

GSM-Control Object for Wonderware Application Server

for M2M remote control in automation
applications using GSM communications

User Manual

Ver 1.x Rev 1.3

PR 000 68

WONDERWARE FINLAND & BALTICS
P.O. Box 38 FIN-00371 Helsinki Finland
tel. int. + 358 9 5404940
fax int. + 358 9 5413541
info@wonderware.fi
www.wonderware.fi



Table Of Contents

Overview	1
GSM-modem and accessories	2
Installing the GSM-Control Object	4
Hardware requirements	4
Software requirements	4
Licensing requirements	4
Files	5
Installation steps	5
Getting and enabling license	7
Software Key	8
Demo License	10
Licensing under License Server	11
Getting started quickly with GSM-Control Object	12
Default configuration	12
Getting started in simulation mode (without GSM-modem connected)	15
Getting started with GSM-modem connected	19
GSM-Control Object configuration in details	23
General tab	23
Users tab	25
Send/Receive tab	26
Alarms tab	28
License Server tab	30
Description of GSM-Control Object attributes	31
General Attributes	31
User Attributes	33
Send/Receive Message Attributes	33
Alarm Message Attributes	34
Error and warning messages	36
GSM-Calendar Object	39
Overview	39
Files	39
Installation steps	39
Members tab	42
Schedulers tab	43
Runtime tab	44
GSM-Calendar Object configuration in details	46
General information	46
UDAs on graphical panel	47
General Attributes	49
Scripts	52
Revision History	53

GSM-CONTROL OBJECT

Overview

GSM-CONTROL OBJECT (GSM-Control SMS Gateway Object) is a Wonderware Application Server domain object created by using the Archestra Object Toolkit and used for 2-way remote control in automation and other applications using standard GSM (Global System for Mobiles) cellular phones and GSM network. Based on the standard GSM cellular phones and other GSM-capable devices, GSM-Control Object offers a low-cost and easy alternative to create wireless control and monitoring applications. Implementation of GSM network's SMS (Short Message Service) technology secures reliable transmission even in the most error-sensitive applications.

The main task of GSM-Control Object is to work as a **gateway** between GSM environment at one side (interfacing with GSM environment by sending/receiving SMS messages) and Wonderware System Platform environment at another side.

The GSM-Control Object is installed, configured, deployed and running on Wonderware Application Server environment, allowing other Wonderware System Platform software to access to SMS data via GSM-Control Object attributes. The remote control is based on GSM SMS messaging two-direction communication, where the GSM-modem (any GSM-modem supporting standard Short Message Service (SMS) AT commands, preferable models are Cinterion MC52iT/MC55iT) is connected to the computer's standard (RS-232) serial port (or to USB port via USB to serial port adapter) and GSM cellular phone (remote GSM phone) is used by remote operator(s). Also the communication between two GSM-modems is supported, i.e. remote GSM-modems can be used instead of remote GSM phone.

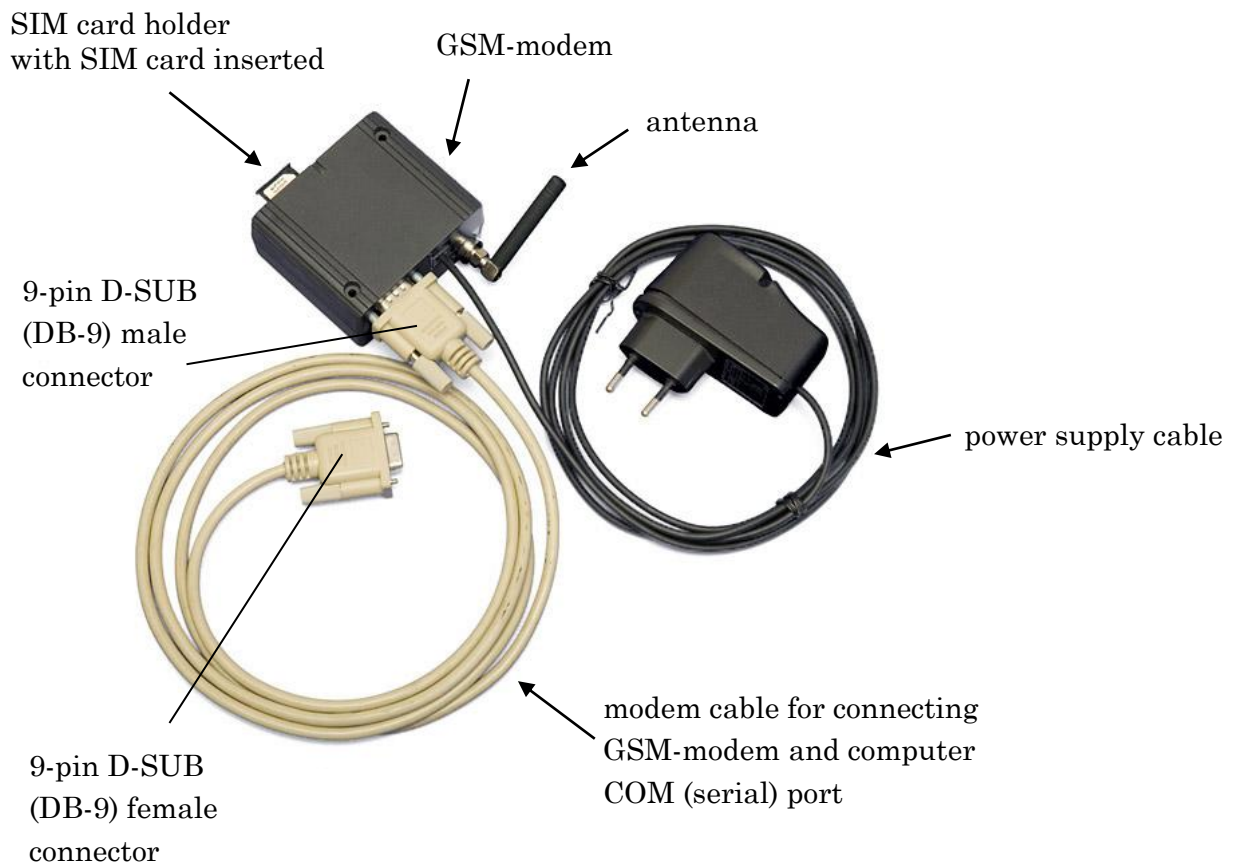
Basically, the data exchange through GSM-Control Object can be initiated both from GSM and Wonderware System Platform environments:

- 1 From **GSM** environment - by sending SMS message to GSM-Control Object, where the received message is checked and processed according the GSM-Control Object current configuration. The received SMS message can contain some data to be transferred (via GSM-Control Object attributes) to Wonderware System Platform applications, e.g. to other Application Server objects, InTouch, Historian, etc. This received SMS message can have the corresponding response message configured (also possibly containing data obtained from Wonderware System Platform applications) - in this case GSM-Control Object will respond with SMS message to the sender.
- 2 From **Wonderware System Platform** environment - when some alarm or event occurs, linked with some GSM-Control Object alarm attribute. In this

case GSM-Control Object will send the correspondingly configured SMS message (possibly containing also some data obtained from Wonderware System Platform applications) to remote GSM operator or device linked to this alarm or event condition. The receiver of such SMS message can respond to GSM-Control Object - for example, by sending some acknowledgment SMS message.

GSM-modem and accessories

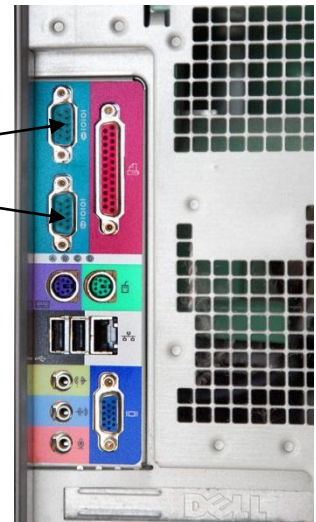
The main function of GSM-Control Object is receiving and sending SMS-messages. For this purpose the external GSM-modem must be connected to computer where GSM-Control Object is running:



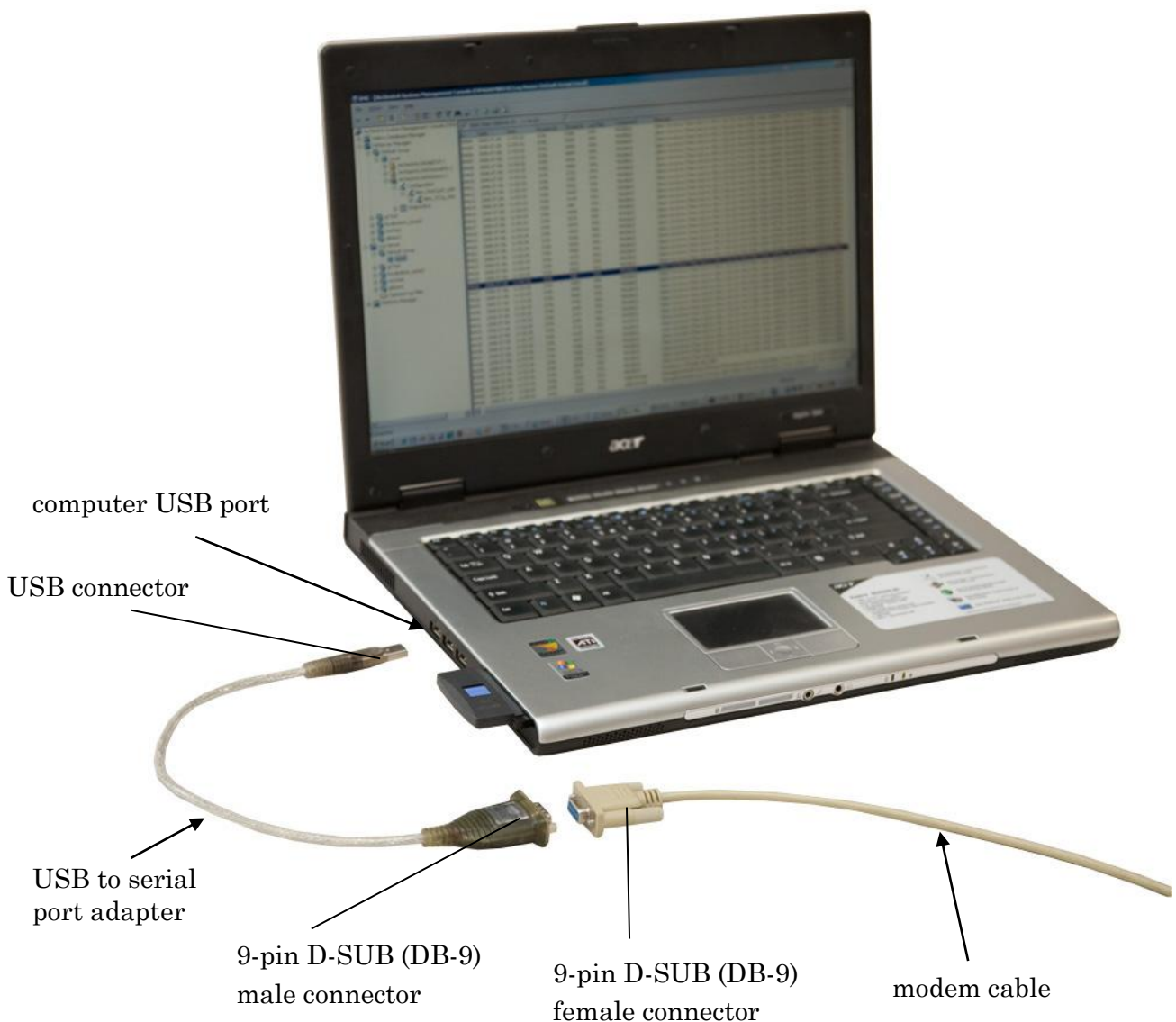
The GSM-modem and all necessary accessories (antenna, power supply cable and modem cable) are already included in **PC GSM SET** package provided by Wonderware Finland & Baltics.

The modem cable 9-pin D-SUB (DB-9) female connector must be inserted into computer COM (serial) port socket (9-pin D-SUB (DB-9) male), usually located on the rear panel of the computer:

Computer
COM ports



If the computer where GSM-Control Object is running has no COM (serial) port, an additional **USB to serial port adapter** is needed (not included in PC GSM SET package and not supplied by Wonderware Finland & Baltics). The adapter must have a 9-pin D-SUB (DB-9) male connector which must be connected to the modem cable 9-pin D-SUB (DB-9) female connector:



There are different kinds of USB to serial port adapters available on the market, as a rule also the corresponding USB/serial driver software is supplied – after installing the driver, the new virtual COM port (physically using USB port) is added to the computer; this COM port must be selected in GSM-Control Object as a COM port where GSM-modem is connected.

