs\014\t 087\", ServerHost: "Port:2001", ActionID: "14", ClientAccount: "GEUTEBRUECK\Kruegel0211")

Door Access
##STXD8DF 12345 Zj\045\t "DA\v 001\u 00001"WILLIAM SMITH\##CR
SMSDoorEvent (DoorState: 4, RequestType: 0, DeviceName: "12345 Zj\045\t "DA\v 001\u 00001"WILLIAM SMITH\", DeviceID: 12345, ServerHost: "Port:2001", ClientAccount: "GEUTEBRUECK\Kruegel0211")

Zone Alarm
##STXD8DF 12345 Za\062\t "BU\z 0232"FRONT DOOR\a 03"OFFICE\u 0568"JOHN SMITH\##CR
PPDeviceAlarm (InterfaceID: "12345 Za\062\t "BU\z 0232"FRONT DOOR\a 03"OFFICE\u 0568"JOHN SMITH\", DeviceAddress: 12345, Sensor: 4, State: 1, ClientAccount: "GEUTEBRUECK\Kruegel0211")

WARNING: Memory Usage
##STXD8DF 12345 Zs\014\t 151\e7"DISPLAY\##CR
SMSSystemInformation (Information: "12345 Zs\014\t 151\e7"DISPLAY\", ServerHost: "Port:2001", ActionID: "14", ClientAccount: "GEUTEBRUECK\Kruegel0211")

WARNING: Line Card
##STXD8DF 12345 Zs\019\t 154\em002\##CR
SMSSystemInformation (Information: "12345 Zs\019\t 154\em002\", ServerHost: "Port:2001", ActionID: "19", ClientAccount: "GEUTEBRUECK\Kruegel0211")

Burglary Alarm
##STXD8DF 12345 Za\060\t "BU\z 001"EAST OFFICE DOOR\a 001"EAST WAREHOUSE \##CR
PPDeviceAlarm (InterfaceID: "12345 Za\060\t "BU\z 001"EAST OFFICE DOOR\a 001"EAST WAREHOUSE \", DeviceAddress: 12345, Sensor: 4, State: 1, ClientAccount: "GEUTEBRUECK\Kruegel0211")

Generic Alarm
##STXD8DF 12345 Zr\060\t "BL\z 001"EAST OFFICE DOOR\a 001"EAST WAREHOUSE \ee"RI\##CR
PPDeviceAlarm (InterfaceID: "12345 Zr\060\t "BL\z 001"EAST OFFICE DOOR\a 001"EAST WAREHOUSE \ee"RI\", DeviceAddress: 12345, Sensor: 4, State: 1, ClientAccount: "GEUTEBRUECK\Kruegel0211")

Fire Alarm
##STXD8DF 12345 Za\037\t "FI\z 001"OFFICE SMOKE DET\ee"RI\##CR
PPDeviceAlarm (InterfaceID: "12345 Za\037\t "FI\z 001"OFFICE SMOKE DET\ee"RI\", DeviceAddress: 12345, Sensor: 4, State: 1, ClientAccount: "GEUTEBRUECK\Kruegel0211")

Service Message - Dirty Smoke Detector
##STXD8DF 12345 Zw\043\t "FI\z 001"OFFICE SMOKE DET\e_"DT\ee"NO\##CR
PPDeviceAlarm (InterfaceID: "12345 Zw\043\t "FI\z 001"OFFICE SMOKE DET\e_"DT\ee"NO\", DeviceAddress: 12345, Sensor: 4, State: 1, ClientAccount: "GEUTEBRUECK\Kruegel0211")

Burglary Zone Bypass by User
##STXD8DF 12345 Zx\085\t "BU\z 001"EAST OFFICE DOOR\u 00001"WILLIAM SMITH \a 001"EAST WAREHOUSE \##CR
PPDeviceAlarm (InterfaceID: "12345 Zx\085\t "BU\z 001"EAST OFFICE DOOR\u 00001"WILLIAM SMITH \a 001"EAST WAREHOUSE \", DeviceAddress: 12345, Sensor: 4, State: 1, ClientAccount: "GEUTEBRUECK\Kruegel0211")

Burglary Zone Reset from Bypass by User
##STXD8DF 12345 Zy\085\t "BU\z 001"EAST OFFICE DOOR\u 00001"WILLIAM SMITH \a 001"EAST WAREHOUSE \##CR
PPDeviceAlarm (InterfaceID: "12345 Zy\085\t "BU\z 001"EAST OFFICE DOOR\u 00001"WILLIAM SMITH \a 001"EAST WAREHOUSE \", DeviceAddress: 12345, Sensor: 4, State: 1, ClientAccount: "GEUTEBRUECK\Kruegel0211")

Burglary Zone Automatic Swinger Bypass by System
##STXD8DF 12345 Zx\085\t "BU\z 001"EAST OFFICE DOOR\u 32765"SWINGER BYPASS \a 001"EAST WAREHOUSE \ee"NO\##CR
PPDeviceAlarm (InterfaceID: "12345 Zx\085\t "BU\z 001"EAST OFFICE DOOR\u 32765"SWINGER BYPASS \a 001"EAST WAREHOUSE \ee"NO\", DeviceAddress: 12345, Sensor: 4, State: 1, ClientAccount: "GEUTEBRUECK\Kruegel0211")

Zone Trip Count
##STXD8DF 12345 Zp\067\t "BU\z 001"EAST OFFICE DOOR\a 001"EAST WAREHOUSE \e 00123\##CR
CustomActionEx (Int64FieldA: 12345, StringFieldA: "DMP Panel", StringFieldB: "Message Unkown", StringFieldC: "12345 Zp\067\t "BU\z 001"EAST OFFICE DOOR\a 001"EAST WAREHOUSE \e 00123\", StringFieldD: "Zp", ClientAccount: "GEUTEBRUECK\Kruegel0211")

Wireless Keypad Tamper-Zone Trouble
##STXD8DF 12345 Zt\037\t "A1\z 202"WIRELESS KEYPAD \##CR
PPDeviceAlarm (InterfaceID: "12345 Zt\037\t "A1\z 202"WIRELESS KEYPAD \", DeviceAddress: 12345, Sensor: 4, State: 1, ClientAccount: "GEUTEBRUECK\Kruegel0211")

Carbon Monoxide Alarm
##STXD8DF 12345 Za\037\t "CO\z 041"GROUND FLOOR CO \##CR
PPDeviceAlarm (InterfaceID: "12345 Za\037\t "CO\z 041"GROUND FLOOR CO \", DeviceAddress: 12345, Sensor: 4, State: 1, ClientAccount: "GEUTEBRUECK\Kruegel0211")

Video Alarm message on XR550 v193+ with camera ID: 1, Recorder ID: 2 and event ID: 123
##STXD8DF 12345 Za\032\t "VA\z 001\a 002\ev123\##CR
PPDeviceAlarm (InterfaceID: "12345 Za\032\t "VA\z 001\a 002\ev123\", DeviceAddress: 12345, Sensor: 4, State: 1, ClientAccount: "GEUTEBRUECK\Kruegel0211")

8.42 MBSecure plug-in

Info: FailOver capable.
See chapter 12. ## Failover capable plugins.

Overview

GConnectMBSecurePlugin is a plugin to integrate MBSecure into G-Connect.
The MB-Secure PRO is the innovative alarm control panel that combines hardware, firmware, licensing and future security in one integrated platform.

Prerequisites

Ensure you have met the following requirements:

- **.NET 7.0 Runtime**: Your system must have the .NET 7.0 runtime installed to run this application. If you do not have it installed, you can download it from the official .NET download page or directly via the following link for Windows (x64):

  [Download .NET 7.0 Runtime (Windows x64)](https://download.visualstudio.microsoft.com/download/pr/e35dac95-2855-44f9-b6c9-dda018d922ba/fcc2416e232942d81435a659024bd4e5/dotnet-runtime-7.0.17-win-x64.exe)

Configuration files

- MBSecurePluginConfig.xml
- RESTWS_ClientConfig.xml
- G-ConnectConfig.xml
- GConnectGCorePluginConfig.xml

Note on Configuration Files:
To ensure the smooth operation of G-CONNECT, please check all configuration files are in the `Configuration` folder. This folder is located inside the G-CONNECT installation directory.

Configuration elements descriptions

- **MBSecurePluginConfig:**
  - APIClientConfigPath: Path to connection configuration file
  - EventSubscriptionList: Contains the subscriptions to the events
    - EventSub: Contains the subscription definition
      - ClassName: Indicates the class of the device, for example: VirtualDoor, VirtualInput, VirtualOutput
      - EventType: Name of the EventType defined in the documentation
      - EventSubType: Name of the EventSubType defined in the documentation
      - UrlId: Contains the URL id defined in the RESTWS_ClientConfig.xml, used for subscribing to the events.
      - Active: When it is set to true, means that is wanted to subscribe to the events. When it is set to false, it means that this subscription will be registered.
  - CommandList: Contains the list of commands to be executed in the MBSecure Device.
    - Command: Identifies the URL to be used for the command execution in the MBSecure Device. URLs are defined in the RESTWS_ClientConfig.xml.
  - QueryList: Contains the list of the queries to be executed on the MBSecure Device.
    - Query: Identifies the URL to be used for the query execution (Message attribute) in the MBSecure Device and the MessageId (Response attribute) to be used to send the response to GCore.

Configuration elements descriptions

- ** RESTWS_ClientConfig.xml:**
  - ServerIP: IP address of the MBSecure Device.
  - ServerPort: MBSecure Device port.
  - ServerUser: User name authorized for the authentication.
  - ServerPassword: Password used for the authentication.
  - Interval: Interval to wait for checking if a new response arrives to the websocket client.
  - ExpirationTime: Time to wait for sending the keepalive.
  - MaxLoginAttempts: Maximum number of login attempts to be allowed when the authentication process fails. After this number of attempts, the plugin will be disposed.
  - URLList: List of URLS that should be used for interacting with the server.
    - URL: Contains the ID and the URLString. The ID is used in the other configurations.

GConnectMBSecurePlugin workflow

1. The plugin checks for the existence of the OPIF file. If not found, the plugin will not be able to start. The OPIF contains the data points from the MBSecure system and can be exported by IQ PanelControl. Refere to MBSecure documentation how to export the OPIF.TXT file.
2. The plugin starts the websocket connection to the MBSecure Device.
3. If the connection is successful, the plugin sends a login request to the server.
4. If the connection is unsuccessful, the plugin stays trying to connect to the server.
5. If the login request is successful then the plugin will attempt to connect to the server. The maximum number of attempts is defined in the configuration.
6. If the login request is successful, the plugin will subscribe to all the events defined in the configuration (OPIF.TXT) and activated (Active true).
7. After the event subscription, the plugin will send the Version and Devices requests to the server, showing the responses in the logs and sent as CustomAction to the Core.
8. When an event is received, the plugin will send the event content to the logs and send the appropriate GCoreAction according to the configuration.

Incoming actions from GCore

Sending commands to a VirtualDoor
If is needed to latch/unlatch a VirtualDoor in the MBSecure Device from GCore, then the SMSDoorEvent can be used following this syntax:

```cmd
SMSDoorEvent (DoorState: 4, RequestType: 1, DeviceID: 18874369, ServerHost: "192.168.2.33")
```
- DoorState: 4 means `unlatch`, 5 means `latch`.
- RequestType: When the value is 1 (`set`), means that the plugin will send a command to the MBSecure server.
- DeviceID: It is the ID of the door (OPIF.TXT) to execute the command.
- ServerHost: It is the IP of the MBSecure server in the network.