Installing the GSM-Control Object

Hardware requirements

The GSM-Control Object has the same hardware requirements as Wonderware Application Server. It is strongly recommended to have computer with 2 gigahertz (GHz) or faster processor, 32/64-bit. A multi-core processor is also strongly recommended. The Intel Itanium 2 processor is not supported.

Software requirements

The Wonderware Application Server 3.1 SP3 or later should be installed prior to GSM-Control Object installation.

Licensing requirements

The GSM-Control Object requires either a **software key** or a **demo license**. As well it can be licensed under **License Server** in case GSM-Control Object is delivered as a part of Wonderware Finland **Automation Intelligence Suite** package.

Without software key, demo license or License Server enabled, GSM-Control Object will support only 1 User and will work only 1 hour after GSM-Control Object deployment (in run-time).

The **demo license** is for free and provides an unlimited functionality (up to 10 Users are supported), but it is valid only for a limited time period. After demo license expiration the GSM-Control Object will support only 1 User and will work only 1 hour after GSM-Control Object deployment (in run-time). The demo license can be obtained by sending inquiry to info@wonderware.fi.

The **software key** should be purchased and it is available for 2 product levels (xxx is the current version of GSM-Control Object release, e.g. 102):

- Product ID: **PR00681 xxx** (maximum 3 Users are supported in run-time)
- Product ID: **PR00682 xxx** (unlimited version)

See *Getting and enabling license* section for details how to get and enable software key or demo license or enable licensing under License Server.

Files

The GSM-Control Object installation package contains the following files:

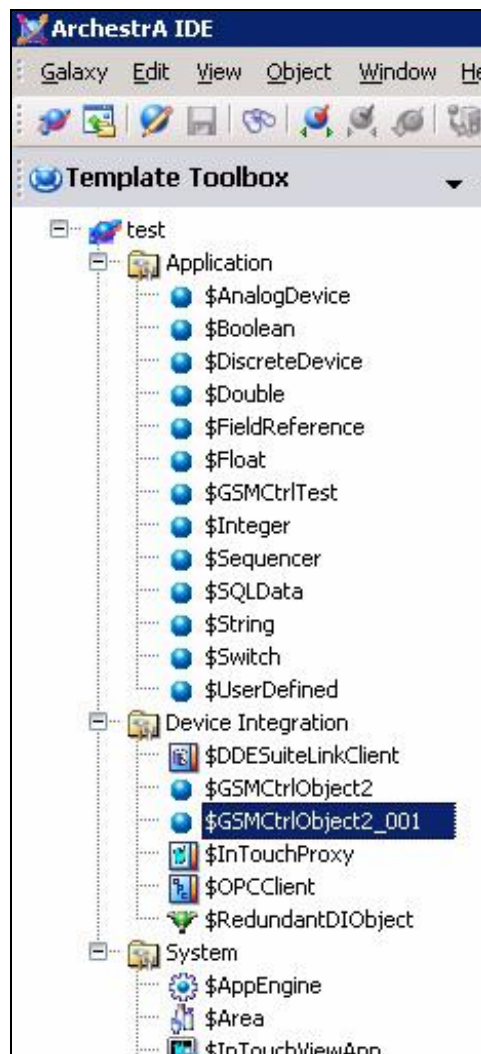
- **\$GSMCtrlObject.aaPKG** – the GSM-Control Object standard description file, containing the default configuration data and GSM-Control Object implementation code; this file can be used on Wonderware Application Server 2012 (version 3.5) and later;
- **GSMCtrlObject.aaPDF** – the GSM-Control Object standard description file, without default configuration; this file should be used on Wonderware Application Server versions earlier as 2012 (version 3.5);
- **\$GSMCtrlTest.aaPKG** – simple test demo object, containing three user defined attributes (UDAs);
- **\$GSMCalendar.aaPKG** – the GSM-Calendar Object implementation code; the GSM-Calendar Object is optional demo object showing in practice how to access the GSM-Control Object attributes from Wonderware Application Server environment – it is a simple calendar that controls when defined users will obtain SMS messages from GSM-Control Object.
- **P068m13.pdf** – User Manual (this document)

Installation steps

1. Copy GSM-Control Object files to some folder, e.g. to C:/ GSM-Control Object folder.

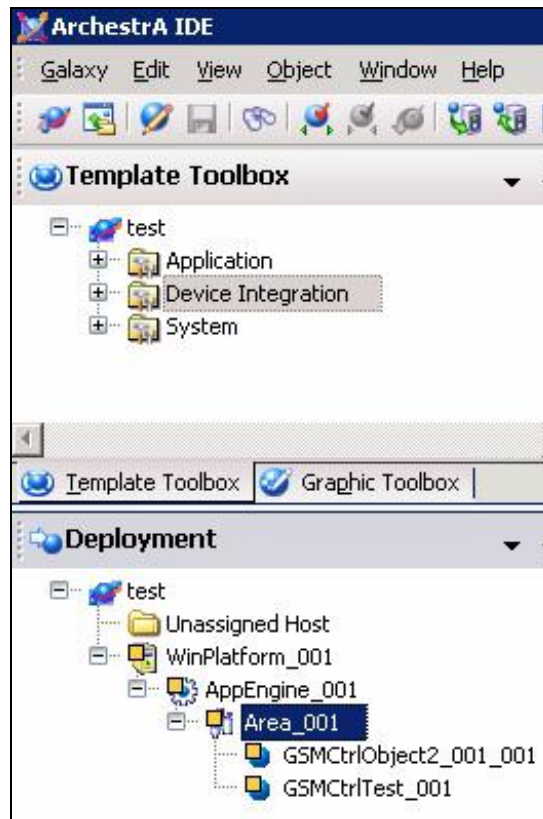
Note: the following explanation covers the GSM-Control Object and the test demo object installation; for GSM-Calendar Object installation refer to “GSM-Calendar Object” section later in this manual.

2. Start the ArchestrA IDE and import the **\$GSMCtrlObject.aaPKG** and **\$GSMCtrlTest.aaPKG** files to a new/existing galaxy (in the further explanation we will assume that a new galaxy **test** is used). For Wonderware Application Server versions earlier as 2012 (version 3.5), import the **GSMCtrlObject.aaPDF** file instead of **\$GSMCtrlObject.aaPKG** – in this case there is no default configuration provided and it should be created manually (by creating new derived template and configuring it). After importing, three new templates are added to Template Toolbox:



Here the **\$GSMCtrlObject2** is GSM-Control Object base template, **\$GSMCtrlObject2_001** is GSM-Control Object derived (child) template and **\$GSMCtrlTest** is test (demo) object template.

3. Create new instances of **\$GSMCtrlObject2_001** and **\$GSMCtrlTest** and locate new instances **GSMCtrlObject2_001_001** and **GSMCtrlTest_001** under **Area_001**:



Now GSM-Control Object and test object are ready for configuration, deployment and run-time use.

4. The GSM-Control Object default configuration is provided for demo purposes only, to serve the explanation in “Getting started quickly with GSM-Control Object” section. The default configuration contains and allows to do the following:

- one User “User1” with phone number “+371777666555”;
- one Alarm message “Alarm1”: the SMS message “alarm!” will be sent to “User1” when test object attribute GSMCtrlTest_001.Alarm value is set to True;
- one pair of Send/Receive messages “SR1”: when GSM-Control Object receives SMS message “?” from “User1”, the response SMS message “Value:{Item1}” will be sent back to “User1”, where {Item1} will be replaced with current value of test object attribute GSMCtrlTest_001.IntegerValue.

Getting and enabling license

To enable the **software key** or **demo license** or licensing under **License Server**, open the GSM-Control Object configuration editor, by double-clicking the instance **GSMCtrlObject2_001_001**, then click on the “License Server” tab:

The screenshot shows the 'License Server' tab of the 'GSMCtrlObject2_001_001' configuration window. The window has several tabs: General, Users, Send/Receive, Alarms, License Server (selected), Object Information, and Script. The 'License Server' tab contains the following fields and controls:

- Product ID:** Text box containing 'PR 006 81'.
- Product Version:** Text box containing '1.2'.
- IP Address:** Empty text box.
- Port:** Text box containing '8001'.
- Customer ID:** Empty text box.
- PC Code:** Text box containing '2496-7475-8b8b-f991'.
- ProductID:** A group box containing two radio buttons:
 - ☒ PR00681 102
 - ☐ PR00682 102
- Software Key:** Text box with an 'Invalid Key' button to its right.
- Demo license text:** Large empty text area.
- Expires:** Empty text box at the bottom.

Software Key

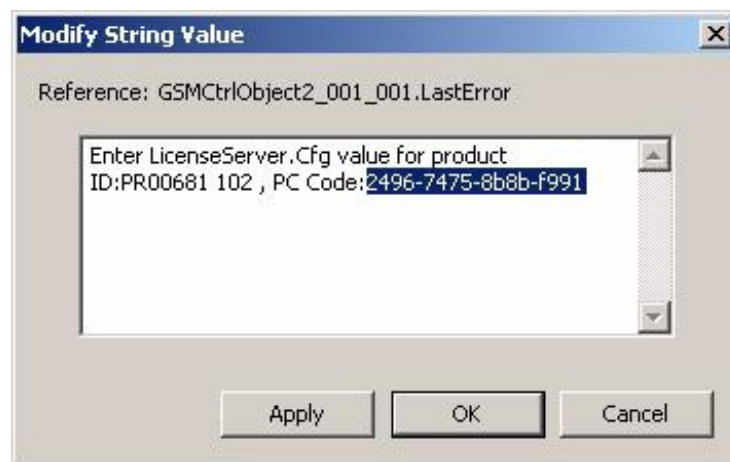
The **software key** should be purchased and currently it is available for two product levels, based on total number of Users configured – maximum 3 Users version (“ProductID” starts with PR00681) and unlimited version (“ProductID” starts with PR00682).

Note: the purchased Software Key permits running and deploying Object on single computer. You should decide on which computer the GSM-Control Object will be used prior of purchase of software key. The editor’s License Server page shows PC Code string from computer where IDE is running. If Object will be deployed on different Platform (computer) then PC Code will be calculated after first deploy of Object. The Product ID will be calculated automatically according to Users count.

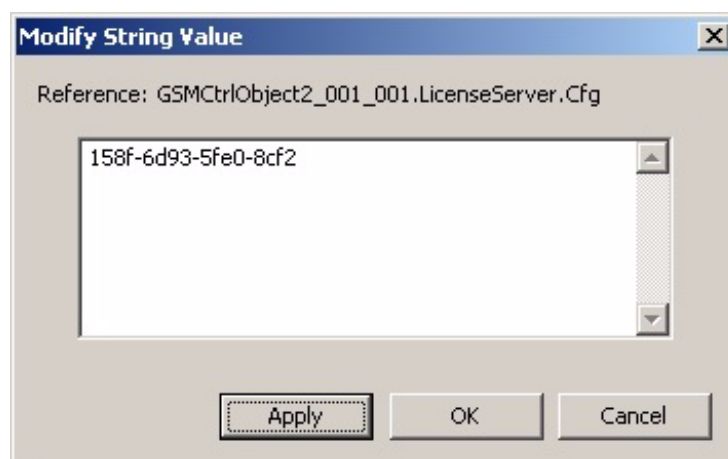
To get and enable the **software key**:

- 1) If GSM-Control Object will be deployed locally then getting and enabling the software key is performed on configuration editor “License Server” tab:

- select the appropriate “ProductID” (e.g. PR00681 xxx or PR00682 xxx);
 - copy/paste the automatically generated “PC Code” to e-mail (or text file or similar) and provide the “ProductID” and “PC Code” strings when ordering the GSM-Control Object product;
 - when GSM-Control Object product is purchased, copy the received “Software Key” to corresponding “Software Key” tab field and check that it is evaluated as “Valid Key” (as well “GSM.: License OK” message should appear in SMC Log Viewer).
- 2) If deploying GSM-Control Object remotely (to other computer) then getting and enabling the software key can be done only after GSM-Control Object was deployed:
- open GSM-Control Object in IDE Object Viewer, select the LastError attribute and open it for modification:



- copy/paste the “ProductID” and “PC Code” to e-mail (or text file or similar) and provide the “ProductID” and “PC Code” strings when ordering the GSM-Control Object product (“ProductID” is based on number of currently configured Users, so in “unlimited” license case at least 4 Users must be configured);
- when GSM-Control Object product is purchased, copy the received “Software Key” to GSM-Control Object “LicenseServer.Cfg” attribute and check if the “GSM.: Valid software license found for PR0068x nnn.” message appears in SMC Log Viewer and GSM-Control Object “LicenseServer.IsLicensed” attribute changes to “true”:

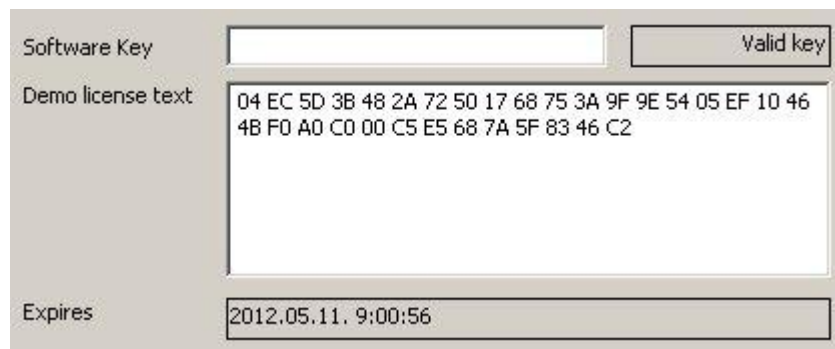


Demo License

The **demo license** is for free and provides an unlimited functionality (up to 10 Users are supported), but it is valid only for a limited time period. After demo license expiration the GSM-Control Object will support only 1 User and will work only 1 hour after GSM-Control Object deployment (in run-time).. The demo license can be obtained by sending request to info@wonderware.fi.

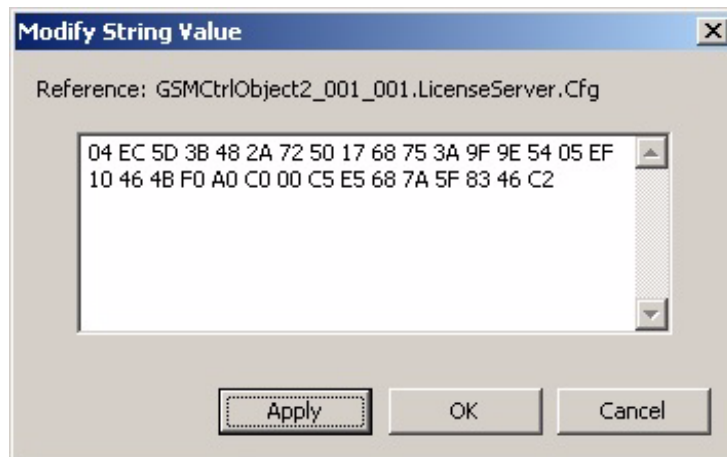
The demo license is a text string containing 64 Hex symbols. When received, paste the demo license text string:

- to “Demo license text” field and check that it is evaluated as “Valid Key” in case GSM-Control Object will be deployed locally:



The screenshot shows a configuration window with three main sections. The top section is labeled 'Software Key' and contains an empty text box with a 'Valid key' button to its right. The middle section is labeled 'Demo license text' and contains a text box with the following 64 hex characters: 04 EC 5D 3B 48 2A 72 50 17 68 75 3A 9F 9E 54 05 EF 10 46 4B F0 A0 C0 00 C5 E5 68 7A 5F 83 46 C2. The bottom section is labeled 'Expires' and contains a text box with the date and time: 2012.05.11. 9:00:56.

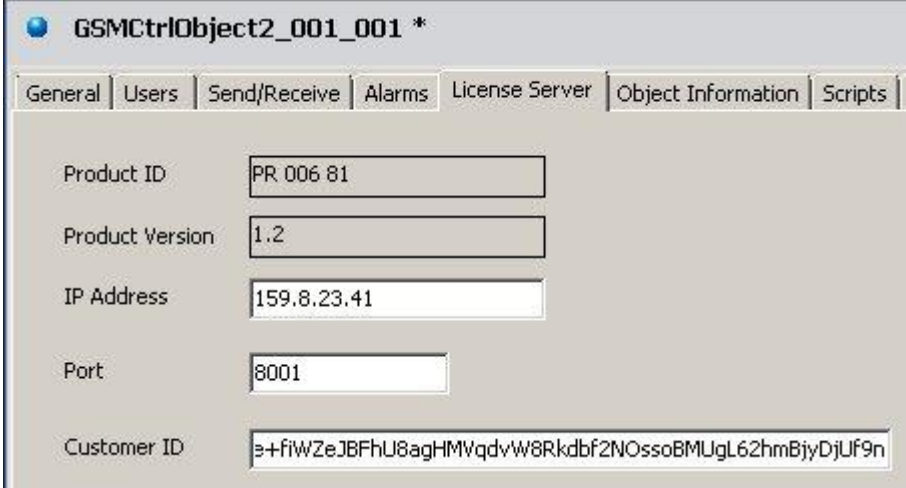
- to GSM-Control Object “LicenseServer.Cfg” attribute (after deploy) and check if message like following appears in SMC Log Viewer: “GSM.: Using demo license! License expires at 5/11/2012 9:00:56 AM” and GSM-Control Object “LicenseServer.IsLicensed” attribute changes to “true” in case GSM-Control Object is deployed remotely:



The screenshot shows a dialog box titled 'Modify String Value'. It has a reference field containing 'GSMCtrlObject2_001_001.LicenseServer.Cfg'. Below the reference field is a text box containing the same 64 hex characters as the previous screenshot: 04 EC 5D 3B 48 2A 72 50 17 68 75 3A 9F 9E 54 05 EF 10 46 4B F0 A0 C0 00 C5 E5 68 7A 5F 83 46 C2. At the bottom of the dialog box are three buttons: 'Apply', 'OK', and 'Cancel'.

Licensing under License Server

To enable the licensing under **License Server** in case GSM-Control Object is delivered as a part of Wonderware Finland **Automation Intelligence Suite** package, the proper “Product ID” and “Product Version” should be selected, the “IP Address” and “Port” of License Server node entered and proper “Customer ID” entered (“Customer ID” is generated automatically by License Server when the License file (.klic file) is obtained – should be copied to GSM-Control Object “Customer ID” field):



The screenshot shows a configuration window titled "GSMCtrlObject2_001_001 *". It has several tabs: General, Users, Send/Receive, Alarms, License Server (selected), Object Information, and Scripts. The License Server tab contains the following fields:

Field	Value
Product ID	PR 006 81
Product Version	1.2
IP Address	159.8.23.41
Port	8001
Customer ID	3+fiWZeJBfH08agHMVqdyW8Rkdbf2NOssoBMUgL62hmBjyDjUf9n

In case the License Server has a proper license for GSM-Control Object, the messages like following will appear in GSM-Control Object node SMC Log Viewer:

“GSM.: Connected to license server”

“GSM.: Found license for product PR 006 81”

“GSM.: Valid license found!”

and GSM-Control Object “LicenseServer.IsLicensed” attribute changes to “true”.

More about licensing under License Server, please see more in separate “Automation Intelligence Suite License Server” guide.

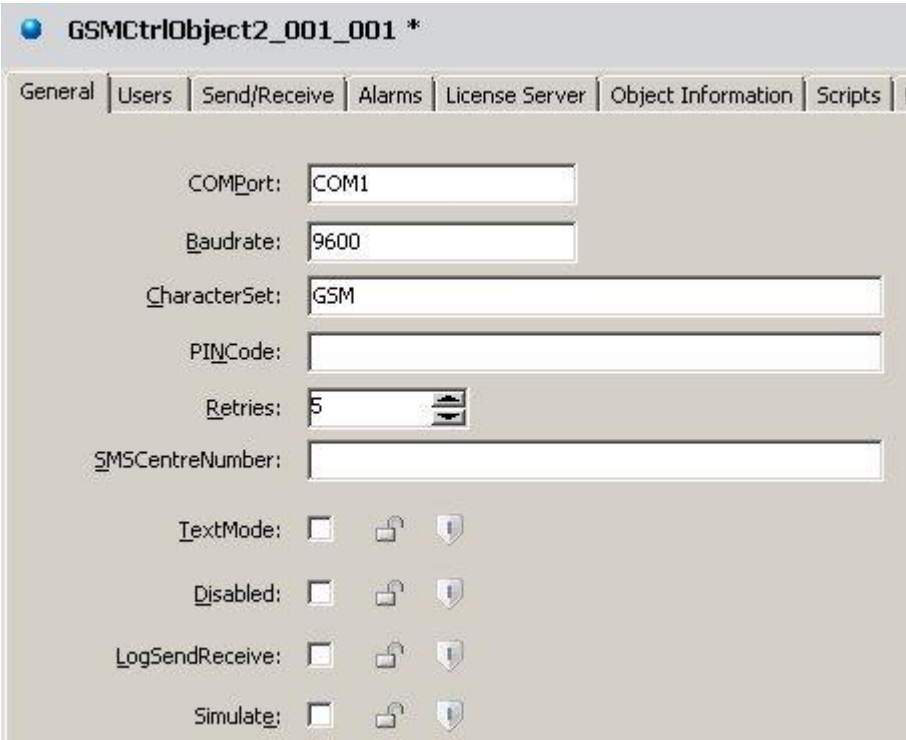
Getting started quickly with GSM-Control Object

The easiest way to start and become familiar with GSM-Control Object is to use the default configuration supplied within GSM-Control Object distribution package.

Default configuration

To examine the GSM-Control Object default configuration, open the GSM-Control Object configuration editor, by double-clicking the instance **GSMCtrlObject2_001_001**. There are four tabs available for GSM-Control Object configuration: "General", "Users", "Send/Receive" and "Alarms". The default configuration is provided for purpose to explain the very basic functions: 1) sending one alarm SMS message and 2) receiving SMS message and sending response SMS message to same phone number. GSM-Control Object default configuration contains the following information:

"General" tab



The screenshot shows the configuration window for the GSM-Control Object instance **GSMCtrlObject2_001_001**. The "General" tab is selected, showing the following settings:

- COMPort:** COM1
- Baudrate:** 9600
- CharacterSet:** GSM
- PINCode:** (empty field)
- Retries:** 5
- SMSCentreNumber:** (empty field)
- TextMode:** ☐ (disabled)
- Disabled:** ☐ (disabled)
- LogSendReceive:** ☐ (disabled)
- Simulate:** ☐ (disabled)

The "General" tab contains the GSM-modem default settings and some common settings. As default the modem is connected to COM port "COM1". baud rate 9600 bps is used, default character set used on GSM-modem is "GSM" and number of "Retries" for sending SMS message is 5.

“Users” tab

GSMCtrlObject2_001_001 *

General **Users** Send/Receive Alarms License Server Object Information

Users

No	Name
1	User1
	...

☐ Users.User1.Disabled

Users.User1.PhoneNumber
+358777666555

Users.User1.Priority
100

The “Users” tab contains one User “User1” with phone number “+358777666555” and with default Priority 100 (the Priority is used only for Alarm messages to specify the order of receiving Alarm SMS in case there are more than one User configured as a receiver).

“Send/Receive” tab

GSMCtrlObject2_001_001

General Users **Send/Receive** Alarms License Object Information Scripts UDAs Extensions

No	Name
1	SR1
	...

SendReceive.SR1.ReceiveMessage
?

SendReceive.SR1.SendMessage
Value:{1}

☐ SendReceive.SR1.Disabled

Users

User
☒ User1

IO Items

No	Name
1	Item1
	...

SendReceive.SR1.Item1.Reference
GSMCtrlTest_001.IntegerValue

The “Send/Receive” tab contains one pair of Send/Receive message “SR1” linked with User “User1” and consisting of Receive message “?” and Send message “Value:{1}”. The Receive message contains the index (taken from “No” column at “IO Items” list view) of embedded IO Item “Item1” associated with test object attribute GSMCtrlTest_001. IntegerValue.

“Alarms” tab

The screenshot shows the configuration window for GSMCtrlObject2_001_001, specifically the 'Alarms' tab. The window has several tabs: General, Users, Send/Receive, Alarms (selected), License, Object Information, and Script. On the left, there is a table with columns 'No' and 'Name'. The first row shows '1' and 'Alarm1'. Below this, there are several empty rows and a scroll bar. On the right, there are several configuration fields for 'Alarms.Alarm1':

- AlarmMessage:** alarm!
- AckMessage:** (empty field)
- AlarmItem.InputSource:** GSMCtrlTest_001.Alarm
- AckItem.OutputDest:** ---
- Timeout:** 3
- SendToAll:** ☐
- VoiceSource:** (empty field)
- Disabled:** ☐

At the bottom right, there is a 'Users' section with a table:

User	
☒ User1	

The “Alarms” tab contains one Alarm message “Alarm1” linked with User “User1”. The text of Alarm message is “alarm!” and the input source of alarm state is associated with test object attribute GSMCtrlTest_001. Alarm. The time-out for sending alarm message is 3 seconds.

The following very basic can be done by using the default configuration:

- 1) If test object attribute GSMCtrlTest_001.Alarm value is set to True then alarm SMS message “alarm!” will be sent to default “User1” (default phone number +358777666555).