Sending commands to a VirtualInput or to a VirtualOutput

If is needed to close/open a VirtualInput or to a VirtualOutput in the MBSecure Device from GCore, then the SMSIONotification can be used following this syntax:

```cmd
SMSIONotification (Status: 0, RequestType: 1, DeviceName: "input", DeviceID: 20971521, ServerHost: "192.168.2.33")
```

- Status: 0 means `high/open`, 1 means `low/close`.
- RequestType: When the value is 1 (`set`), means that the plugin will send a command to the MBSecure Device.
- DeviceName: `input` means the command is for a VirtualInput, `output` means the command is for a VirtualOutput.
- DeviceID: It is the ID of the VirtualInput or VirtualOutput to execute the command.
- ServerHost: It is the IP of the MBSecure server in the network.

Clearing alarms to a partition

If is needed to clear the alarm state of a partition in the MBSecure Device from GCore, then the SMSSystemInformation can be used following this syntax:

```cmd
SMSSystemInformation (Status: 5, Information: "device type", ServerHost: "192.168.2.33", ActionID: 123456)
```

- Status: 5 means `all normal`.
- Information: `clearPartitions`.
- ServerHost: IP address of the intended MBSecure device.
- ActionID: It is the ID of the partition to clear alarms.

Sending commands to system to modify the date/time of the device

If is needed to modify the date/time of the device, then the SetClock can be used following this syntax:

```cmd
SetClock (Date: "2024/03/16 18:25:00,000 GMT-04:00")
```

- Date: It is the date/time formatted according to the ISO 8601 standard.

Sending queries to a Partition, VirtualDoor, VirtualInput, VirtualOutput

If is needed to know the state of any of the mentioned device types in the MBSecure Device from GCore, then the SMSSystemInformation can be used following this syntax:

```cmd
SMSSystemInformation (Status: 0, Information: "device type", ServerHost: "MBSecure Host IP Address", ActionID: 123456)
```

- Status: 0 means `unspecified`.
- Information: Any one of these types: `partitions`, `APs`, `inputs`, `outputs`, `doors`.
- ServerHost: MBSecure Host IP address of the intended MBSecure device.
- ActionID: It is the ID of the device to query.

9 Licensing

As of version 2.0.0.0 licensing has changed.
Each connection to a G-Core server needs a single **G-Core/XMLConnect** licence. It doesn't matter, where theses licenses are hosted. If the first connected G-Core server provides enough licenses to connect all configured G-Core servers, the others do not need their own ones. But it's also possible to host a single XMLConnect licence on each G-Core server. Some plug-ins need additional licenses. These can also be provided by one connected G-Core server or even distributed over all servers.

As of version 4.0.0.0 **G-Connect** needs a new license of type **G-Connect Integration Service** (8.32001) instead of G-Core/XMLConnect.

The following list shows the plug-ins consuming additional licenses, licence types and count:

Plug-in	Licence type	Count
~~Telnet~~	~~G-Connect Integration Service (8.32001)~~	~~1 per connected client~~
Face Recognition	FaceRec	1 per channel
Briefcam	Briefcam	1
NightWatch	G-Connect Integration Service (8.32001)	1 per connected base station
OnGuard	G-Connect-OnGuard (8.32002)	1 per connected G-Core
Nedap (as of 4.0.0.66)	G-Connect-Nedap (8.32003)	1 per connected G-Core

10 Configuration

The service needs two configuration files. If the recommended installation path is kept, the file "G-Core_XML.exe.config" can be found in "C:\Program Files\Geutebrueck\GCoreXMLService\" and "GenericXMLConfig.xml" is stored in "C:\Program Files\Geutebrueck\GCoreXMLService\Configuration\". Additional configuration files belonging to the plug-ins are stored in the same path as "GenericXMLConfig.xml".

Changes as of version 4.0.0.0:
The name of the executable G-Core_XML.exe is changed to ***G-Connect.exe***.
The name of the configuration file G-Core_XML.exe.config is changed to ***G-Connect.exe.config***.
The default path of the G-Connect service is ***C:\Program Files\Geutebrueck\GConnectService***.

10.1 Start and stop the service

After installation by default the **G-Connect** service is started automatically. After changes in the configuration the service must be restarted. To achieve this, the Windows services panel must be opened. You can open it from the control center or by typing "services.msc" plus pressing the Enter-key in the Windows search field.

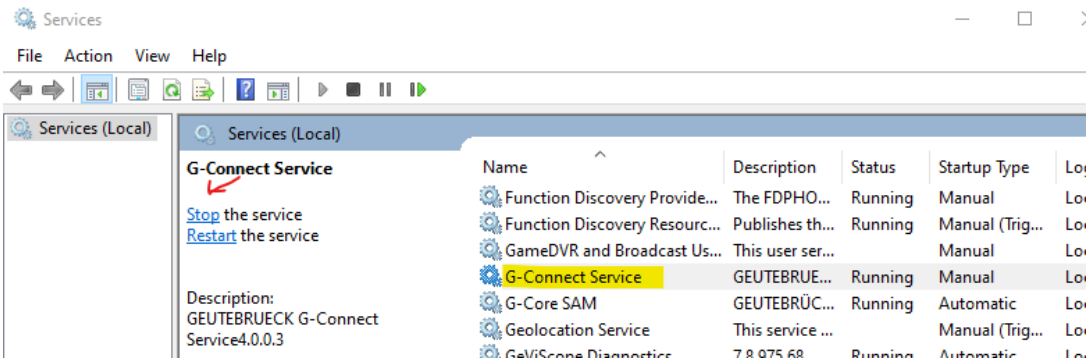

Figure 21: Windows services

To stop the service right click on the service entry an click “Stop” in the drop down menu or click “Stop” on the left side of the panel.
To start it again, click “Start”.

10.2 Application configuration

The application configuration file “G-Connect.exe.config” is located in the executable path “C:\Program Files\Geutebrueck\GConnectService\”.

```
<configuration>
    <configSections> ... </configSections>
    <startup> ... </startup>
  <runtime> ... </runtime>
    <userSettings>
        <G_Connect.Properties.Settings>
            <setting name="ApplicationServiceName" serializeAs="String">
                <value>G-Connect</value>
            </setting>
            <setting name="ApplicationServiceDescription" serializeAs="String">
                <value>GEUTEBRUECK G-Connect Service</value>
            </setting>
            <setting name="ApplicationServiceDisplayName" serializeAs="String">
                <value>G-Connect Service</value>
            </setting>
            <setting name="LogLevel" serializeAs="String">
                <value>3</value>
            </setting>
            <setting name="MaxLogTime" serializeAs="String">
                <value>10</value>
            </setting>
            <setting name="ConfigFile" serializeAs="String">
                <value>G-ConnectConfig.xml </value>
            </setting>
            <setting name="ConfigPath" serializeAs="String">
                <value>Configuration</value>
            </setting>
        </G_Connect.Properties.Settings>
    </userSettings>
</configuration>
```

Figure 22: G-Connect.exe.config

The following parameters are used:

Name	Range	default	Usage
ApplicationServiceName	String	G-Connect	Internal

ApplicationServiceDescription	String	GEUTEBRUECK G-Connect Service	Windows service panel
ApplicationServiceDisplayName	String	G-Connect Service	Windows service panel
LogLevel	0 .. 3	1	0 = no logging, 1 = only Info, Warning and Error 2 = plus Debug 3 = plus Trace
MaxLogTime	In days	10	Max number of days to be logged.
ConfigPath	String	Configuration	Path to the configuration files
ConfigFile	String	G-ConnectConfig.xml	Name of the basic configuration file

10.3 Main G-Connect configuration

⚠ As of version 3.0.0.1 the main configuration stops containing the G-Core connection details and Action and Events descriptions. Theses are now contained in the GCore Plug-in configuration. The main configuration is stored in the file “G-ConnectConfig.xml”. A setup tool to manipulate this file doesn’t yet exist. The file must be changed manually. Special xml editors like e.g. *Altova XMLSpy* may facilitate the process.
To define the xml structure a schema was created. Please refer to the linked document:
Linked%20Documents/G-ConnectConfig.pdf
Following some additional descriptions of elements and attributes are added.
In general “Member” attributes connect the different message fields to eachother. Names may be freely chosen but must be unique.

10.3.1 Element ThirdPartyConnection

The “ThirdPartyConnection” element contains name and path of the plug-in that connects to a 3rd party application.
The ID attribute contains a string for referencing the plug-in. This string can be designed freely but must be unique for each plug-in.

⚠ As of version 4.0.0.44 this element also contains the path of the configuration file. This allows loading a plug-in multiple times with different configurations. E.g.

10.3.2 Element XPathExpressionList

The element “XPathExpressionList” contains a list of XPath expression sets to evaluate the element and/or attribute values.

The “XPathExpressionSet” element provides a list of XPath expressions to evaluate element and/or attribute values. The expressions are referenced by a set ID to fetch multiple values of a single XML document.

The element “XPathExpression” describes the path inside the XML document to read or write an attribute or element value. The “Member” attribute connects the expression result e.g. to a message parameter. An XPath expression may contain a member reference embraced in curly braces.
Example: `<XPathExpression Expression="Vehicles/Vehicle[Plate=&quot;{Plate}&quot;]/Type" Member="Type"></XPathExpression>`

⚠ Quotes inside a string in an XLM attribute must be translated to ***"***

⚠ As of version 4.0.0.51 XPath 2.0 functions are available for querying XML documents.
Example: `<XPathExpression Expression="root/item[matches(Title0,'^\s*{County}-{Middle}\s*{Number}$')]/Title0" Member="FoundPlate"></XPathExpression>`

10.3.3 Element Predefined

The “Predefined” element contains data which is fixed for the system such as e.g. a client number. It may contain some special names to be used for calculations:

- *#dt* – calculates the timestamp of the current date and time
- *#mul(val1, val2)* – multiplies val1 and val2
- *#div(val1, val2)* – divides val1 by val2
- *#add(val1, val2)* – adds val2 to val1
- *#sub(val1, val2)* – subtracts v2 from val1
- *#LinuxDt2DtOffset(val)* – converts Linux timestamp in seconds(int) or milliseconds(long) into a DateTime value.
- *#HWInstFromGlobalChannel* – calculates the hardware instance id from the global channel number. This only works if the camera is conncted to G-Core to which GConnect is also connected.
- *#NPRTemplate (val1, val2, val3)* – operates a string (val1) with help of a template (val2) and another string (val3) containing characters to be inserted into the first string (val1). E.g. #NPRTemplate((NRGB1234), (XX:XXZZZZ), (-)) results in “NR-GB 1234”.
- *#DateTimeFormat(val1, val2)* - Converts the DateTime value given in val1 into a date time string controlled by the format string given in val2.
  E.g. #DateTimeFormat((2021/08/10 13:09:55), (dd.MM.yyyy HH:mm:ss)) results in “10.08.2021 13:09:55”
- *#ServerVersion (val)* – Requests the software version of the given recorder. *val* represents the server id defined in the dedicated plug-in configuration.
- *#SDKVersion* – Requests the SDK version the recorder plug-in is based on.
- *#ServerGUID (val)* – Requests the unique ID of the recorder represented by *val*.
- *#ServerMediaChannels (val)* – Requests the number of media channels of the recorder represented by *val*.
- *#ServerInputs (val)* – Requests the number of assigned digital inputs of the recorder represented by *val*.
- *#ServerOutputs (val)* – Requests the number of assigned digital outputs of the recorder represented by *val*.
- *#LocalIPAddr* – Requests the IP address of the machine the G-Connect service is running on.
- *#LocalMAC* – Requests the MAC of the machine the G-Connect service is running on.
- *#Trim(val)* – removes all leading and trailing whitespaces from val.
- *#RegEx(val1, val2)* – Evaluates a string *val2* with a regular expression val1 containing groups. Regular expressions with groups can be tested here.
  Example:
  #RegEx(^(?<id>[0-9]{1,5})\s?(?<product>.*), {Products}). This expression separates the members id and product from the member Products. This only works with strings.