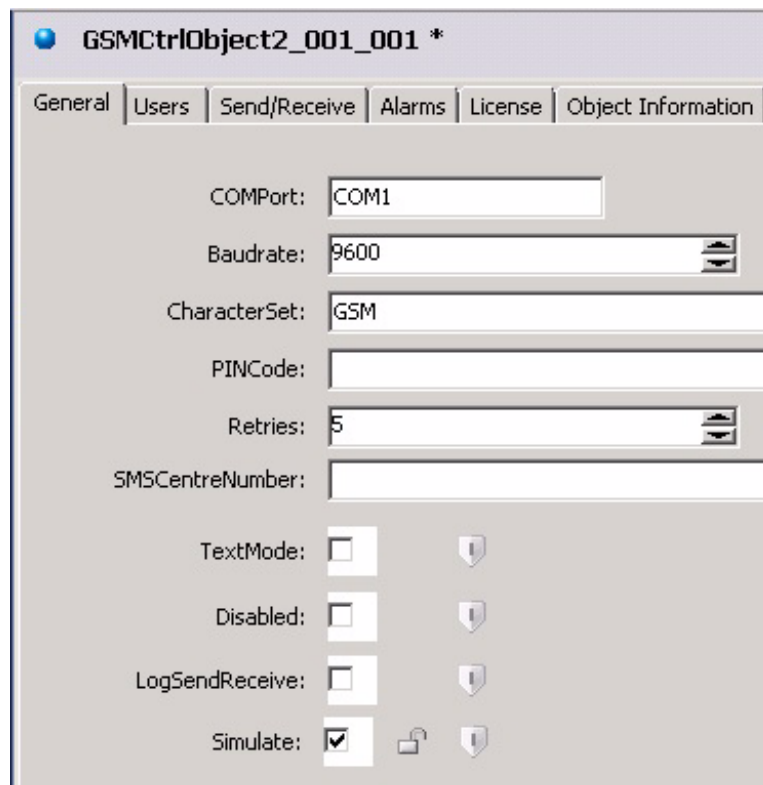
2) If GSM-Control Object receives SMS message “?” from default “User1” (default phone number +358777666555) then response SMS message “Value:{1}” will be sent back to “User1”, where {1} will be replaced with current value of test object attribute GSMCtrlTest_001. IntegerValue (for example: “Value:13”).

How to get started quickly in simulation mode (without modem connected) and with modem connected is explained below.

Getting started in simulation mode (without GSM-modem connected)

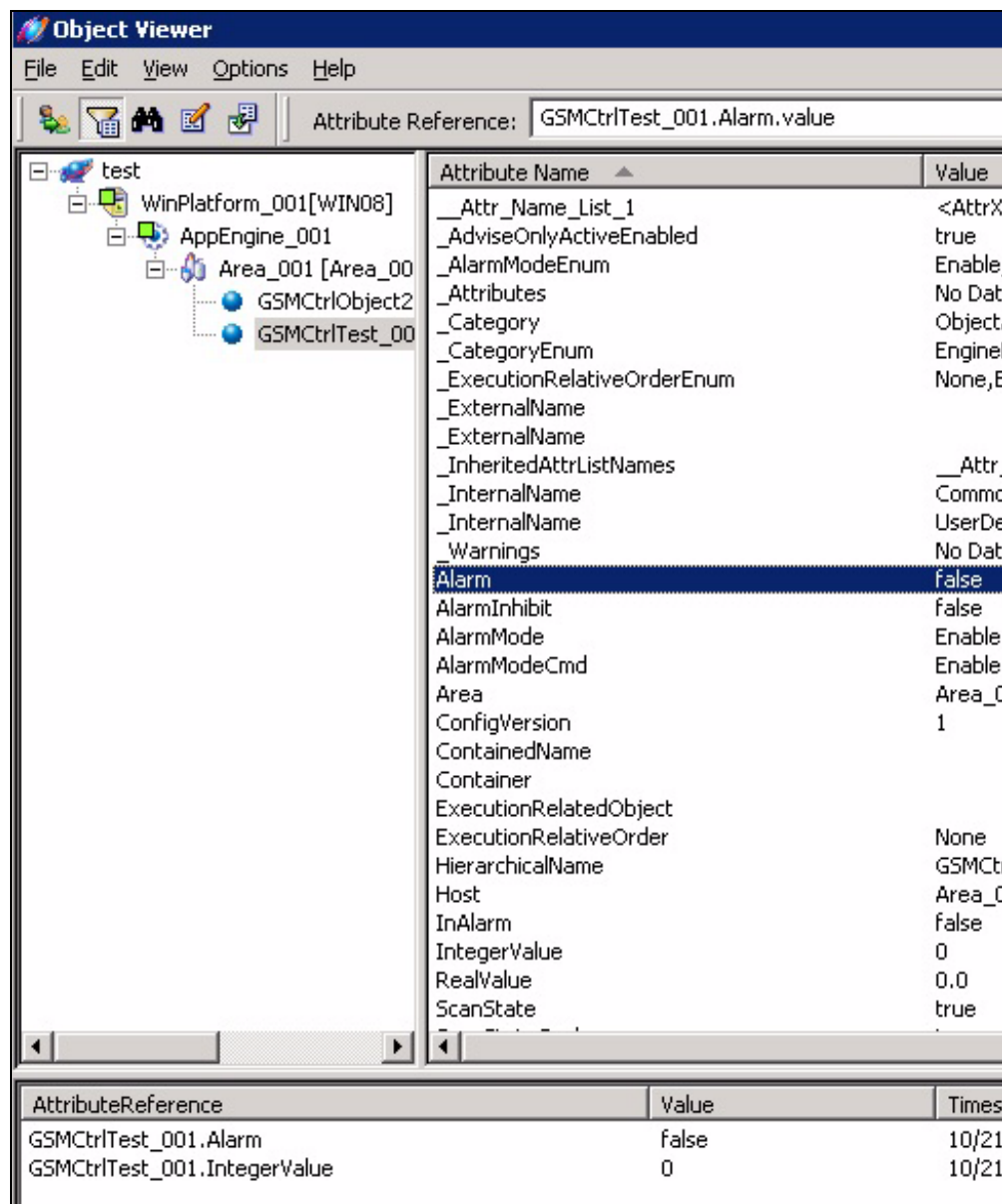
The simulation mode is useful to examine the GSM-Control Object functionality without GSM-modem connected and without real sending/receiving of SMS messages via GSM-modem. In simulation mode the texts of sent and received SMS messages are logged to SMC Log Viewer. The following sequence can be used to start quickly with GSM-Control in simulation mode and using the default configuration:

- 1 Undeploy (if not yet undeployed) the **GSMCtrlObject2_001_001** instance and in “General” tab select (check) the “Simulate” check-box in “General” tab, save/close the GSM-Control Object configuration:

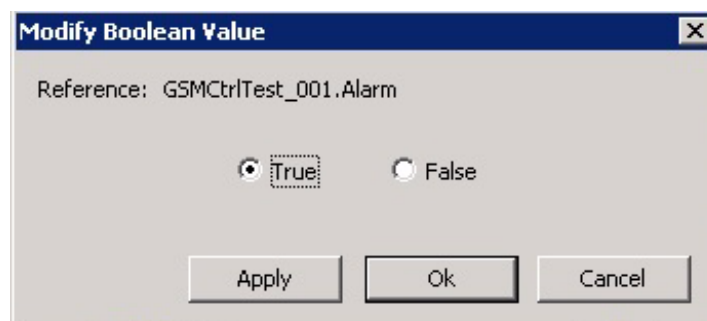


- 2 Deploy **GSMCtrlObject2_001_001** and **GSMCtrlTest_001** instances.

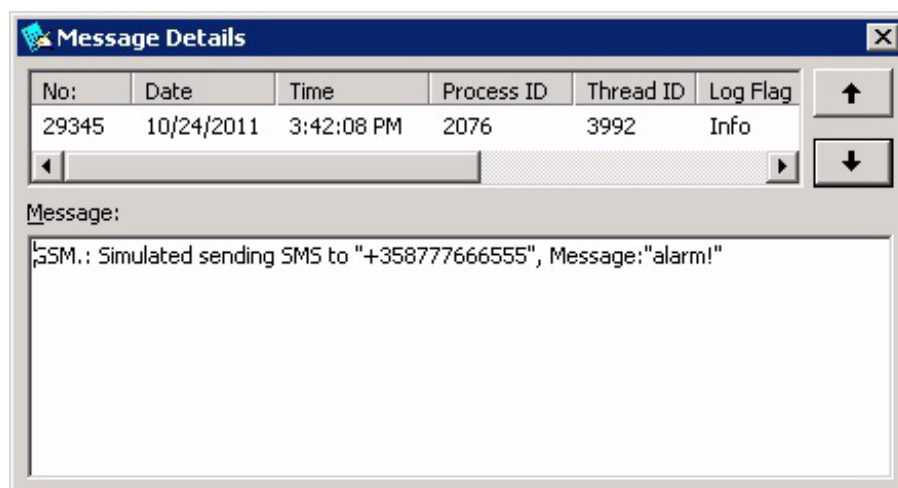
- 3 Open the **GSMCtrlTest_001** in IDE Object Viewer and select/add to watch the **GSMCtrlTest_001.Alarm** and **GSMCtrlTest_001.IntegerValue** attributes:



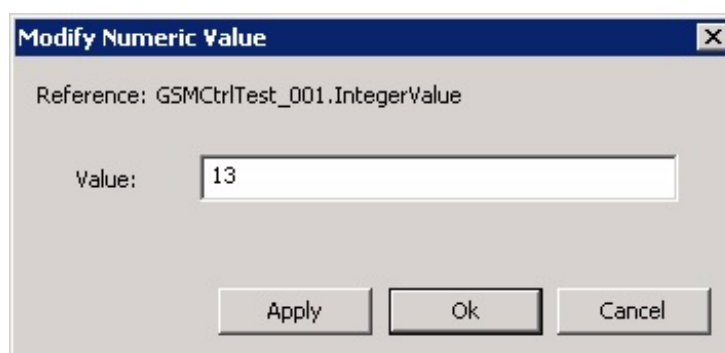
- 4 Modify the **GSMCtrlTest_001.Alarm** value to True:



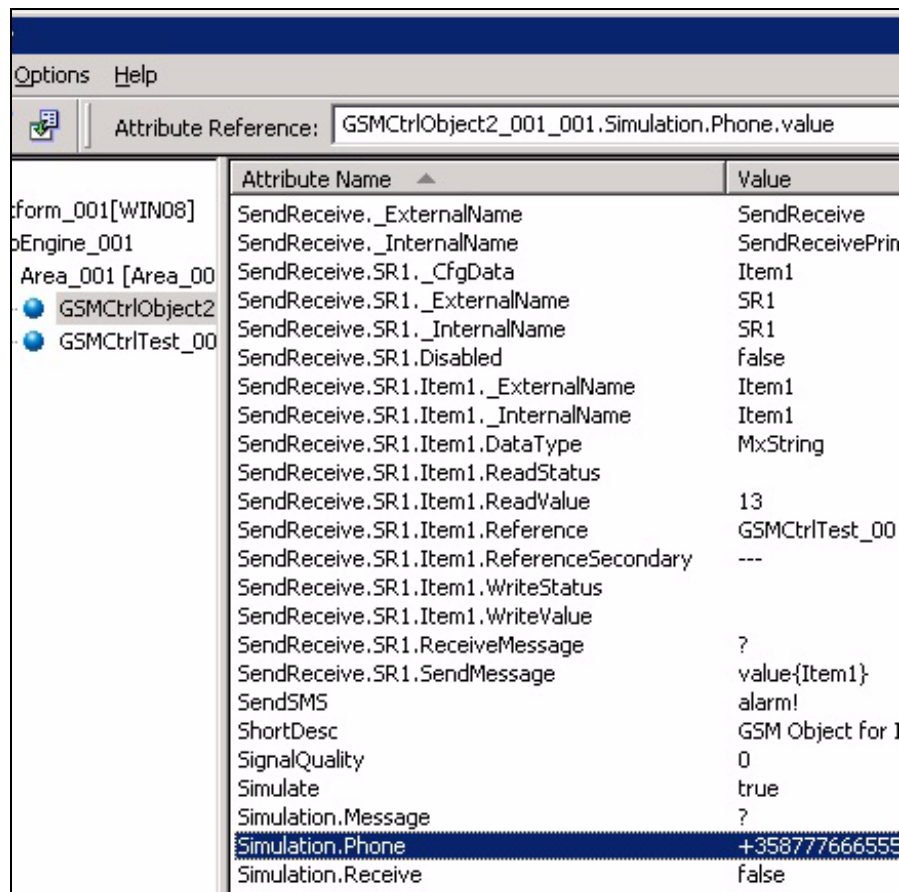
- 5 Open (if not yet opened) the SMC Log Viewer – the message about the simulated message "alarm!" sent to +35877766655 is logged:



- 6 In Object Viewer, modify the **GSMCtrlTest_001.IntegerValue** value to 13:

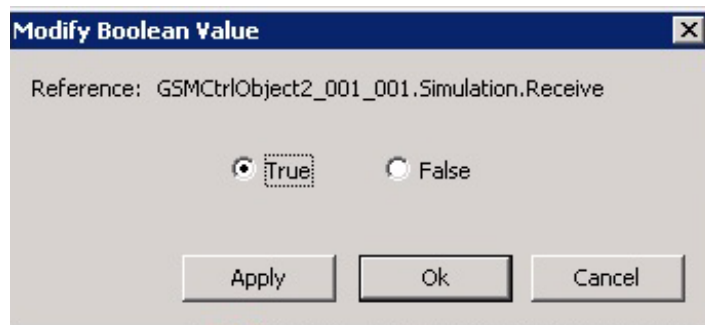


- 7 Modify the values of special Simulation attributes:

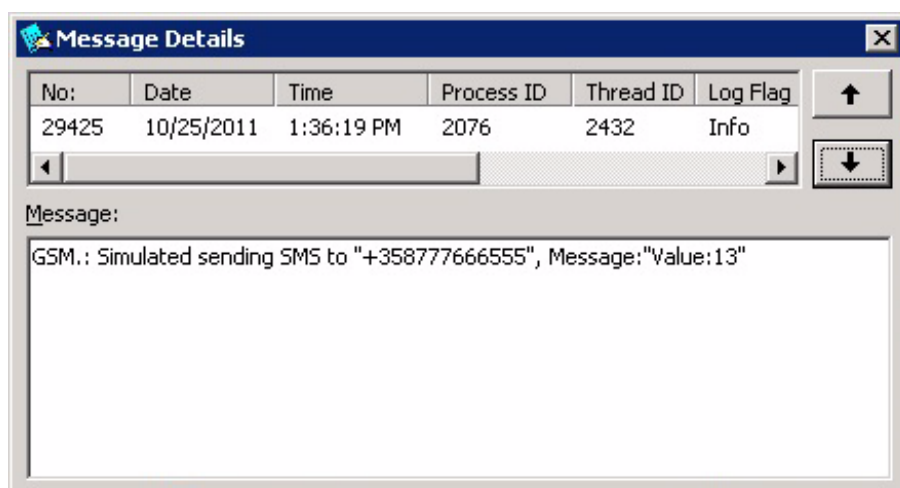
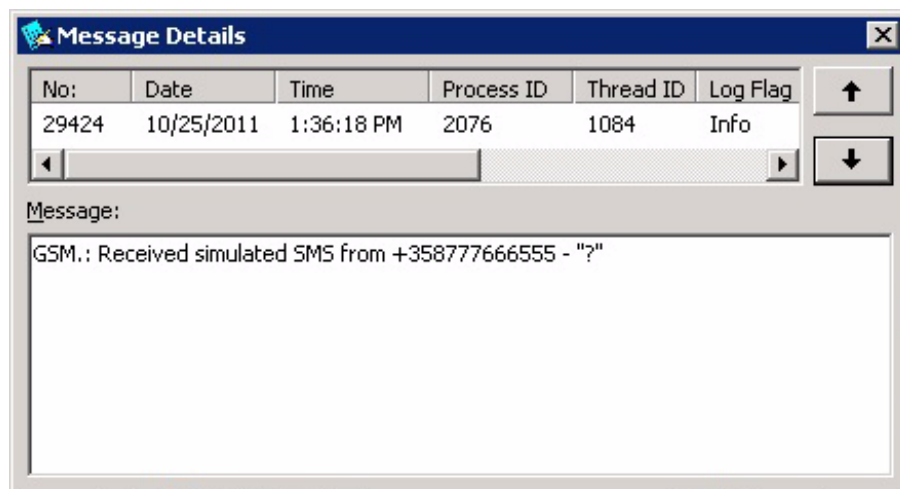


Modify the **GSMCtrlObject2_001_001.Simulation.Message** string to "?" and **GSMCtrlObject2_001_001.Simulation.Phone** string to "+358777666555".

- 8 Modify the **GSMCtrlObject2_001_001.Simulation.Receive** value to True, thus initiating the receiving of simulated SMS message:



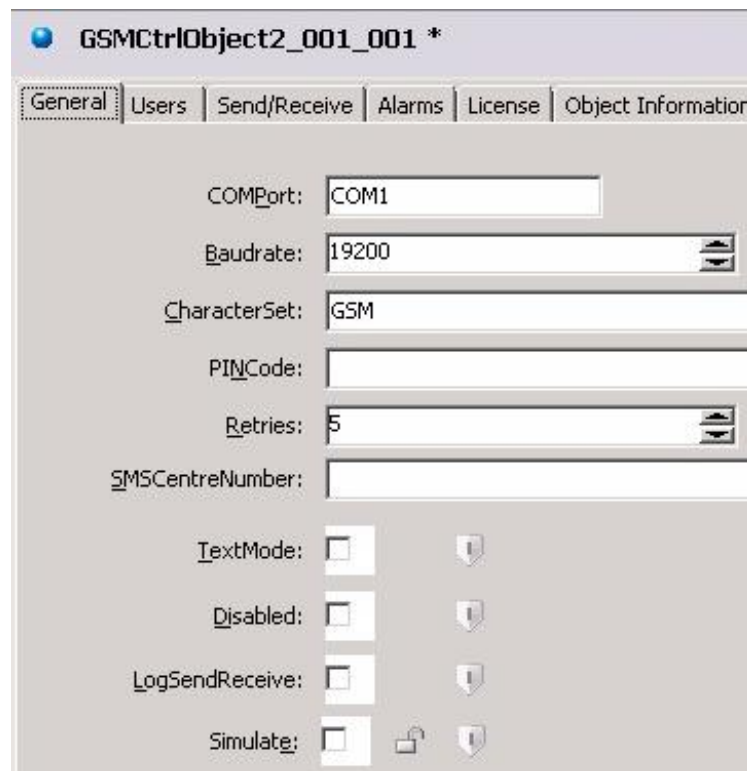
- 9 The following messages now are logged to SMC Log Viewer, showing: after simulated SMS message "?" was received from phone number +358777666555, the response SMS message "Value:13" was sent back to same phone number:



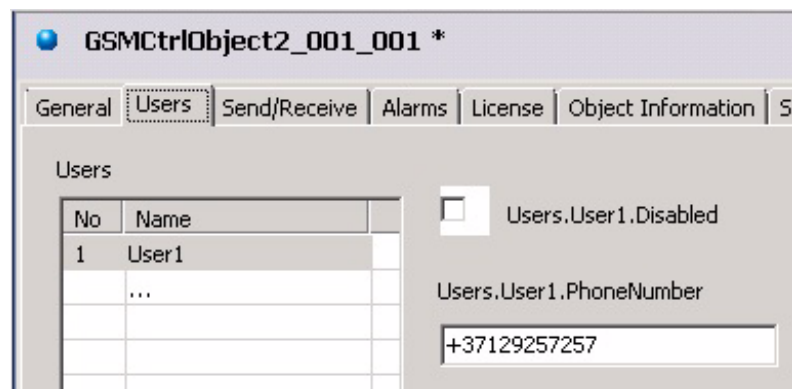
Getting started with GSM-modem connected

The following explains the sequence how to start quickly with GSM-Control Object by using GSM-modem and sending/receiving of SMS messages by mobile phone:

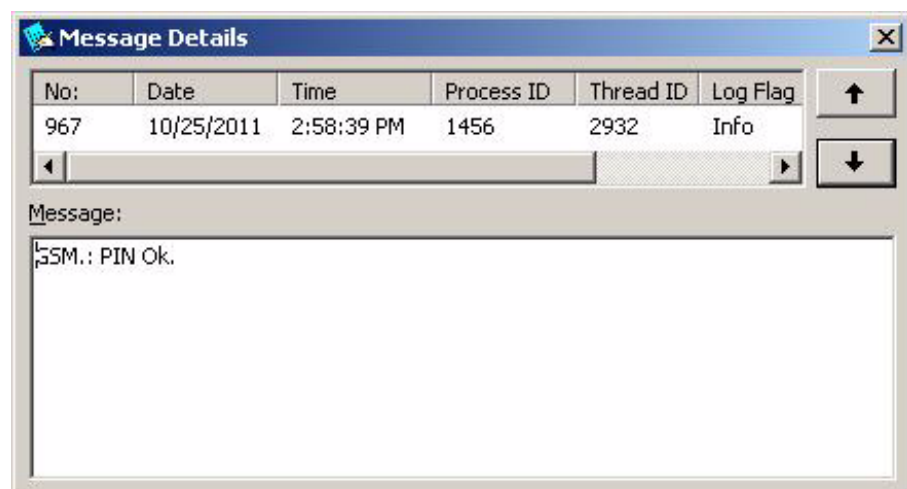
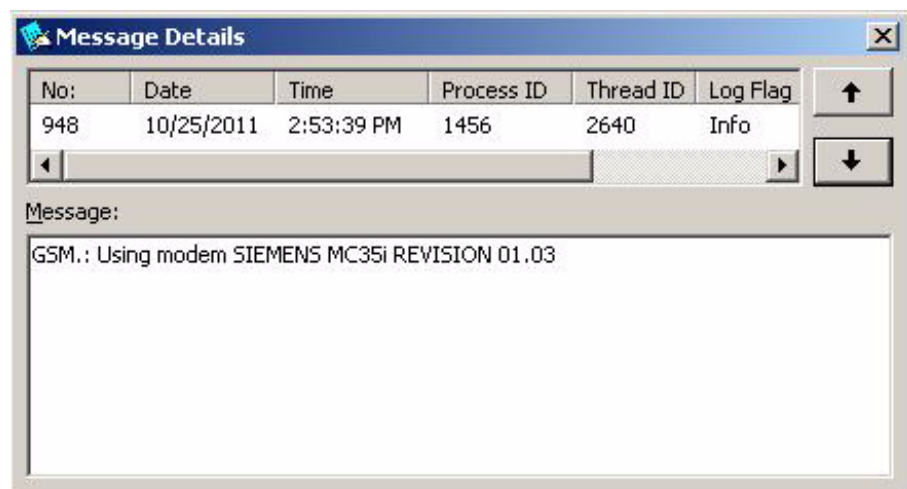
- 1 Find out and remember the PIN code (if enabled) and phone number of the SIM card to be used with GSM-modem.
- 2 Insert the SIM card into the GSM-modem. Connect GSM-modem to computer serial port, e.g. to COM1 and power-up the GSM-modem. It is assumed the GSM-modem has default serial port configuration: baud rate 19200, 8 data bits, 1 stop bit, no parity, no flow control. If GSM-modem settings differs from default then it is necessary to restore the default settings - it can be done e.g. by Windows HyperTerminal program by issuing AT&F ("Set all current parameters to manufacturer defaults") command.
- 3 Undeploy (if not yet undeployed) the **GSMCtrlObject2_001_001** instance and in "General" tab change the "Baudrate" setting to 19200 (or to other in case the modem current baudrate is different) and enter the PIN code (in case it is enabled on SIM card):