- *#UTCToLocalDateTime(val)* – Converts a UTC DateTime value into a local DateTime value.
- *#Replace((val1),(val2),(val3))* – Replaces all occurrences of string val2 in val1 with string val3. If an empty string shall be set, the text “null” must be written.
  Example: #Relace((NR GB999), ( ), (null)) removes all spaces in “NR GB999”. The result is NRGB999.

  ⚠ This function is available as of version 4.0.0.46.
- *#LocalHostname* – Retrieves the hostname of the local device.

  ⚠ This function is available as of version 4.0.0.195.

⚠ As of version 4.0.0.39 for functions with multiple parameters each parameter must be additionally enclosed in brackets. E.g: *#RegEx((val1), (val2))*.

⚠ As of version 4.1.0.8 the function *#DateTime2LinuxTimeSeconds(DateTime value)* converts a DateTime value into a Unix timestamp.

The operations *#mul*, *#div*, *#add* and *#sub* only work with integers and double values. The latter may contain decimal points and exponents.
The operation *#LinuxDt2DtOffset* only operates on int and long values.
The val1 parameter in *#DateTimeFormat* function must be type of DateTime, DateTimeOffset or string. The string type must contain a valid date and time representation e.g. ISO 8601.

The “Member” child of “Predefined” element maps the predefined value to any message. The “Type” attribute is currently not used.

The member value may contain a placeholder to be exchanged by another member value.
Example:

```
<Member Name="FaceRecImageDestPath"
Value="d:\Projects\CyberExtruder\Images\MCSImages\Ch{Channel}\FaceRecImage.jpg" Type="String"/>
```

{Channel} will be exchanged by its value.

⚠ As of 1.1.1.102 this also works for members to be calculated by rules.
For supported data types (attribute “Type”) see 10.3.11.

10.3.4 Element MappingList

The element “MappingList” contains a list of different mappings.

The element “Mapping” defines the way the service behaves. It defines the flow of messages from one plug-in to another and which values from the received message are taken to the send messages.
A mapping is devided into a message source (RequestSource) and a message destination (RequestDestination). The destination devides into a *Write*, *Read* and *Response* part.
“Write” sends a message to the destination plug-in. “Read” reads a message from the destination plug-in. “Response” sends a message back to the requesting plug-in. “Response” only works after “Read”.
Reference Ids to an XPathExpressionSet, RuleSet, XMLFile and XMLStylesheet have to start with value 1. The value 0 is reserved for “no reference”.
In one request multiple Write, Read and Response actions are allowed.

10.3.5 Element Stylesheets

The “Stylesheets” element contains a list of stylesheets to validate incoming XML documents. The “Path” attribute of “Stylesheets” defines the stylesheet file including its path.

10.3.6 Element XMLFiles

The “XMLFiles” element contains a list of XML files which are used for sending messages to a plug-in.

10.3.7 Element Rulesets

“Rulesets” are a mighty toolset to calculate dependencies of data fields in the messages and decide which message to send depending on the content of an incoming message.

“Rules” consist of “IF” and “ELSE” clauses, “AND” and “OR” conjunctions and conditions.
A “Condition” may compare *Int*, *String* or *DateTime* type values. *DateTime* type values are handled as *DateTimeOffset* including the time zone.

“Value” may contain a member name in curly braces. Before processing the condition it will then be exchanged by the dedicated member value.

10.3.8 Element LookupTables

As of version 4.0.0.41 the configuration may contain lookup tables. They can be defined to add multiple destination values for one source value.
Example: A source provides an area id. A camera is placed in this area to be referenced by a resulting action. The area id can be used as source to add a dedicated channel id to the resulting action which shall be sent to G-Core.

```
<LookupTables>
  <LookupTable SrcMember="AreaId">
      <SrcMemberValue Value="1">
          <DstMember Member="GlobalChannelId" Value="42"/>
          <DstMember Member="RecorderId" Value="Local"/>
      </SrcMemberValue>
      <SrcMemberValue Value="2">
          <DstMember Member="GlobalChannelId" Value="57"></DstMember>
          <DstMember Member="RecorderId" Value="Recorder_001"/>
      </SrcMemberValue>
  </LookupTable>
</LookupTables>
```

10.3.9 Namespaces list

As of version 4.1.0.8 a namespace list was added to the configuration file. It can be referenced from XPath expressions to use namespace abbriviations.

10.3.10 Configuration examples

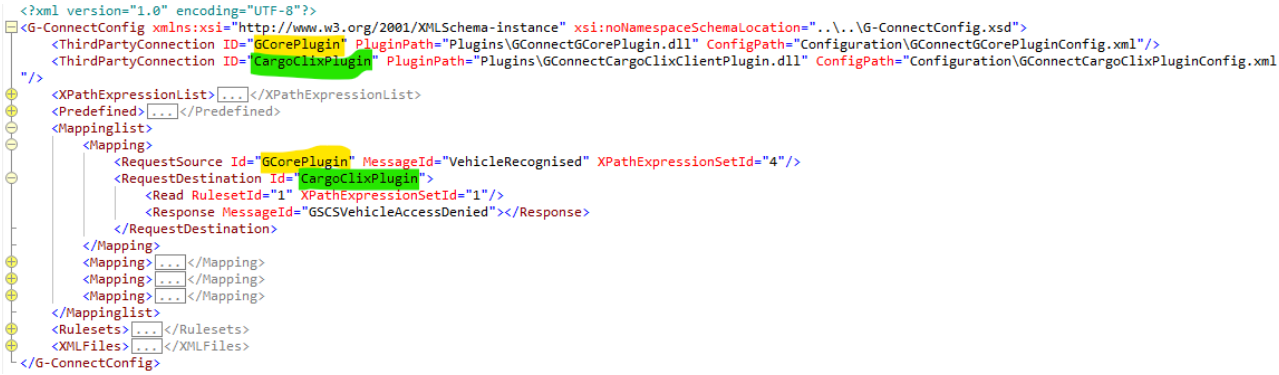

Figure 23: Reference to ThirdPartyConnection ID in Mapping

Figure 24: Reference to XPathExpressionSets and Rulsets in Mapping

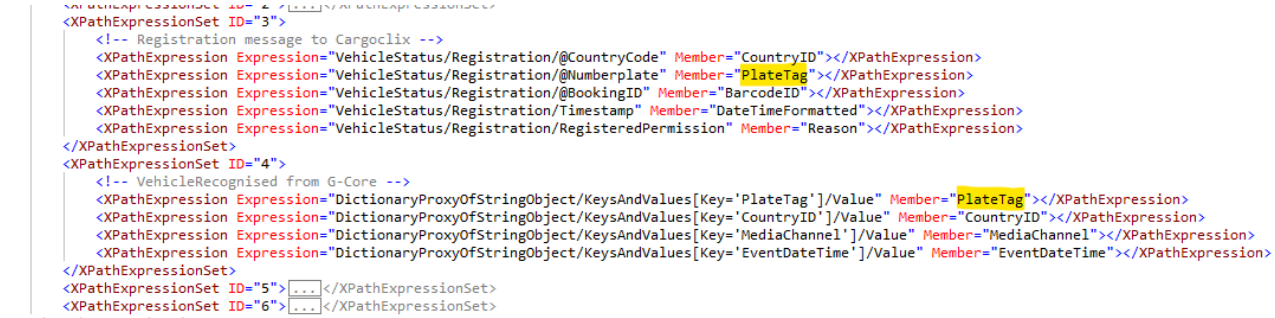

Figure 25: References between XPathExpressions

```
<XPathExpressionSet ID="1">
    <!-- Read from Cargoclix with plate tag -->
    <XPathExpression Expression="Vehicles/Vehicle[Numberplate=&quot;{PlateTag}&quot;]/StartTime" Member="TimerangeStart"></XPathExpression>
    <XPathExpression Expression="Vehicles/Vehicle[Numberplate=&quot;{PlateTag}&quot;]/EndTime" Member="TimerangeEnd"></XPathExpression>
    <XPathExpression Expression="Vehicles/Vehicle[Numberplate=&quot;{PlateTag}&quot;]/BookingID" Member="BarcodeID"></XPathExpression>
</XPathExpressionSet>
<XPathExpressionSet ID="2">...</XPathExpressionSet>
<XPathExpressionSet ID="3">...</XPathExpressionSet>
<XPathExpressionSet ID="4">
    <!-- VehicleRecognised from G-Core -->
    <XPathExpression Expression="DictionaryProxyOfStringObject/KeysAndValues[Key='PlateTag']/Value" Member="PlateTag"></XPathExpression>
    <XPathExpression Expression="DictionaryProxyOfStringObject/KeysAndValues[Key='CountryID']/Value" Member="CountryID"></XPathExpression>
    <XPathExpression Expression="DictionaryProxyOfStringObject/KeysAndValues[Key='MediaChannel']/Value" Member="MediaChannel"></XPathExpression>
    <XPathExpression Expression="DictionaryProxyOfStringObject/KeysAndValues[Key='EventDateTime']/Value" Member="EventDateTime"></XPathExpression>
</XPathExpressionSet>
<XPathExpressionSet ID="5">...</XPathExpressionSet>
<XPathExpressionSet ID="6">...</XPathExpressionSet>
</XPathExpressionList>
<Predefined>...</Predefined>
<Mappinglist>...</Mappinglist>
<Rulesets>
    <Ruleset ID="1">
        <Rule>
            <IfClause MessageId="GSCSVehicleAccessGranted">
                <Conjunction>
                    <AND>
                        <Condition Member="TimerangeStart" Operand="LessThanOrEqual" Value="{EventDateTime}"></Condition>
                        <Condition Member="TimerangeEnd" Operand="greaterThanOrEqual" Value="{EventDateTime}"></Condition>
                    </AND>
                </Conjunction>
                <Member Name="Reason" Value="Granted" Type="String"></Member>
            </IfClause>
            <ElseClause>...</ElseClause>
```