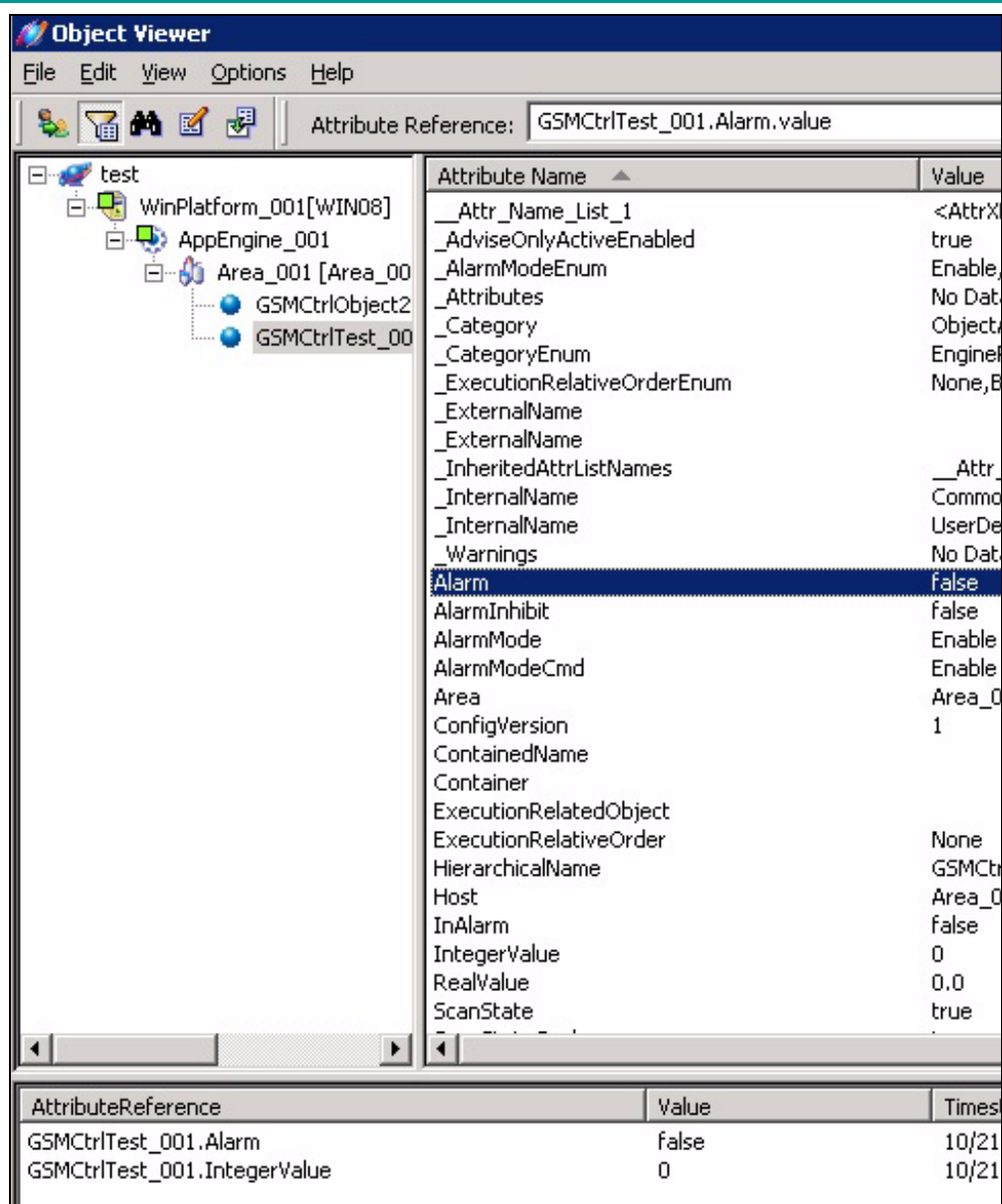
- 4 In "Users" tab change the "User1" phone number to your mobile phone number used to send/receive SMS messages to/from GSM-Control Object (in further explanation the phone number "+37129257257" will be used):



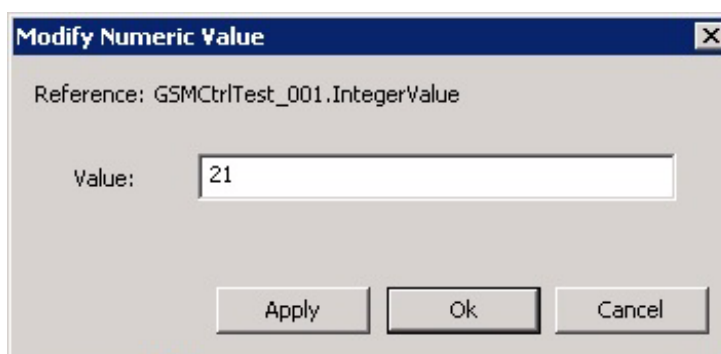
- 5 Save/close the GSM-Control Object configuration.
- 6 Deploy **GSMCtrlObject2_001_001** and **GSMCtrlTest_001** instances.
- 7 Open (if not yet opened) the SMC Log Viewer and check if messages like following are logged, indicating the GSM-modem and PIN code (in case enabled on SIM card) were detected and GSM-modem is ready:



- 8 Open the **GSMCtrlTest_001** in IDE Object Viewer and select/add to watch the **GSMCtrlTest_001.Alarm** and **GSMCtrlTest_001.IntegerValue** attributes:



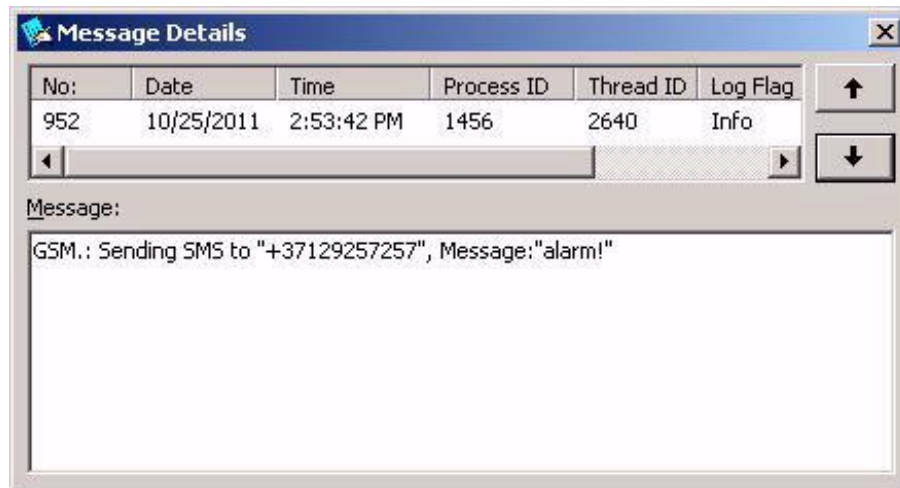
9 Modify the **GSMCtrlTest_001.IntegerValue** value to 21:



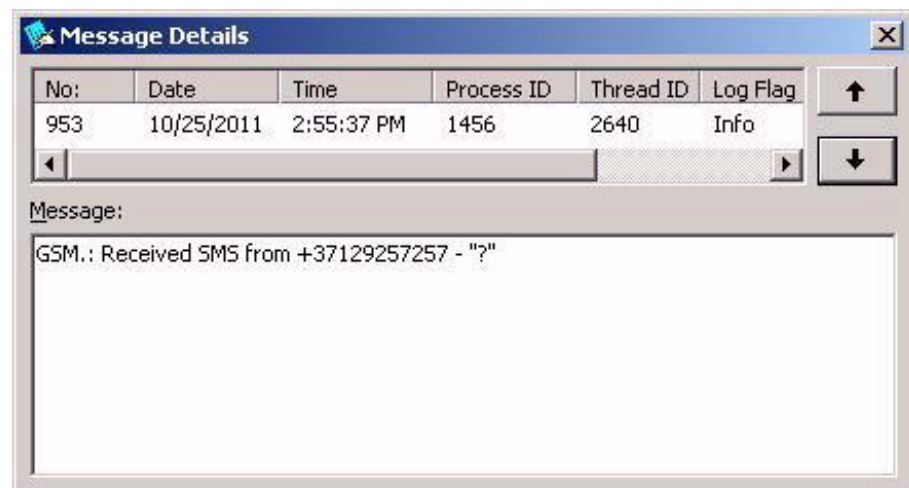
10 Modify the **GSMCtrlTest_001.Alarm** value to True:



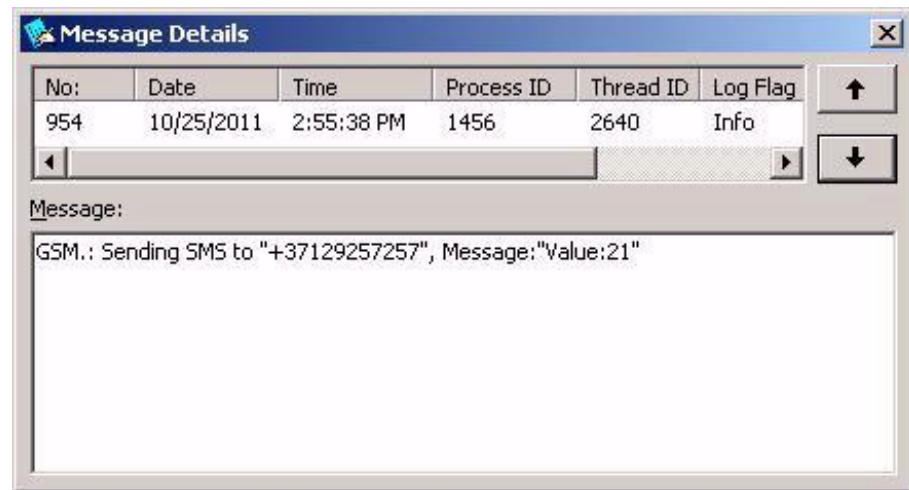
- 11 The SMS message "alarm!" should be sent to and received by mobile phone; in SMC Log Viewer the corresponding message is logged:



- 12 From your mobile phone, respond to the sender with SMS message "?". In SMC Log Viewer the message indicating the receiving of this SMS message should be logged:



- 13 Upon receiving the SMS message "?", GSM-Control Object will respond to sender with SMS message "Value:21" containing the current value of **GSMCtrlTest_001.IntegerValue** attribute; in SMC Log Viewer the corresponding message is logged::

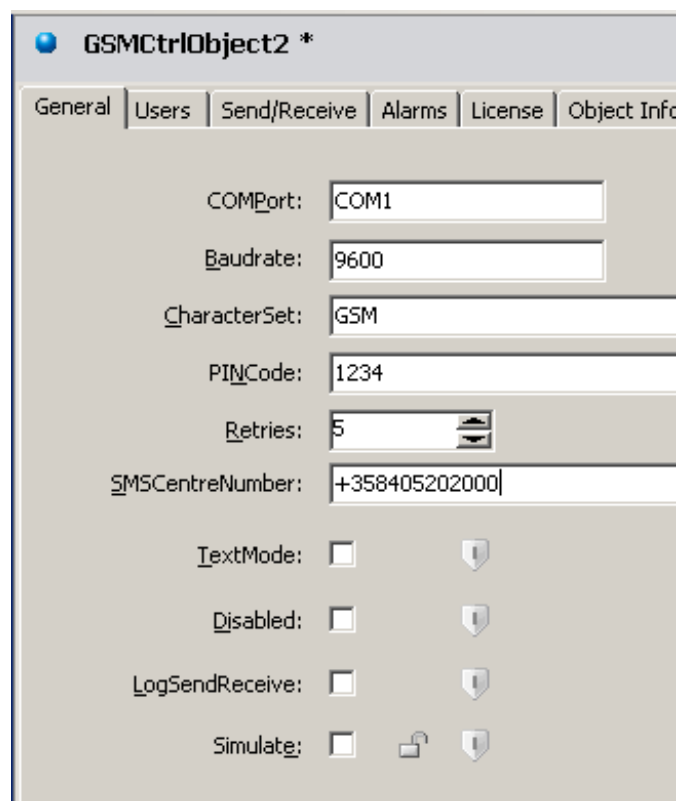


GSM-Control Object configuration in details

This section explains the GSM-Control Object configuration in details.

General tab

The “General” tab contains the GSM-modem default settings and some common settings:



The following can be entered in “General” tab:

COMPort: The name of serial communication device (COM port) where GSM-modem is connected. Typical names are COM1, COM2...

Baudrate: Serial communication baud rate in bits per second used to communicate with modem. Other communication parameters are fixed - parity: none, stop bits: 1, no flow control.

CharacterSet: The name of modem character set used. The character set is used to configure modem with AT+CSCS command at GSM-Control Object runtime start-up. Default character set is "GSM".

PINCode: Place to enter the modem SIM card PIN code, in case modem is secured with PIN code. Empty field means PIN code is not in use.

Retries: Number of retries for execution of AT command, in case of reply time-out from modem.

SMSCentreNumber: Here the optional SMS center number can be entered (in case the SMS center number stored on modem SIM card is not suitable). If this field is empty (default and recommended) then GSMControl Objec2 uses preconfigured (default) SMS center number stored in modem SIM card.

TextMode: If checked, AT+CMGF=1 command (enabling SMS text mode) is used to configure the modem. As default, SMS PDU mode is used and AT+CMGF=0 command is executed to use modem in PDU mode.

Disabled: Disables the sending of SMS messages through the modem.

LogSendReceive: Enables logging of modem serial communications to Wonderware ArcestraA Logger.

Simulate: Enables working in simulation mode, when SMS are not sent/received, but instead logged in ArcestraA Logger. The simulation of receiving SMS message can be done by using object runtime attributes Simulation.Message, Simulation.Phone and Simulation.Receive – see "Description of GSM-Control Object attributes" section later in this manual.

Users tab

The “Users” tab contains information about all configured Users:

No	Name
1	User1
...	...

☐ Users.User1.Disabled
 Users.User1.PhoneNumber: +358777666555
 Users.User1.Priority: 100

The following can be entered in “Users” tab:

Users.No: User index. User with smaller index number have higher priority if more than one User is selected for sending alarm message.

Users.Name: The name of virtual primitive containing the User configuration attributes.

Users.UserName.Disabled: Controls the User enabled/disabled state. By default User is enabled. Disabled User means that SMS received from this User are not handled and no alarms are sent to this User.

Users.UserName.PhoneNumber: Contains User phone number in international format, used in AT+CMGS and ATD commands.