Figure 26: References between XPathExpressions and Rules

```
<XPathExpressionSet ID="3">
    <!-- Registration message to Cargoclix -->
    <XPathExpression Expression="VehicleStatus/Registration/@CountryCode" Member="CountryID"></XPathExpression>
    <XPathExpression Expression="VehicleStatus/Registration/@Numberplate" Member="PlateTag"></XPathExpression>
    <XPathExpression Expression="VehicleStatus/Registration/@BookingID" Member="BarcodeID"></XPathExpression>
    <XPathExpression Expression="VehicleStatus/Registration/Timestamp" Member="DateTimeFormatted"></XPathExpression>
    <XPathExpression Expression="VehicleStatus/Registration/RegisteredPermission" Member="Reason"></XPathExpression>
</XPathExpressionSet>
<XPathExpressionSet ID="4">
    <!-- VehicleRecognised from G-Core -->
    <XPathExpression Expression="DictionaryProxyOfStringObject/KeysAndValues[Key='PlateTag']/Value" Member="PlateTag"></XPathExpression>
    <XPathExpression Expression="DictionaryProxyOfStringObject/KeysAndValues[Key='CountryID']/Value" Member="CountryID"></XPathExpression>
    <XPathExpression Expression="DictionaryProxyOfStringObject/KeysAndValues[Key='MediaChannel']/Value" Member="MediaChannel"></XPathExpression>
    <XPathExpression Expression="DictionaryProxyOfStringObject/KeysAndValues[Key='EventDateTime']/Value" Member="EventDateTime"></XPathExpression>
</XPathExpressionSet>
<XPathExpressionSet ID="5">...</XPathExpressionSet>
<XPathExpressionSet ID="6">...</XPathExpressionSet>
</XPathExpressionList>
<Predefined>
    <!-- Porsche -->
    <Member Name="DateTimeNow" Value="#dt" Type="String"/> <!-- #dt will be exchanged by the current timestamp -->
    <Member Name="DateTimeFormatted" Value="#DateTimeFormat({EventDateTime}), (dd.MM.yyyy HH:mm:ss)" Type="String"/>
</Predefined>
```

Figure 27: References between Predefined values and XPathExpressions

```
<XPathExpressionList>
    <!-- aus Waagen-Datei -->
    <XPathExpressionSet ID="1">...</XPathExpressionSet>
    <!-- zu SQLite Plugin Insert -->
    <XPathExpressionSet ID="2">...</XPathExpressionSet>
    <!-- aus Kassendatei -->
    <XPathExpressionSet ID="3">
        <XPathExpression Expression="rdc:LineItemDetail/rdc:ScanCode" Member="ScanCode" NamespaceId="1"/>
        <XPathExpression Expression="rdc:LineItemDetail/rdc:ExtendedPrice" Member="ExtendedPrice" NamespaceId="1"/> <!-- Float 1.55 EUR -->
    </XPathExpressionSet>
    <!-- zu SQLite Plugin Select -->
    <XPathExpressionSet ID="4">...</XPathExpressionSet>
    <!-- von SQLite Plugin Antwort auf Select und Cleanup -->
    <XPathExpressionSet ID="5">...</XPathExpressionSet>
    <!-- Zu SQLite Plugin Delete -->
    <XPathExpressionSet ID="6">...</XPathExpressionSet>
    <!-- zu G-Core Datensatz Zeitüberschreitung -->
    <XPathExpressionSet ID="7">...</XPathExpressionSet>
</XPathExpressionList>
<Predefined>...</Predefined>
<Mappinglist>...</Mappinglist>
<Namespaces ID="1">
    <Namespace Key="rdc" URI="http://invia.fujitsu.com/RetailDATACenter/rdc.xsd"></Namespace>
</Namespaces>
```

Figure 28: Refrence between XPath expression and Namespaces list

10.3.11 Supported data types and formats

The following data types are currently supported:

- **Int32**: -2147483648 .. 2147483647
- **Int64**: -9223372036854775808 .. 9223372036854775807

- **Double**: ddd,ddd.ddd with d = digit. Thousands may be separated, decimal digits start after the decimal point.
- **String**: all printable charachters
- **DateTime**: YYYY/MM/DD HH:mm:ss,ttt GMT+/-HH:mm with YYYY = Year 4 digits, MM = Month 2 digits, DD = Day 2 digits, HH = Hour 2 digits, mm = Minutes 2 digits, ss = Seconds 2 digits, ttt = Milliseconds 3 digits, GMT… = Timezone
- **TMediaChannelID**: The field value must either contain the global channel id or the channel name. G-Connect parses the channel configuration of the connected G-Core and tries to match the global channel id or name. If the channel can be found, it is used for responding action. The G-Core plug-in extends the member name by *_ChannelName*, *_GlobalNo* and *_ChannelId*.
- **TPlcRect**: currently not supported
- **TEventTypeID**: currently not supported
- **Enum**: The field EnumTyp must contain a value from the list GEUTEBRUECK.Gng.PLC.ParameterFlag. The corresponding value in the response action is set to the correct enum value.
- **DateTime_ISO**: The field value must contain a date/time in ISO 8601 format. (*YYYY-MM-DD***T***hh*:*mm*:*ss+02*:*00*)
- **Bool**: Boolean values ***true*** and ***false*** in all allowed spellings.
- **TResourceID**: As field of a received action the G-Core plug-in extends the member name by *_GUID* and *_Name*.

Data presentation is location specific, it may differ for different countries.

10.4 RESTful Client configuration

As of version 4.0.0.57 a generic RESTful client is part of the core software. Already developed and future plug-ins may make use of it.
ServerIP and **ServerPort** define the destination of a request. If the **ServerPort** is set to 0 (zero) or not even set, it will be left out in the URL. If **ServerUser** and **ServerPassword** are set, an authentication header with username and password combined to a base64 formatted string will be added to the request. **UseSSL** defines if an HTTP or HTTPS request is made. ValidateSSLCertificate defines whether a certificate shall be validated. If it is set to false, the connection will work even if a certificate is missing or faulty. In this case a warning is written to the log file. The **URLList** contains extensions for different requests which are references by the **URL** attribute **ID.** A fix URI part can be defined as attribute to the element **URLList**. This part will always be inserted between the IP address / port part and the URI part defined in the child elements **URLString**. The **URL** attribute **Command** defines the type of the command. *GET*, *POST*, *PUT* and *DELETE* are supported. The **ContentType** defines the format of the content in case of commands *POST* or *PUT*. It depends on how the content is provided by the calling plug-in. In most cases *application/json* is used which is the default value. **AcceptedContentTypes** contains a list of content types to define the format responses are expected. **RequestHeader** contains a list of key/value pairs to be added to the header of a request. Configured values overwrite potentially hardcoded values.
A complete description of the REST client configuration structure can be found in Linked Documents\RESTAPIClient.pdf.

As of version 4.0.0.191 the optional element **AuthenticationType** was introduced. Possible values for **Type** are *None*, *Basic* or *Digest*. Anyway if **ServerUser** is empty, no authentication is requested.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2018 sp1 (x64) (http://www.altova.com)-->
<REST_API_ClientConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="..\..\RESTAPIClient.xsd">
    <!-- Name or IP address of the server -->
    <ServerIP>iodine.onlineintelligence.co.za</ServerIP>
    <!-- Port of the server. In case the default port 80 shall be used, this element maybe deleted -->
    <ServerPort>9443</ServerPort>
    <!-- For basic authentication a username must be added here. If no authentication is needed, this element maybe left blank or deleted -->
    <ServerUser></ServerUser>
    <!-- If needed the password must be encrypted. Use the AesPasswordEncrypter.exe tool. If not needed, this element maybe left blank or deleted -->
    <ServerPassword></ServerPassword>
    <!-- If a ServerUser is set, authentication may be requested according to the AuthenticationType setting. Basic and Digest are supported -->
    <AuthenticationType Type="None"/>
    <!-- Set to true if SSL encrypted transport shall be used. Default value is false. If no SSL encrypted transport is needed, the element maybe deleted -->
    <UseSSL>true</UseSSL>
    <!-- List of destination URLs. The ID references the command set in the XMLConnect...PluginConfig.xml -->
    <URLList>
        <URL ID="PostMessage" URLString="Integrations/web/OAFRestService?deviceType=Geutebruck" Command="POST" ContentType="application/json"/>
    </URLList>
    <!-- List of accepted content types. Defines the data formats to be accepted by the client -->
    <AcceptedContentTypes>
        <AcceptedContentType>application/json</AcceptedContentType>
    </AcceptedContentTypes>
    <RequestHeader>
        <RequestHeaderItem Name="APIKey" Value="
IylusvJHaZyJiuTOkUzrkEmmwMWM77jnpJpwtpP0nTxRyyOixmFV6ue0wiGJeCnzcewHqyZSjvVoquSHJGUneQcktQp7kAM6imltHuAjgb9JlDlhfekTDSixdEWpvezG"/>
        <RequestHeaderItem Name="Handler" Value="za.co.onlineintelligence.applications.ob.signaltower.webservice.SignalTowerRestServiceRequest"/>
    </RequestHeader>
</REST_API_ClientConfig>
```

The URL for the PostMessage request in this example looks as follows:
https://iodine.onlineintelligence.co.za:9443/Integrations/web/OAFRestService?deviceType=Geutebruck

⚠ As of version 4.0.0.204 the url may contain variables which are filled at runtime.
Example:

```
<URLList>
    <URL ID="CounterReset" URLString="api/observables/{CounterId}/count.json" Command="PUT" ContentType="application/json"/>
</URLList>
```

Additionally the REST client plug-in configuration must contain the variable name

```
<POSTCommands>
    <POSTCommandID ContentType="text" UrlVarIds="CounterId">CounterReset</POSTCommandID>
</POSTCommands>
```

10.5 Plug-ins

Plug-ins use their own configuration because requirements may vary. Configuration files for plug-ins are also located in the configuration path of the basic service. The following sections describe the configurations of the currently available plug-ins.

⚠ As of version 4.0.0.44 the path to the plug-in configuration file can be set in the G-Connect configuration file (see 10.3.1)

10.5.1 G-Core Plug-in

Please refer to the configuration file documentation Linked Documents/GConnectGCorePluginConfig.pdf.
Below there are some more descriptions of the elements and attributes used in this configuration.

The “GCoreServer” element defines the connection parameters to a G-Core server and the repeat interval for the live sign action.
The attribute “GngPort” is currently not used.
The “GngPassword” can either contain an encrypted or decrypted value. The “IsEncryptedPassword” attributes controls how the password is processed.
The “LivecheckTimeout” attribute defines the interval to repeat the livecheck CustomAction.

⚠ As of version 2.0.0.0 multiple GCoreServer entries are allowed. To identify each server a string type ID is introduced.

⚠ As of version 4.0.0.53 the new element “ServerIdKey” contains the member name to reference the Server Id in an XPath expression in G-ConnectConfig.xml.

The “Actionlist” element lists all G-Core actions used to initiate requests and carry responding data.

The “GngActionCfg” element contains the name and description of a single G-Core action representation. The attribute “ActionReferenceID” identifies the action in a message.

The attribute “SuppressDuplicates” is currently not used.

“Subscribe” registeres the action to be received by the PLC.

The “ActionDescription” element carries the parameter description and optional constraints to refer to dedicated field contents.

The “ActionParam” element describes a single parameter of a G-Core action. The “Member” attribute links the parameter value to other parts of the configuration. It may take special formatting including regular expressions or names in curly braces, depending on the use of the action. If the action shall be used as request for querying data from an XML source, the action parameter may look like this:

```
<ActionParam Name="String" Type="String" Member="^Plate:\s?(?&lt;
Plate&gt;.*),\s?ChannelNo:\s?(?&lt;ChannelNo&gt;.*)"/>
```

Here the parameter carries multiple member variables to be processed when the action is received from G-Core. “<” and “>” must be substituted by their xml symbols “<” and “>”.
For use as response action to return data to G-Core, the variant `<ActionParam Name="PlateNo" Type="String" Member="Plate: {Plate}, AreaName: {AreaName}"/>` is used. The names marked with curly brackets are exchanged by their connected values.
For supported data types in attribute “Type” see 10.3.11. To convert a member into “TMediaChannel” it must contain either the global camera ID or the camera name. A camera channel with this global ID or name must be configured in G-Core.
The attribute “optional” is currently not used.

The attribute “Key” is currently not used.

The attribute “EnumType” defines the value of the *ParameterFlag* enumeration. This can be used for every enum type parameters in the G-Core Action parameter description.
Example: The parameter “Gender” in the action “FaceRecResult” is of type “FRGender” which is an enumeration. The configuration looks like following:

```
<ActionParam Name="Gender" Type="Enum" optional="true" Member="FaceRecGenderGng" EnumType="pf_EnumFRGender"/>
```

Which “EnumType” name to be used please look up in the following list:

```
public const int pf_EnumDigitalInputState = 4096;
public const int pf_EnumPPCableKind = 4169;
public const int pf_EnumPPAlarmState = 4168;
public const int pf_EnumTrafficDirection = 4167;
public const int pf_EnumGTectClientVCAType = 4166;
public const int pf_EnumGTectAlarmReason = 4165;
public const int pf_EnumPlcSceneStoreModificationType = 4164;
public const int pf_EnumPPSensorKind = 4170;
public const int pf_EnumPlcLpsStatus = 4163;
public const int pf_EnumPlcExportAbort = 4161;
```

```
public const int pf_EnumPlcExportSuccess = 4160;
public const int pf_EnumPlcBackupFormat = 4159;
public const int pf_EnumPlcExportMarker = 4158;
public const int pf_EnumPlcSkidataMsgCodeTransaction = 4156;
public const int pf_EnumHTTPCommandType = 4155;
public const int pf_EnumFRGender = 4162;
public const int pf_EnumPlcSkidataMsgCodeExit = 4154;
public const int pf_EnumIPSwitchOps = 4171;
public const int pf_EnumE2VCAEventType = 4173;
public const int pf_EnumSMSInterComInformation = 4193;
public const int pf_EnumSMSSystemStatus = 4192;
public const int pf_EnumSMSIOStatus = 4191;
public const int pf_EnumSMSOperatorStatus = 4190;
public const int pf_EnumSMSCardReaderError = 4189;
public const int pf_EnumSMSAccessResult = 4188;
public const int pf_EnumZoomDirection = 4172;
public const int pf_EnumSMSDoorState = 4187;
public const int pf_EnumLenelEventID = 4185;
public const int pf_EnumLenelEventType = 4182;
public const int pf_EnumLenelAccessResult = 4181;
public const int pf_EnumIODeviceType = 4176;
public const int pf_EnumLPRPlateSize = 4175;
public const int pf_EnumLPRColor = 4174;
public const int pf_EnumSMSRequestType = 4186;
public const int pf_EnumPlcSkidataMsgCodeEntry = 4152;
public const int pf_EnumPlcSkidataControl = 4151;
public const int pf_EnumPlcBacklightMode = 4150;
public const int pf_EnumPlcDatabaseStatus = 4118;
public const int pf_EnumABCapacityWarning = 4115;
public const int pf_EnumPlcViewerAlarmQueueSelection = 4114;
public const int pf_EnumPlcMessageCode = 4110;
public const int pf_EnumPlcMessageSource = 4107;
public const int pf_EnumUserLoginFailureCode = 4106;
public const int pf_EnumSystemKey = 4120;
public const int pf_EnumVideoSensorKind = 4104;
public const int pf_EnumVMDGroup = 4102;
public const int pf_EnumADArea = 4101;
public const int pf_EnumVideoInterlaceType = 4100;
public const int pf_EnumVideoSignalType = 4099;
public const int pf_EnumVideoSignalNorm = 4098;
public const int pf_EnumDigitalOutputState = 4097;
public const int pf_EnumSystemLED = 4103;
public const int pf_EnumVMDCycle = 4121;
public const int pf_EnumSafebagStep = 4122;
public const int pf_EnumPlcViewerPlayMode = 4126;
public const int pf_EnumPlcDayNightMode = 4149;
public const int pf_EnumPlcPumpStatus = 4148;
public const int pf_EnumPlcPluginState = 4147;
public const int pf_EnumPlcSpecialConstants = 4146;
public const int pf_EnumPlcNPRRestriction = 4145;
public const int pf_EnumPlcPOSStatus = 4144;
public const int pf_EnumPlcResourceChangeKind = 4142;
public const int pf_EnumPlcResourceKind = 4140;
public const int pf_EnumPlcFRCNotification = 4138;
public const int pf_EnumPlcClientType = 4136;
public const int pf_EnumPlcPluginError = 4135;
public const int pf_EnumPlcDatabaseRing = 4134;
public const int pf_EnumPlcImageExportType = 4133;
public const int pf_EnumPlcViewerAlarmState = 4131;
public const int pf_EnumPlcViewerAlarmPlayMode = 4129;
```

```
  public const int pf_Unavailable = 0;
  public const int pf_Enum = 4096;
```

⚠ As of version 4.0.0.33 the enumeration has changed:

```
public enum ParameterFlag
{
  Unavailable = 0,
  Enum = 4096,
  EnumDigitalInputState = 4096,
  EnumDigitalOutputState = 4097,
  EnumVideoSignalNorm = 4098,
  EnumVideoSignalType = 4099,
  EnumVideoInterlaceType = 4100,
  EnumADArea = 4101,
  EnumVMDGroup = 4102,
  EnumSystemLED = 4103,
  EnumVideoSensorKind = 4104,
  EnumUserLoginFailureCode = 4106,
  EnumPlcMessageSource = 4107,
  EnumPlcMessageCode = 4110,
  EnumPlcViewerAlarmQueueSelection = 4114,
  EnumABCapacityWarning = 4115,
  EnumPlcDatabaseStatus = 4118,
  EnumSystemKey = 4120,
  EnumVMDCycle = 4121,
  EnumSafebagStep = 4122,
  EnumPlcViewerPlayMode = 4126,
  EnumPlcViewerAlarmPlayMode = 4129,
  EnumPlcViewerAlarmState = 4131,
  EnumPlcImageExportType = 4133,
  EnumPlcDatabaseRing = 4134,
  EnumPlcPluginError = 4135,
  EnumPlcClientType = 4136,
  EnumPlcFRCNotification = 4138,
  EnumPlcResourceKind = 4140,
  EnumPlcResourceChangeKind = 4142,
  EnumPlcPOSStatus = 4144,
  EnumPlcNPRRestriction = 4145,
  EnumPlcSpecialConstants = 4146,
  EnumPlcPluginState = 4147,
  EnumPlcPumpStatus = 4148,
  EnumPlcDayNightMode = 4149,
  EnumPlcBacklightMode = 4150,
  EnumPlcSkidataControl = 4151,
  EnumPlcSkidataMsgCodeEntry = 4152,
  EnumPlcSkidataMsgCodeExit = 4154,
  EnumHTTPCommandType = 4155,
  EnumPlcSkidataMsgCodeTransaction = 4156,
  EnumPlcExportMarker = 4158,
  EnumPlcBackupFormat = 4159,
  EnumPlcExportSuccess = 4160,
  EnumPlcExportAbort = 4161,
  EnumFRGender = 4162,
  EnumPlcLpsStatus = 4163,
  EnumPlcSceneStoreModificationType = 4164,
  EnumGTectAlarmReason = 4165,
  EnumGTectClientVCAType = 4166,
  EnumTrafficDirection = 4167,
```

```
    EnumPPAlarmState = 4168,
    EnumPPCableKind = 4169,
    EnumPPSensorKind = 4170,
    EnumIPSwitchOps = 4171,
    EnumZoomDirection = 4172,
    EnumE2VCAEventType = 4173,
    EnumLPRColor = 4174,
    EnumLPRPlateSize = 4175,
    EnumIODeviceType = 4176,
    EnumLenelAccessResult = 4181,
    EnumLenelEventType = 4182,
    EnumLenelEventID = 4185,
    EnumSMSRequestType = 4186,
    EnumSMSDoorState = 4187,
    EnumSMSAccessResult = 4188,
    EnumSMSCardReaderError = 4189,
    EnumSMSOperatorStatus = 4190,
    EnumSMSIOStatus = 4191,
    EnumSMSSystemStatus = 4192,
    EnumSMSInterComInformation = 4193,
    EnumVCAType = 4194,
    EnumCANBusDataType = 4195,
    EnumCounterType = 4196,
    EnumCounterStatus = 4197,
    EnumTemperatureState = 4198,
    EnumTemperatureUnit = 4199,
    EnumKeepAliveDeviceType = 4200,
    EnumFailoverState = 4201,
    EnumFRIDClass = 4202
}
```

As of version 4.0.0.63 the enumeration values

```
    EnumVehicleStatus = 4203,
    EnumVCATriggerType = 4204
```

have been added.

The “ActionConstraint” element is used for filtering incoming G-Core actions by particular parameter and values. The “Value” attribute may take a regular expression to filter for data ranges as e.g. `<ActionConstraint Name="Int" Value="^[0-9]{3}$"/>.`

As of version 4.0.0.46 action constraints also work for events.

The element “Eventlist” contains a list of G-Core Event descriptions used to trigger actions inside the **G-Connect** service.

As of version 4.0.0.53 the attributes “ImageKey” and “ImageChannelKey” contain the member names to reference the event images and related channel ids in an XPath expression in G-ConnectConfig.xml. Both values may contain a {0} as placeholder for a counter value beginning with 0. If an event id is assigned to multiple channels, all event images can be retrieved and distinguished by the counter value. Example: “Image{0}” will be transferred to to the G-Connect service as “Image0”, “Image1”, etc.

As of version 4.0.0.188 it is possible to define multiple different action to trigger a single event as can be done in GSet.

The “EventReferenceID” is used for referencing the Event in the message.

The “EventState” defines which state shall be used for trigger. Only states containing event data (action data) can be used. I.e. it is not possible to trigger an internal action by a stopped state if the event is stopped by timeout.

“Subscribe” registers the event to be received by the PLC.

⚠ As of version 3.0.0.29 a new element “GngFunctionList” and its sub elements “GngFunction” is introduced. With these new elements G-Core functions can be defined to be called by the basic service. Parameters are defined by the element “GngFunctionParam”.
In this version the function **DirectImageFeed** is implemented only.

⚠ As of version 4.0.0.30 the plug-in signals some special events:

- Connect / Disconnect a G-Core server
- Connect / Disconnect a camera

Therefore the new elements **ServerConnected**, **ServerDisconnected**, **MediaChannelConnected** and **MediaChannelDisconnected** have been added. These elements define the message ids and member names to link the data to other elements in the main configuration file **G-ConnectConfig.xml**.

⚠ As of version 4.0.0.195 additional fields have been added.

10.5.2 File Watcher Plug-in

With the XML file watcher plug-in the configuration file “GConnectFileWatcherPluginConfig.xml” is included. It is structured as described in Linked Documents/GConnectFilePluginConfig.pdf.

The “FileWatcherPath” element defines the directory to be observed by the file watcher.

⚠ As of version 4.0.0.202 the element “Filter” controls the type of file to be observed. It may contain a null or empty value, the string “none” or filter strings for the file extension like “*.xml”, “*.json”, “*.csv” or “*.jpg”. Other file extensions are possible. A null or empty value or “none” mean no filtering at all. “none” is the default if no filter is set.

⚠ As of version 4.1.0.3 the boolean element “DeleteSource” is added. It controls whether the source files shall be deleted. For registering mp4 files it must be set to false because if the registered file is deleted, it cannot be processed in another plug-in. The default value is “true” for backwards compatibility.
Additionally the value “mpeg4” was added to the “Filetype” enumeration. If this type is selected, new files are just registered for operation and reported to the core to be processed in another plug-in.

10.5.3 Telnet Plug-in

The XML Telnet plug-in contains the configuration file “GConnectTelnetPluginConfig.xml”. See Linked Documents/GConnectTelnetPluginConfig.pdf.

The “Port” element defines the TCP/IP port clients can connect to.

The element “DataFormat” accepts following values:

- dfXML – XML formatted data,
- dfJson – JSON formatted data,
- dfAuto – XML or JSON formatted data,
- dfRegEx – Data wich matches the given “RegularExpression”.

10.5.4 File Provider Plug-in

The File provider plug-in needs the configuration file “GConnectFileProviderPluginConfig.xml” which uses the same stylesheet as the File watcher plug-in (see 10.5.2). In this case the element “FileWatcherPath” contains path and filename. Except the element “CacheLocally” is set to true. Then the element “FileWatcherPath” only contains the path without the filename. The extension of the file to be read determines the converter to be used. Possible extensions are “.xml”, “.json” or “.csv”.

⚠ As of version 4.0.0.202 JPEG formatted image files can be processed as well. The element “Filter” was added. It controls the type of file to be observed. “Filter” may contain a null or empty value, the string “none” or filter strings for the file extension like “*.xml”, “*.json”, “*.csv” or “*.jpg”. Other file extensions are possible. A null or empty value or “none” mean no filtering at all. “none” is the default if no filter is set.

10.5.5 Webservice File Save Plug-in

The plug-in receives data to be processed by a file with the format described in Linked Documents/NetworkCopy.pdf. It opens a Web client and calls the URL given in „Source/@Path“. The received data is stored in a file defined by „Destination“. The destination folder may be a shared network folder. In this case the element “Host” has to contain a “\\” followed by the hostname or IP address. The password must be stored encrypted (see 7).

10.5.6 Face Recognition Plug-in

The client must post it’s data in the `x-www-form-urlencoded` format as Key/Value pair. Two key types must be present and are operated: Channel and XMLData. The key names are configurable (see Linked Documents/GConnectFaceRecConfig.pdf).

In element “Route” e.g. `http://localhost:9000/FaceRec` is represented by route „{controller}/{id}“. The name „FaceRec“ is mandatory for this plug-in.

The element “Codepage” identifies the characterset used to convert the received data into text.

The “RegularExpression” for **CyberExtruder** export is defined as below