Users.UserName.Priority: Contains the Priority of User, used only for Alarm messages to specify the order of receiving Alarm SMS in case there are more than one User configured as a receiver and Ack message is configured. The Users with smaller Priority value will receive Alarm SMS before Users with greater Priority value. For example, if for Alarm message there are three Users configured (User1 has Priority = 100, User2 has Priority = 50, User3 has Priority = 101) and

Alarms.AlarmName.AckMessage is not empty (response Ack SMS is expected to be received from some of Users), at first the Alarm SMS-message will be sent to User2, then in case after **Alarms.AlarmName.Timeout** there is no response – to User1, and then – to User3.

Send/Receive tab

The “Send/Receive” tab contains information about all configured Send/Receive message pairs:

GSMCtrlObject2 *

General Users **Send/Receive** Alarms License Object Information Scripts UDAs Extensions Graphics

No	Name
1	SR1
...	...

SendReceive.SR1.ReceiveMessage
?

SendReceive.SR1.SendMessage
Value:{1}

☐ SendReceive.SR1.Disabled

Users

User
☒ User1

IO Items

No	Name
1	Item1
...	...

SendReceive.SR1.Item1.Reference
GSMCtrlTest_001.IntegerValue

The following can be entered in “Send/Receive” tab:

No: The index of configured Send/Receive message pair. The Send/Receive with smaller index is processed first. The received SMS is handled/processed by looking at all configured Send/Receive pairs.

Name: Name of virtual primitive containing the configuration attributes for SendReceive messages pair. In further explanation this name is referred to as “*SRName*”.

SendReceive.SRName.ReceiveMessage: Contains the Receive message (SMS message received by GSM-Control) text and optionally the values for configured IO items. Each configured Receive message content should be unique enough to allow determine it and perform the right processing. The placeholders of IO items values can be entered as **IO Item Index** or **IO Item Name** and enclosed in figure brackets {}.

SendReceive.SRName.SendMessage: Contains the Send message (SMS message sent from GSM-Control) text and optionally the values for configured IO items. The Send message (if configured – means Send message can be empty) will be send back to ReceiveMessage sender in case the received SMS message matches with some of configured ReceiveMessages. The placeholders of IO items values can be entered as **IO Item Index** or **IO Item Name** and enclosed in figure brackets {}.

SendReceive.SRName.Disabled: The SendReceive messages pair can be disabled from processing by checking this check box. Later at run-time it can be enabled by using SendReceive.SRName.Disabled runtime attribute – see “Description of GSM-Control Object attributes” section later in this manual.

Users: Here can be selected the Users linked with this SendReceive messages pair.

IO Items.No: Indexes of configured IO Items, available for ReceiveMessage or SendMessage.

IO Items.Name: Names of configured IO Items. Each configured IO Item creates separate virtual primitive, containing IO Item configuration attributes.

SendReceive.SRName.ItemName.Reference: contains the Galaxy attribute name, which value will be read from Galaxy (and sent from GSM-Control in SendMessage) or written to Galaxy (as received by GSM-Control in ReceiveMessage).

Alarms tab

The “Alarms” tab contains information about all configured Alarm messages:

GSMCtrlObject2 *

General | Users | Send/Receive | **Alarms** | License | Object Information | Scripts | UDAs | Extensions

No	Name
1	Alarm1
	...

Alarms.Alarm1.AlarmMessage
alarm!

Alarms.Alarm1.AckMessage
[Empty]

Alarms.Alarm1.AlarmItem.InputSource
GSMCtrlTest_001.Alarm

Alarms.Alarm1.AckItem.OutputDest

Alarms.Alarm.Timeout:
3 seconds

☐ Alarms.Alarm.SendToAll:

Alarms.Alarm1.VoiceSource
[Empty]

☐ Alarms.Alarm1.Disabled

Users

User
☒ User1

IO Items

No	Name
	...

Item
[Empty]

The following can be entered in “Alarms” tab:

No: The indexes of all configured Alarms. GSM Control Object each scan period sequentially is checking all configured (and enabled) alarms for presence of alarm True state. The execution sequence is from alarm with first index, to the last. All received SMS messages are passed to each (enabled) alarm, in the order of growing indexes - in purpose to check if there is matching with configured AckMessage.

Name: Name of virtual primitive containing alarm configuration attributes. . In further explanation this name is referred to as “AlarmName”.

Alarms.AlarmName.AlarmMessage: Contains the Alarm message (to be sent from GSM-Control) text and optionally the values for configured IO items. The placeholders of IO items values can be entered as **IO Item Index** or **IO Item Name** and enclosed in figure brackets {}.

Alarms.AlarmName.AckMessage: Contains exact characters, which should be sent back to GSM-Control to acknowledge the alarm. IO Items are not supported in AckMessage.

Alarms.AlarmName.AlarmItem.InputSource: Contains the Galaxy Boolean attribute name, used to provide the alarm state. When this attribute value changes to True, the corresponding configured Alarm is switched into alarm state and GSM-Control Object will send Alarm SMS message.

Alarms.AlarmName.AckItem.OutputDest: Contains the Galaxy Boolean attribute name, used to provide alarm ack state and to get the True value, in case GSM Control Object receives the AckMessage. If AckItem referred attribute already has True value, then alarm is considered as acknowledged and Alarm SMS message is not sent.

Alarms.AlarmName.Timeout: Contains the time-out in seconds for waiting for AckMessage before sending the alarm message to next User. This time-out is not applicable in case the **Alarms.AlarmName.AckMessage** field is empty (no response Ack message is expected).

Alarms.AlarmName.SendToAll: If checked - all selected Users will receive the Alarm SMS message, regardless of whether or not the

Alarms.AlarmName.AckMessage field is empty or filled: in case

Alarms.AlarmName.AckMessage field is not empty then response Ack message is expected from any of Users linked with this Alarm message.

Alarms.AlarmName.VoiceSource: Here can be entered the optional path to sound (WAV) file, which will be played (after the SMS sending) after a voice call command “ATD{phone number};” modem command successful execution. If this entry is empty then ATD command (voice call) is not executed. This alarm voice call option requires computer sound card audio output line is connected to modem audio input line (there is separate audio connector socket on the modem).

Alarms.AlarmName.Disabled: The Alarm message can be disabled from processing by checking this check box. Later at run-time it can be enabled by using **Alarms.AlarmName.Disabled** runtime attribute – see “Description of GSM-Control Object attributes” section later in this manual.

Users: Here can be selected the Users linked with this Alarm message.

IO Items.No: Indexes of configured IO Items, available for inclusion in AlarmMessage.

IO Items.Name: Names of configured IO Items, available for inclusion in AlarmMessage.

IO Items.ItemName.InputSource: contains the Galaxy attribute name, which value will be read from Galaxy and sent from GSM-Control in Alarm SMS message.

License Server tab

The “License Server” tab is used to enable the **software key** or **demo license** or licensing under **License Server**:

The screenshot shows the 'License Server' tab of the 'GSMCtrlObject2_001_001 *' dialog. The tab is selected among 'General', 'Users', 'Send/Receive', 'Alarms', 'License Server', 'Object Information', and 'Scripts'. The form contains the following fields and controls:

- Product ID:** Text box containing 'PR 006 81'.
- Product Version:** Text box containing '1.2'.
- IP Address:** Empty text box.
- Port:** Text box containing '8001'.
- Customer ID:** Empty text box.
- PC Code:** Text box containing '2496-7475-8b8b-f991'.
- ProductID:** A group box containing two radio buttons: 'PR00681 102' (selected) and 'PR00682 102'.
- Software Key:** Text box containing '158f-6d93-5fe0-8cf2'. To its right is a 'Valid Key' button.
- Demo license text:** A large empty text area.
- Expires:** Empty text box at the bottom.

For details how to enable the software key or demo license or licensing under **License Server**, please refer to “Getting and enabling license” section (pages 7... 10 upper in this manual).

Description of GSM-Control Object attributes

This section contains the information about GSM-Control Object attributes.

General Attributes

Attribute	Data Type	Access Type	Description
Baudrate	Integer	R/W	Defines the COM port baud rate
CharacterSet	String	R/W	Some of character sets supported by your modem.
COMPort	String	R/W	Defines the COM port name.
Disabled	Boolean	R/W	If disabled then SMS messages will not be sent.
LastError	String	R	Contains the string describing error or empty string in case recovered from errors.
LicenseServer.Cfg	String	R/W	The string where GSM-Control Object current licensing information is displayed and can be entered: Software Key (string like "158f-6d93-5fe0-8cf2"), Demo License (string like "04 EC 5D 3B 48 2A 72 50 17 68 75 3A 9F 9E 54 05 EF 10 46 4B F0 A0 C0 00 C5 E5 68 7A 5F 83 46 C2") or License Server licensing data (string containing License Server node IP Address and Port number and "CustomerID" generated by License Server)
LicenseServer.CustomerID	String	R/W	"CustomerID" generated by License Server
LicenseServer.IP	String	R/W	License Server node IP Address
LicenseServer.IsLicensed	Boolean	R	Current status of GSM-Control Object licensing: true – licensed, false – not licensed
LicenseServer.Port	Integer	R/W	License Server node Port number
LogSendReceive	Boolean	R/W	Enables/disables logging of modem serial communications to Wonderware ArchestraA Logger.
Modem.CharacterSets	String array	R	Contains the array of modem's supported character sets.
PINCode	String	C	Modem PIN Code field.

PINFailed	Boolean	R	Indicates the invalid PIN code.
ReceivePhone	String	R	Contains the phone number from last received SMS message.
ReceiveSMS	String	R	Contains the last received SMS message text.
Retries	Integer	R/W	Number of retries for execution of AT command, in case of reply time-out from modem
SendPhone	String	R/Set	Contains the last sent SMS message phone number. Can be used to set phone numbers for sending custom (not pre-configured) messages.
SendSMS	String	R/Set	Contains the text of last sent SMS message. Writing new value into this attribute will initiate the sending of custom (not pre-configured) SMS message to the phone number currently placed in SendPhone attribute.
SignalQuality	Integer	R	Indicates the modem GSM signal strength.
Simulate	Boolean	R/W	Controls the simulation mode. In simulation mode SMS are not sent, but instead logged into Wonderware ArchestrA. In simulation mode the processing of simulated ReceiveMessages is enabled.
Simulation.Message	String	R/W	Simulated SMS text.
Simulation.Phone	String	R/W	Simulated SMS phone number.
Simulation.Receive	Boolean	R/W	Initiates the receiving of simulated SMS message. The False state – indicates finishing of simulated SMS receiving.
SMSCentreNumber	String	W	The SMS Centre number can be configured or entered in Runtime.
Statistics.ReceivedSMS	Integer	R	Indicates the number of received SMS messages.
Statistics.RejectedSMS	Integer	R	Indicates the number of received invalid SMS messages.
Statistics.SentSMS	Integer	R	Indicates the number of sent SMS messages.
Status	Boolean	R	Indicates the communication status with

			modem. True – good communications, False – failed to communicate.
TextMode	Boolean	R/W	Controls the modem SMS sending/receiving mode – PDU or text mode. The PDU mode is preferable. Text mode can be used for debug purposes.

User Attributes

Attribute	Data Type	Access Type	Description
Users.UserName.Disabled	Boolean	R/W	Controls the User availability for handling SMS messages. If True then no SMS messages are handled from this phone number nor sent to this phone number.
Users.UserName.PhoneNumber	String	R/W	Contains User phone number in international format. The empty string means any phone number is accepted.
Users.UserName.Priority	Integer	R/W	The Priority of User, used only for Alarm messages to specify the order of receiving Alarm SMS in case there are more than one User configured as a receiver and Ack message is configured.

Send/Receive Message Attributes

Attribute	Data Type	Access Type	Description
SendReceive.SRName.ReceiveMessage	String	R/W	Specifies the Receive SMS message pattern in format string[{item}string{item}...]. If such pattern is received then data is extracted from SMS message text and written to external IAS attributes.

SendReceive.SRName.SendMessage	String	R/W	Specifies the Send SMS message pattern in format [string][{item}[string]... The SMS message will be sent in response to received SMS message which matches with ReceiveMessage pattern. The {item} placeholders will be replaced with external IAS attribute values.
SendReceive.SRName.Disabled	Boolean	R/W	Controls Enable – Disable state of this Send/Receive pair.
SendReceive.SRName.Item.Reference	String	R/W	Contains external data item attribute name.