```
^Channel=(?&lt;Channel&gt;[0-9]+)\s?&amp;\s?XMLData=(?&lt;XMLData&gt;.*)
```

10.5.7 File Copy Plug-in

The structure of the configuration file GConnectFileCopyConfig.xml is described in Linked Documents/GConnectFileCopyConfig.pdf.

The element “DeleteSource” states whether the source file is deleted after copying.

This element “ClearPathsAtStartup” contains a list of folders to be cleared at service start up. Deletion is only operated if the element “DeleteSource” is set to *true*.

"Folder" contains the path and in case of a shared folder also the domain, host and credentials.
The host entry has to start with "\\" followed by the hostname or IP address.
The password must be stored encrypted. (See 7).

10.5.8 CarmenGo Plug-in

The configuration file GConnectCarmenGoPluginConfig.xml is formatted as described in Linked Documents/GConnectCarmenGoPluginConfig.pdf.

⚠ As of version 4.0.0.51 the plug-in has been improved to wor with CarmenGo 1.4.0.
It now uses the additional configuration file **RESTAPI_ClientConfig.xml**. The path of the additional file is stored in the element *AdditionalConfigurationFile* in the G-Connect/CarmenGo plug-in configuration.
There are some new elements as well. The *LoginCommandId* contains the link to the Login command Url defined in the REST API configuration file. This element carries a user name and an encrypted password. For password encryption see 7. Also new are the elements *GetRecentRecognitionCommandId* and *GetImageCommandId*. They contain the links to the particular Urls defined in the REST API configuration.

10.5.9 CargoClix Plug-in

This plug-in needs two configuration files. The file **RESTAPI_ClientConfig.xml** defines the RESTful API client commands (10.4) while the file **GConnectCargoClixPluginConfig.xml** defines the behavior of the plug-in itself.

If a basic authentication at the server is needed, it can be defined by adding the user and password. The password must be AES encrypted. An encryption tool is available (see 7 - Password encryption)

The "ID" of the "URL" element references the REST API command and its attributes. The reference is used in the **GConnectCargoClixPlugin** configuration.

The structure of the **GConnectCargoClixPluginConfig.xml** is described in Linked Documents/GConnectCargoClixPluginConfig.pdf

The element "RegistrationQueueSize" defines the maximum size of the registration cache. Setting it to 0 means there is no limitation. If the cache overflows, the oldest entry is deleted, meaning this registration dataset is lost.

If a registration dataset cannot be delivered, sending will be repeated in the defined "RegistrationRepetitionInterval" until it's received successfully.

The "RequestRepetionInterval" defines the interval to repeat requests for the current vehicle list.

The element "GETVehicleListCommandID" corresponds to REST_API_ClientConfig/URLList/URL/ID in RESTAPI_ClientConfig.xml.

The element "POSTRegistrationCommandID" corresponds to REST_API_ClientConfig/URLList/URL/ID in RESTAPI_ClientConfig.xml.

⚠ As of version 4.0.0.44 the new element "ClientConfigFile" contains the path and filename of the REST API client configuration file.

10.5.10 RESTfull API Plug-in

The configuration file GConnectRESTServicePluginConfig.xml is structured as described in Linked Documents\GConnectRESTServiceConfig.pdf.

The “Port” defines at which port the API service listens for requests.

The “Route” defines the path to be called with the URI. By default it’s just /api

The “Codepage” isn’t used in this configuration.

“JsonRoot” defines the root name for conversion of the Json data to XML data.

“JsonItem” defines the item name for conversion of the Json data to XML data.

“MultiPartRootElement” defines the name of the MultiPart root element of the XML document to be sent to the base application.

“MultiPartImageElement” defines the name of the element containing the image path.

“MultiPartFile” defines where to store the received image. It may be a shared folder as well.

⚠ As of version 4.0.0.204 the plug-in supports SSL and basic authentication.
“UseSSL” determines whether SSL (https) shall be used. Default setting is “false”.
If “User” contains a value other than null or empty, basic authentication is required.
The “Password” must be stored encrypted. (See 7).

10.5.11 Briefcam Plug-in

The configuration files for the Briefcam plug-in use the same format as the RESTful API plug-in (see 10.5.10). If pictures and video files shall be downloaded, the Webservice File Save plug-in is used additionally (see 10.5.5).

10.5.12 NightWatch Plug-in

The NightWatch plug-in configuration uses the file ***GConnectNightwatchPluginConfig.xml*** to define the communication port, the maximum request buffer length and the Logger configuration path. See Linked Documents/GConnectNightwatchPluginConfig.pdf.

10.5.13 Mercareon Plug-in

The Mercareon plug-in devides into two main tasks:

- Fetching and handling xml files
- Evaluating xml files

The configuration covers these tasks as far as senseful. See Linked Documents/GConnectMercareonPluginConfig.pdf.

The “FTPHost” element with its attributes and childs defines the FTP connection to the remote directory.

The password must be stored encrypted (see 7).

The “LocalDirectory” element defines the local path to where the files are copied. A shared folder in the local network may be used for this purpose as well. If no shared folder is used, “ShareDomain”, “ShareHost”, “ShareUser” and “SharePassword” do not need to be set.

If used, the “SharePassword” must be stored encrypted (see 7).

“ClearLocalFilePattern” defines the type of files to be handled. E.g.: *.* or *.xml.

“FetchRemoteFilePattern” defines the type of files to be copied. E.g.: *.* or *.xml.

“XMLFileExtension” defines the type of files to be evaluated. E.g.: .xml.

“DateTimePattern” defines the structure of the date/time entries in the data files.
E.g.:”yyyy-MM-dd HH:mm:ss”.

“FileRootElementName “defines the name of the root element in the source xml file.

“ListRootElementName” defines the name of the root element in the xml document as part of the locally held booking list.

“XPathExpKeyCriteria” defines the XPath expression to retrieve the data field which functions as key for each data element in the booking list.

“XPathListExchangeCriteria” defines the XPath expression to retrieve the data field which is used as key for exchanging the dataset in the booking list.

“XPathListRemoveCriteria” defines the XPath expression to retrieve the data field which is used as key to decide if a dataset will be removed from the booking list.

“TimeOfDayToStartClearing” defines the time of day when clearing of the local path will be started.

“DaysToKeepFiles” defines the number of days data file will be kept.

“FTPWatcherInterval” defines the interval in milliseconds to check the remote directory for new files.

10.5.14 REST Client Plug-in

The plug-in uses two configuration files:

- GConnectRESTClientPluginConfig.xml defines the behavior of the plug-in itself.
- RESTAPI_ClientConfig.xml defines the RESTFul command interface.

For configuration see Linked Documents/GConnectRESTClientPluginConfig.pdf and Linked Documents/RESTAPIClient.pdf.
The meanings of the elements in XMLConnectRESTClientPluginConfig.xml are described below:
The **POSTCommandID** references the **URL** in the **URLList** in RESTAPI_ClientConfig.xml.
The **POSTRepetitionInterval** defines the time between the repetition of a command if sending was not successful.
The **POSTQueueSize** defines the maximum number of commands contained in the queue. 0 means no limitation. If the queue overflows, the oldest commands will be skipped.
The **RequestCommandID** references the **URL** in the **URLList** in RESTAPI_Client.xml. (Currently not used)

The **RequestRepetitionInterval** defines the time between repetitions of the command if sending failed. (Currently not used)

⚠ As of version 4.0.0.44 the new element **ClientConfigFile** contains the path and filename of the REST API client configuration file.

⚠ As of version 4.0.0.204 RequestCommandID and POSTCommandID contain an additional attribute **UrlVarIds**. It contains a comma separated list of IDs to define variable parts of the URL. (See 10.4). Furthermore the element **POSTCommandID** contains an attribute **ContentType** wich may have the following values:

- text – the content is plain text which can also be a single integer value.
- xml – the content is xml formatted.
- json – the content is json formatted.

This attribute doesn't necessarily correspond to **ContentType** in the REST API client configuration.

⚠ As of version 4.0.0.211 the PostRepetitionInterval element contains an attribute for the number of repetitions. If set to -1 (default) there is no restriction. POST or PUT commands are only repeated if connection got lost.

10.5.15 TCP Client (UnityIS) Plug-in

To operate this plug-in the configuration file GConnectTCPClientPluginConfig.xml is needed. The structure is described in Linked Documents\GConnectTCPClientPluginConfig.pdf.
The element **TCPServer** defines the connection details for the UnityIS instance. An authentication is not needed yet but the **User** attribute is used for command requests.
The element **CommandID** contains a reference name used in XmlConnectConfig.xml.
The elements **Params/ParamID** contain reference names as well.
ResponseTimeout contains a value in milliseconds to await a response.

10.5.16 VideoGuard Plug-in

The VideoGuard plug-in needs the configuration file **GConnectVideoGuardPluginConfig.xml**. It's structure is described in Linked Documents\GConnectVideoGuardPluginConfig.pdf. Although part of the configuration the command and field names are fixed. If they are changed, the plug-in won't work anymore.

10.5.17 FLIR Plug-in

The FLIR plug-in needs the configuration file **GConnectFlirPluginConfig.xml**. Its structure is described in Linked Documents\GConnectFlirPluginConfig.pdf.
The elements **RequestMessage** under **RequestMessages** point to additional XML files containing command requests to be sent to the radar server. The commands are referenced by the attribute **Name** internally. In the current version the zone configuration request is hardcoded and must be called ***ZonesConfigRequest***. Other commands are not implemented yet.

10.5.18 OnGuard Plug-in

The **OnGuard** plug-in works with the configuration files ***GConnectOnGuardPluginConfig.xml*** and ***RESTAPI_ClientConfig.xml***. For a close description of the structure of ***GConnectOnGuardPluginConfig.xml*** see Linked Documents\GConnectOnGuardPluginConfig.pdf.
The structure of ***RESTAPI_ClientConfig.xml*** is described in Linked Documents\RESTAPIClient.pdf.
The connection address to **OnGuard** is set in ***RESTAPI_ClientConfig.xml***.

The URLs to the supported commands are also defined here. They are referenced by the particular IDs. For testing purpose the Application-Id "`OPEN_ACCESS_NON_PRODUCTION`" may be defined. It overwrites the hardcoded value.
GConnectOnGuardPluginConfig.xml defines the supported commands to communicate to OnGuard via the OpenAccess interface:

- **Authenticate** calls the URL referenced by the CommandId. Username, Password and DirectoryId must correspond to the assigned OpenAccess user. The password must be stored encryptedly and the directory ID must match the directory name defined in **OnGuard System Administration**. If the internal directory is used, <*Internal*> must be set here. Inside the configuration file the characters '<' and '>' must be translated to '<' and '>' because these characters are not allowed in text fields inside an XML document.
- **Logout** is called when the **G-Connect** service is stopped.
- **CreateSubscription** subscribes to the hardware events provided by OnGuard
- The **EventBridgeId** just references the URL to connect the SignalR service needed to receive the events.
- **SignalRLogLevels** defines the log level of the SignalR connection. The valid range is "All", "Events", "Messages", "None" and "StateChanges".
- **KeepAlive** sends a signal frequently to the OpenAccess service. If no response is received, the connection is probably broken and must be set up again.
- **CountEvents** defines whether incoming events are counted in G-Connect. This is only for testing purpose.
- **ExecuteMethods** collects supported methods like Alarm acknowledgement, sending generic alarms or influencing hardware contacts.
- **GetLenelType** is reserved for requesting type information from **OnGuard** and is currently not used
- **AcknowledgeEvent** is used to define the data mapping of the database.
- **GetLenelVersion** retrieves the **OnGuard** version information
- **GetLenelDirectories** retrieves the directory list defined in **OnGuard**
- ⚠ As of version 4.0.0.44 the new element **ClientConfigFile** contains the path and filename of the REST API client configuration file.

10.5.19 Start external processes Plug-in

The plug-in enables the service to start external processes. It is configured by the file **GConnectRunProcessesPluginConfig.xml**. The structure of this file is described under Linked Documents\GConnectRunProcessesPluginConfig.pdf.
For each process to be started the file contains an entry with the following attributes:

- **Filename** – The name of the file to start the process.
- **Arguments** – Command line arguments for starting.
- **User** / **Password** – Only needed if the process shall be started in the context of a user login.
- **Continuous** – Defines whether the process shall be restarted automatically if it is stopped for an reason.

The **Path** attribute of the element **Process** defines the directory containing the process executable.

⚠ As of version 4.0.0.212 the following parameters are added:

- **AtStart** – Flag to signalize if the process shall be started at startup of the plug-in.
- **MessageId** – Text to link the process to a message

The **Arguments** field may contain variables enclosed in curly brackets.

10.5.20 SQL Server notification Plug-in

The plug-in gets notofications from a connected SQL Server if dedicated rows in a table are changed or inserted.
The configuration file **GConnectSQLPluginConfig.xml** defines the connection to the database, the table to be observed and the SQL command to determine the notification. Additionally it defines a timestamp field to contain the timestamp of changing or inserting the row in the database. The structure of this file is described in Linked Documents\GConnectSQLPluginConfig.pdf.
The **Connection** element contains the following fields:

- **Datasource** – Instance of the SQL Server. For connection to a database on another device this field will not be set.
- **NetworkLibrary** – Communication protocol. Only be needed for a connection to a database on another device.

  Network protocol codes

NAME	NETWORK LIBRARY
dbnmpntw	Named Pipes
dbmslpcn	Shared Memory (local machine connections only, might fail when moving to production...)
dbmssocn	Winsock TCP/IP
dbmsspxn	SPX/IPX
dbmsvinn	Banyan Vines
dbmsrpcn	Multi-Protocol (Windows RPC)
dbmsadsn	Apple Talk
dbmsgnet	VIA

  In case of connecting via TCP/IP use dbmssocn.
- **InitialCatalog** - defines the database to be connected to.
- **User** – Username to log into the database
- **Password** – Password to log into the database. Must be encrypted using the AESEncryption tool (see section 7).
- **IntegratedSecurity** – must be set to true if the windows user shall be used for authentication and the SQL Server is running on the same machine. Otherwise it must be set to false.
- **IP** – IP address of the device if the SQL Server is running on another device.
- **Port** – Port for connecting to a database on another device.

The element **Table** defines the table to be notified.
The element **CommandString** contains the SQL command to read the notified data.
The element **TimestampField** points to the column containing a timestamp to be set whenever a row changes or a new one is inserted. This field is needed to identify the changed or inserted rows.

10.5.21 NTP client Plug-in

The NTP client plug-in configuration file **GConnectNTPClientPluginConfig.xml** just contains the identifying member names to link the values in the basic G-Connect configuration file **G-ConnectConfig.xml**. See Linked Documents\GConnectNTPClientPluginConfig.pdf.

10.5.22 Health Plug-in

The health plug-in configuration file **GConnectHealthPluginConfig.xml** mainly contains the identifying member names to link the values in the basic G-Connect configuration file **G-ConnectConfig.xml**. See Linked Documents\GConnectHealthPluginConfig.pdf.
Additionally the **UpdateInterval** element defines the frequence to check the hard disk statuses.

10.5.23 LoadWebContent Plug-in

For complete configuration structure see Linked Documents\GConnectLoadWebContentPluginConfig.pdf.
Content carries the member name for referencing the web content.
Data lists the members for transporting the meta data.
URL contains the member name for referencing the URL to fetch the web content.

As of version 4.1.0.22 the URL element may contain up to three attributes called “User”, “Password” and “Domain”. If “User” is set, the plugin tries to authenticate the user. The password must be stored encryptedly.

10.5.24 Message blocker plug-in

For complete configuration structure see Linked Documents\GConnectMessageBlockerPluginConfig.pdf.
For each message to block messages of the same type a timeout can be defined. After the timeout expired the message will be passed even if the blocking condition is still fulfilled.
If a message of the same type is received but the condition is not met, it will pass and will be the base for the next received messages of this type. The timeout is started again.
Only one or multiple conditions may be defined. Each condition relates to a member field of the message. For comparison two operations are available: “equal” or “unequal”.

Example:
Repeating messages of the type ONVIFNotification shall be blocked if the content of field “Plate” is the same. Blocking shall be active for 5 Seconds.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2018 sp1 (x64) (http://www.altova.com)-->
<GConnectMessageBlockerPluginConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="
..\..\GConnectMessageBlockerPluginConfig.xsd">
    <Message ID="OnvifNotificationMessage" BlockingTimeout="5000">
        <Condition Member="Plate" Operation="equal"/>
    </Message>
</GConnectMessageBlockerPluginConfig>
```

10.5.25 Message combine plug-in

For complete configuration structure see Linked Documents\GConnectMessageCombinePluginConfig.pdf.
Messages dependant on each other are organized in groups. The configuration defines these groups with a message type “SuccessMessageID” for response if all messages of this group are received and a message type “FailedMessageID” if at least one message is missing after the receive “Timeout” is expired. Messages can be filtered by constraints, i.e. to be part of the group a message must fulfill a defined constraint. E.g. an incoming CustomAction is only assigned to a dedicated group if its Int value equals 5.
If a message is received multiple times while the plug-in is waiting for expiration of the “Wait” timeout, an “ExchangeCriterion” dedicates whether the values in the message group are exchanged or not.

10.5.26 SAP SOAP client

For the complete structure of the configuration file see Linked Documents\GConnectSAPSOAPClientConfig.pdf.

The parameter "**Binding**" links to the binding description in *%GConnectService%\Plugins\GConnectSAPSOAPClient.dll.config*.
"**Authentication**" contains username and password (encrypted) to authenticate to the SOAP service.
"**SAPProtocol/SAPField**" contains the data fields to be sent to the service where "**Key**" is the fieldname in the service and "**Value**" links to the xml data provided by the G-Connect core (see 10.3.2).

10.5.27 SQLite database

For the complete structure of the configuration file see Linked Documents\GConnectSQLitePluginConfig.pdf

Database defines the path and the file name of the SQLite database.
Table defines the **Create**, **Insert**, **Select** and **Delete** commands for querying the tables of the database. Reference names to the G-Connect configuration must be enclosed in curly braces.
The **Cleanup** command determines the rule for regularly deleting rows from a table.

10.5.28 Process XML files

11 Used Licenses

Some parts of the application and plug-ins use third party libraries.
The following table shows which third party libraries are used and under which license they are provided:

Library	Version	License	Link to license terms
Microsoft.AspNet.WebApi.Client	5.2.9	Microsoft Software License	https://www.microsoft.com/web/webpi/eula/net_library_eula_ENU.htm
Microsoft.AspNet.WebApi.Core	5.2.9	Microsoft Software License	https://www.microsoft.com/web/webpi/eula/net_library_eula_ENU.htm
Microsoft.AspNet.WebApi.Owin	5.2.9	Microsoft Software License	https://www.microsoft.com/web/webpi/eula/net_library_eula_ENU.htm
Microsoft.AspNet.WebApi.OwinSelfHost	5.2.9	Microsoft Software License	https://www.microsoft.com/web/webpi/eula/net_library_eula_ENU.htm
Microsoft.Owin	4.2.2	Apache License	http://www.apache.org/licenses/LICENSE-2.0
Microsoft.Owin.Host.HttpListener	4.2.2	Apache License	http://www.apache.org/licenses/LICENSE-2.0
Microsoft.Owin.Hosting	4.2.2	Apache License	http://www.apache.org/licenses/LICENSE-2.0
Newtonsoft.Json	13.0.3	MIT license	https://licenses.nuget.org/MIT
Owin	1.0.0	Apache License	http://www.apache.org/licenses/LICENSE-2.0
Newtonsoft.Json.Schema	3.0.14	Propriate	https://raw.githubusercontent.com/JamesNK/Newtonsoft.Json.Schema/master/LICENSE.md
WinSCP	5.17.10	MPL-2.0	https://licenses.nuget.org/MPL-2.0
Log4net	2.0.12	Apache License	http://www.apache.org/licenses/LICENSE-2.0
SuperSocket	1.6.6.1	Apache License	http://www.apache.org/licenses/LICENSE-2.0
SuperSocket.Engine	1.6.6.1	SuperSocket.Engine	http://www.apache.org/licenses/LICENSE-2.0
SoapHttpClient	3.0.0	GNU GPL 3.0	https://www.gnu.org/licenses/gpl-3.0.html

WebSocketSharp-netstandard	1.0.1	MIT license	https://github.com/PingmanTools/websocket-sharp/blob/master/LICENSE.txt
Microsoft.Bcl.AsyncInterfaces	5.0.0	MIT license	https://licenses.nuget.org/MIT
Microsoft.Extensions.Configuration	5.0.0	Apache-2.0	http://www.apache.org/licenses/LICENSE-2.0
Microsoft.Extensions.Configuration.Abstractions	5.0.0	Apache-2.0	http://www.apache.org/licenses/LICENSE-2.0
Microsoft.Extensions.Configuration.Binder	5.0.0	Apache-2.0	http://www.apache.org/licenses/LICENSE-2.0
Microsoft.Extensions.DependancyInjection	5.0.1	Apache-2.0	http://www.apache.org/licenses/LICENSE-2.0
Microsoft.Extensions.DependancyInjection.Abstractions	5.0.0	Apache-2.0	http://www.apache.org/licenses/LICENSE-2.0
Microsaft.Extensions.Http	5.0.0	Apache-2.0	http://www.apache.org/licenses/LICENSE-2.0
Microsoft.Extensions.Logging	5.0.0	Apache-2.0	http://www.apache.org/licenses/LICENSE-2.0
Microsoft.Extensions.Logging.Abstractions	5.0.0	Apache-2.0	http://www.apache.org/licenses/LICENSE-2.0
Microsoft.Extensions.Options	5.0.0	Apache-2.0	http://www.apache.org/licenses/LICENSE-2.0
Microsoft.Extensions.Primitives	5.0.0	Apache-2.0	http://www.apache.org/licenses/LICENSE-2.0
System.Buffers	4.5.1	MIT license	https://licenses.nuget.org/MIT
System.ComponentModel.Annotations	5.0.0	MIT license	https://licenses.nuget.org/MIT
System.Memory	4.5.4	MIT license	https://licenses.nuget.org/MIT
System.Net.Sockets	4.3.0	Microsoft software license	https://dotnet.microsoft.com/en/dotnet_library_license.htm
System.Numerics.Vectors	4.5.0	MIT license	https://licenses.nuget.org/MIT
System.Rutime.CompilerServices.Unsafe	4.7.1	MIT license	https://licenses.nuget.org/MIT
System.Threading.Tasks.Extensions	4.5.4	MIT license	https://licenses.nuget.org/MIT
Microsoft.AspNet.SignalR.Client	2.4.1	Apache-2.0	http://www.apache.org/licenses/LICENSE-2.0
System.Diagnostics.DiagnosticSource	5.0.1	MIT license	https://licenses.nuget.org/MIT
System.Runtime.CompilerServices.Unsafe	5.0.0	MIT license	https://licenses.nuget.org/MIT
System.ValueTuple	4.5.0	MIT license	https://licenses.nuget.org/MIT
EntityFramework	6.4.4	Apache-2.0	http://www.apache.org/licenses/LICENSE-2.0
Stub.System.Data.SQLite.Core.NetFramework	1.0.113.3	SQLite	SQLite Copyright
System.Data.SQLite	1.0.113.7	SQLite	SQLite Copyright
System.Data.SQLite.Core	1.0.113.7	SQLite	SQLite Copyright
System.Data.SQLite.EF6	1.0.113	SQLite	SQLite Copyright
System.Data.SQLite.Linq	1.0.113	SQLite	SQLite Copyright
XPath2	1.1.2	MS-PL	'MS-PL' reference (nuget.org)

12 Failover / Backup functionality

12.1 Overview

Failover is a new functionality implemented for covering the scenario when the G-Connect server gets down or lose the connection with the G-Core server, then another G-Connect server configured in backup mode, assumes the control loading all the plugins again, until the G-Connect main is restored.

12.2 Failover capable plugins

Only these G-Connect plugins which connects to the 3rd party server are capable to use the Failover functionality. If the 3rd party connects with a given IP address to G-Connect but G-Connect is not reachable, G-Connect Failover/Backup will not work.

Failover capable pulgins:
"RESTful client";
"TCP client";
"Mercareon";
"Denia";
"OnGuard";
"Metrasens";
"Online Intelligence";
"Carmen Go";
"Bastille";
"FLIR-PSR";
"UnityIS";
"Nedap";
"CIAS IB server";
"Fibersensys APU";
"Southwest RPMII";
"SAP SOAP Client";
"CargoClix";
"MessageBlocker";
"MessgeCombine";
"PlatesmartAPI";
"Logflows";
"CCure9000";

12.3 Configuration file

GConnectFailsafeConfig.xml should be inside Configuration folder, inside the G-CONNECT installation folder.

G-Connect in Main mode
```xml
<?xml version="1.0" encoding="UTF-8"?>
<G-ConnectFailSafeConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                         xsi:noNamespaceSchemaLocation="GConnectFailsafeConfig.xml">
    <MessagingConfiguration>
        <Model>mmXML</Model>
    </MessagingConfiguration>
    <MessagingQueueConfiguration>
        <Type>mqBASIC</Type>
    </MessagingQueueConfiguration>
    <Failsafe>
        <Mode>fsMain</Mode>
        <GCoreHost>GCore IP for G-Connect Main</GCoreHost>
    </Failsafe>
    <HealthService>
        <CredentialsConfiguration>
```

```
            <User>gcore-user for G-Connect Main</User>
            <Password>gcore-passwork</Password>
            <IsEncryptedPassword>false</IsEncryptedPassword>
        </CredentialsConfiguration>
        <EndpointConfiguration>
            <Host>G-Connect Main IP</Host>
            <Port>Listening port. Default is 9000.</Port>
            <Route>{controller}/{action}/{id}</Route>
            <UserSSL>false</UserSSL>
        </EndpointConfiguration>
    </HealthService>
    <HealthChecker>
        <ObservedGConnect>
            <Host>G-Connect Backup IP</Host>
            <Port>G-Connect Backup listening port</Port>
            <Route>{controller}/{action}/{id?}</Route>
            <UserSSL>false</UserSSL>
        </ObservedGConnect>
        <CredentialsConfiguration>
            <User>gcore-user for G-Connect Backup</User>
            <Password>gcore-passwork</Password>
            <IsEncryptedPassword>false</IsEncryptedPassword>
        </CredentialsConfiguration>
        <HealthCheckerConfiguration>
            <Interval>Interval for checking.3000 is default</Interval>
        </HealthCheckerConfiguration>
    </HealthChecker>
</G-ConnectFailSafeConfig>
```

G-Connect in Backup mode

```xml
<?xml version="1.0" encoding="UTF-8"?>
<G-ConnectFailSafeConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                         xsi:noNamespaceSchemaLocation="GConnectFailsafeConfig.xml">
    <MessagingConfiguration>
        <Model>mmXML</Model>
    </MessagingConfiguration>
    <MessagingQueueConfiguration>
        <Type>mqBASIC</Type>
    </MessagingQueueConfiguration>
    <Failsafe>
        <Mode>fsBackup</Mode>
        <GCoreHost>GCore IP for G-Connect Backup</GCoreHost>
    </Failsafe>
    <HealthService>
        <CredentialsConfiguration>
            <User>gcore-user for G-Connect Backup</User>
            <Password>gcore-passwork</Password>
            <IsEncryptedPassword>false</IsEncryptedPassword>
        </CredentialsConfiguration>
        <EndpointConfiguration>
```

```
            <Host>G-Connect Backup IP</Host>
            <Port>Listening port. Default is 9000.</Port>
            <Route>{controller}/{action}/{id}</Route>
            <UserSSL>false</UserSSL>
        </EndpointConfiguration>
    </HealthService>
    <HealthChecker>
        <ObservedGConnect>
            <Host>G-Connect Main IP</Host>
            <Port>G-Connect Main listening port</Port>
            <Route>{controller}/{action}/{id?}</Route>
            <UserSSL>false</UserSSL>
        </ObservedGConnect>
        <CredentialsConfiguration>
            <User>gcore-user for G-Connect Main</User>
            <Password>gcore-passwork</Password>
            <IsEncryptedPassword>false</IsEncryptedPassword>
        </CredentialsConfiguration>
        <HealthCheckerConfiguration>
            <Interval>Interval for checking.3000 is default</Interval>
        </HealthCheckerConfiguration>
    </HealthChecker>
</G-ConnectFailSafeConfig>
```

MessagingConfiguration
This configuration is for future. Specify the kind of message format that will be used for communication
between plugins.

MessagingQueueConfiguration
This configuration is for future. Specify the kind of queue used for the message queue. mqBASIC means
the default behavior as asynchronous queue.

Failsafe
Section for specifying the work mode for G-Connet.

- Mode: Work mode. It could be **fsMain** for G-Connect working as Main or **fsBackup** for G-Connect working as Backup.
- GCoreHost: It should be the G-Core ip where G-Connect is connected

HealthService
This is the configuration for the Health Service. Will have a **CredentialsConfiguration** section and a **EndpointConfiguration** section. These sections will have the information required for publishing the Endpoints for asking for G-Core connection and for requesting the release of the plugins for taking control process.

- **CredentialsConfiguration**
  - User: G-Core user for used for testing a G-Core connection.
  - Password: G-Core password for the user above.

- **EndpointConfiguration**
  - Host: G-Connect server IP
  - Port: Port where will be published the endpoint.
  - Route: Configuration route for the endpoint
  - UserSSL: Define if SSL will be used to access the endpoint.

HealthChecker
This is the service that will try to connect to the Health service endpoint of the other G-Connect. It role will be to ask on every moment if the other G-Connect can connect to it's respective G-Core server. Depending of the result, the current G-Connect will request the control or not.

- **ObservedGConnect**: This is the health service endpoint of the other G-Connect server.
  - Host: G-Connect server IP
  - Port: Port where will be published the endpoint.
  - Route: Configuration route for the endpoint
  - UserSSL: Define if SSL will be used to access the endpoint.
- **CredentialsConfiguration**
    - User: G-Core user for used for testing a G-Core connection.
    - Password: G-Core password for the user above.
- HealthCheckerConfiguration
  - Interval: Specify how long is the interval needed to wait for asking again to the health service endpoint

12.4 Note
It is necessary to execute G-Connect as administrator rights to allow the creating of health service endpoint

12.5 G-Connet Main workflow
1. At startup, Main will check if can connect to the Backup specified in HealthChecker configuration section.
2. If Backup is connected and Main have connection with G-Core, Main will request the control from Backup. If Main can't connect to Backup, then it will take the control anyway and report as Action.
3. When Main takes the control, plugins are loading and message queue is created. Health service and HealthChecker are created by default.

12.6 G-Connect Backup workflow
1. At startup, Backup will create the Health service and the HealthChecker.
2. If the Backup HealthChecker detects that the configured G-Connect Main is online and have connection with G-Core, then it will stay monitoring to Main.
Backup will not connect to the 3rd party.
3. If Main fails to respond to Backup, the Backup takes the control. Backup will load the plugins and connect to the 3rd party until Main request the control again.
4. If Main takes the control, Backup will release the plugins and disconnect from the 3rd party connection.

12.7 Licenseing

G-Connect main AND GConnect backup will pull a license.
The G-Connect executables resides on different hardware (PC).
GConnect backup will not pull a license for the 3rd party systems (plugins).