Alarm Message Attributes

Attribute	Data Type	Access Type	Description
Alarms.AlarmName.AlarmMessage	String	R/W	Contains the Alarm message (to be sent from GSM-Control) text and optionally the values for configured IO items. The placeholders of IO items values can be entered as IO Item Index or IO Item Name and enclosed in figure brackets {}.
Alarms.AlarmName.AckMessage	String	R/W	Contains exact characters, which should be sent back to GSM-Control to acknowledge the alarm. IO Items are not supported in AckMessage.
Alarms.AlarmName.AlarmItem.InputSource	String	R/W	Specifies name of external IAS attribute, which non-zero value

			initiates the alarm.
Alarms.AlarmName.AckItem.OutputDest	String	R/W	Specifies name of external IAS attribute, which receives the alarm acknowledgement.
Alarms.AlarmName.SendToAll	Boolean	R/W	Controls sequence how alarm SMS is sent to selected Users. If True, SMS is sent to all Users at once. When False, SMS is sent to Users one by one, with Timeout interval until someone acknowledges alarm or AlarmItem becomes False.
Alarms.AlarmName.Timeout	Integer	R/W	Specifies amount of time in seconds, how long to wait for Alarm acknowledge SMS prior sending alarm SMS to next User or repeat the alarm SMS sending.
Alarms.AlarmName.VoiceSource	String	R/W	Path to sound (WAV) file, which will be played (after the SMS sending) after a voice call command "ATD{phone number};" modem command successful execution. If this entry is empty then alarm voice call is not performed.
Alarms.AlarmName.Disabled	Boolean	R/W	In case True, the current Alarm configuration is not used in SMS processing.

Error and warning messages

This section contains the information about GSM-Control Object error and warning messages.

Text in ArcestraA Logger	Reason / Action
Demo license period expired.	The computer system time is outside of time interval in Demo license key. Please purchase software key or request new Demo license.
Exceeded demo license user limit!	The demo license has 10 Users limit. Reduce the number of Users.
Invalid software key for ProductID:PR0068X XXX CustomerID: hhhh-hhhh-hhhh-hhhh	The software key was not entered or it is invalid for computer where GSM-Control Object is deployed. Enter the purchased software key by using SwKey attribute.
Too many users for Product ID : PR00682 XXX	The selected product ID has exceeded allowed number of Users. Delete some Users or purchase the GSM-Control Object unlimited version.
Running in demo mode! Enter software key for ProductID:PR00680 XXX CustomerID:hhhh-hhhh-hhhh-hhhh	The GSM-Control object is working in one hour demo mode, because software key is not entered or is invalid.
Demo period timed out!	The GSM-Control Object finished to work in one hour demo mode. The outgoing SMS sending and received SMS handling is disabled. Redeploy the GSM-Control object, or enter purchased software key.
SMS received from {PhoneNumber} not handled due to license limitations. Acceptable phone numbers – LIST.	The GSM-Control Object has number of Users limitation. The received SMS sender is not in the GSM-Control Objects internal list of valid phone numbers after the first use of number. The number of valid phones cannot exceed the User limit. LIST shows locked phone numbers.
Timeout when set PIN code!	The AT+CPIN= command timed out. PIN code set state is uncertain, and further attempts enter PIN code are stopped. Please enter PIN code manually e.g. by using Hyper Terminal.
PIN code requested, but not configured!	Modem requests PIN code, but PIN code is not entered in GSM-Control Object configuration. Further communication with modem is stopped. Please enter PIN code or disable PIN code requesting.

Failed set PIN code! Failed execute PIN code query! Invalid response to PIN query!	AT+CPIN command failed or invalid PIN was configured. Further communication with modem is disabled. Please configure PIN code manually e.g. by using Hyper Terminal.
PIN1 or PUK code requested!	The modem is requesting PIN1 or PUK code. Further communication with modem is disabled. Please configure PIN code manually e.g. by using Hyper Terminal.
Failed parse PDU {PDU data} - error {Reason}	The received PDU data is in invalid or unsupported format. Please contact Wonderware Finland Technical Support.
SMS message is modified or truncated!	The GSM-Control Object does not support long SMS splitting in multiple SMS. The sent SMS was modified prior the sending.
Failed dial voice {Number} - error {Reason}	Failed to make voice call to phone Number.
Rejecting incoming call!	The modem received voice call. The GSM-Control Object hangs all received calls.
Network - not registered!	The modem is out of GSM network coverage.
Network - registered to home network.	The modem is registered in home GSM network.
Network - searching...	The modem searches GSM network.
Network - registration denied!	
Network - unknown!	Unknown error with GSM network registration.
Network - registered, roaming.	
Old simulated SMS not yet handled!	The simulated SMS was sent, but handling of previous simulated SMS was not finished. Please retry send simulated SMS.
Please enable simulation to use simulated SMS.	The Simulate Attribute is False. Change Simulate attribute in either runtime or by editor.
Custom message "Message"not sent! Phone number not yet specified. Please use SendPhone to set phone number.	The GSM-Control Object does not know the phone number were custom custom (not pre-configured) SMS should be sent. Use SendPhone attribute to specify the phone number.
Failed set Ack item ITEM.	The Ack item does not have write permission.
Failed open COM port "COMX"! Error Reason!	The COM port cannot be opened. Check other processes for using COM port or check

	hardware.
Status changed for port State1 to State2.	The communication with modem either failed or recovered.
Timeout error when sending AT COMMAND!	The modem did not sent response to AT command during the reply time-out.
Error response received to command "AT COMMAND"! – Error Code.	The modem responded with error code to AT command.
Failed set item "ITEM" to Value.	The IO item has no write permission.
Exception while writting to com port! Message – Error Message.	The writing to COM port causes exception.
Exception in com port read! Message : Error Message.	Reading COM port causes exception.
Exception in GSMEngine thread – Exception info.	The software exception was caught in scan period handler.
Exception in GSMComPort thread – Exceprion info.	The software exception was caught in COM port thread.
Exception in read callback! Message : Exceprion info.	The software exception was caught in asynchronous read handler.
Exception on com port write callback! Message : Exception info.	The software exception was caught in asynchronous write handler.

GSM-Calendar Object

This section explains the GSM-Calendar Object, what serves as an example how in practice to access/use the GSM-Control Object from Wonderware Application Server environment.

Overview

The GSM-Calendar Object allows to access and control GSM-Control Object attributes from Wonderware Application Server environment. GSM-Calendar Object is a simple calendar that controls when defined users will obtain SMS messages GSM-Control Object. It allows to configure GSM-Control Object once and then perform the scheduled receiving/sending of SMS messages to individual Users based on defined calendar. The GSM-Calendar Object is a standalone “user-defined” Application Server object together with its visualization control panels that can be shown on InTouch. The GSM-Calendar Object is a “freeware” type software that can be modified and copied to achieve special customer requirements, it is just an example how to build advanced GSM communication using GSM-Control Object.

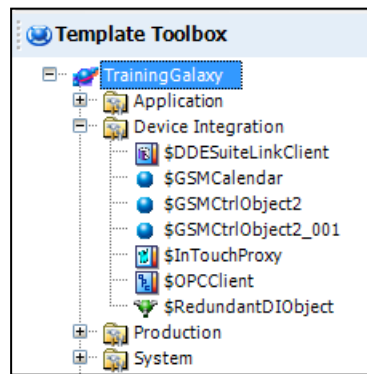
Files

The GSM-Calendar Object installation consists of one file:

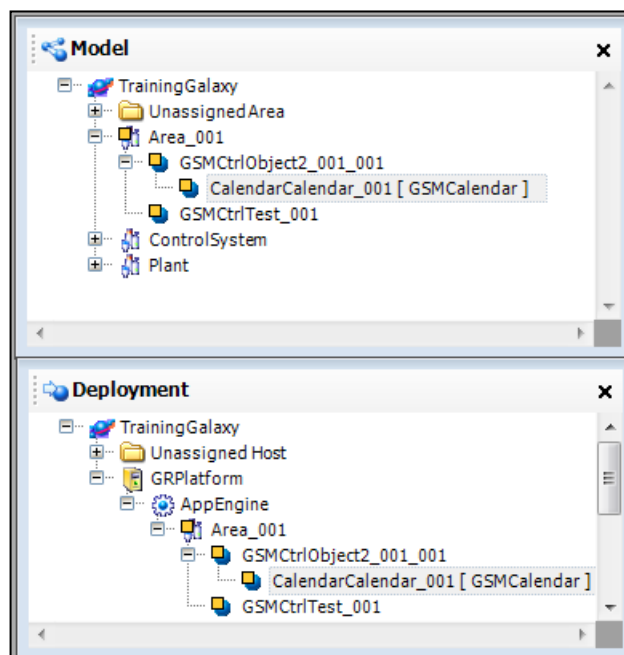
- **\$GSMCalendar.aaPKG** – the GSM-Calendar Object implementation code; the GSM-Calendar Object is optional demo object showing in practice how to access the GSM-Control Object attributes from Wonderware Application Server environment – it is a simple calendar that controls when defined users will obtain SMS messages from GSM-Control Object

Installation steps

1. Copy GSM-Calendar Object installation files to some folder, e.g. to C:/ GSM-Control Object folder.
2. Start the ArchestrA IDE and import the **\$GSMCalendar.aaPKG** file to a new/existing galaxy. After importing, the new template is added to Template Toolbox:



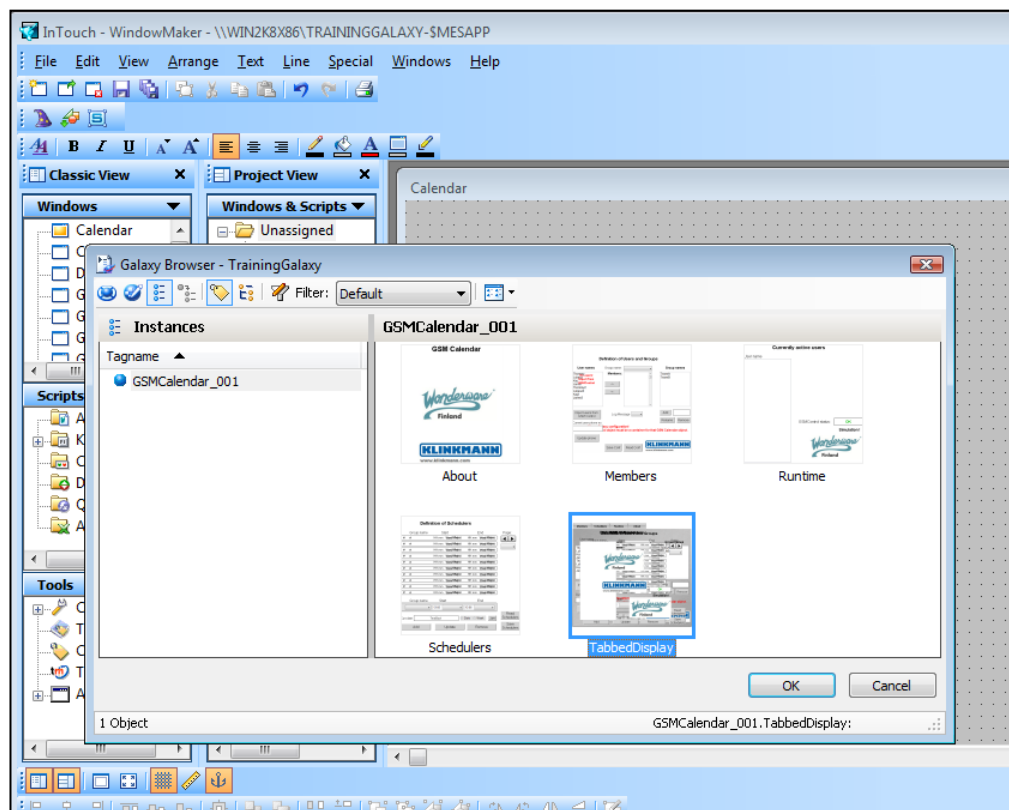
3. Create new instance of **\$GSMCalendar** and locate new instance of **GSMCalendar_001** inside GSMCtrlObject2_001_001 object (be sure that the new instance of calendar is a child of GSMCtrlObject2_001_001 object):



Now GSM-Calendar Object is ready for configuration, deployment and run-time use.

4. The GSM-Control Object should be deployed at first or together with GSM-Calendar Object. That order guarantees that GSM-Calendar Object can read and initialize all attributes properly.

5. For visualization on InTouch it is necessary to create the instance of “\$InTouchViewApp” template and inside “InTouch – Window Maker” application “Embed ArchestrA Graphic” of GSM-Calendar Object instance named “Tabbed Display”:



Members tab

The “Members” tab contains three main sets: Users, Groups and Members. On this tab you can create and configure groups and their members:

User names: area is a list of “Users” defined in GSM-Control Object. This is the list of people that will receive SMSs and are allowed to respond to GSM-Control Object. This list must be exactly the same as the list of “Users” in GSM-Control Object. To import current definition of “Users” from GSM-Control Object, use the “Import users from GSM Control” button.

Group names: area is a list of defined in GSM-Calendar Object groups. The group is a set that contains some “Users”. The group definition is used to set schedulers inside calendar. Group items can be added/removed/renamed.

Members: list of “Users” attached to the certain group. By selecting group name in drop down list (“Group name” in middle area), the list below the “Members” shows currently assigned members. Use horizontal arrows to add/remove users from selected group.

Log Message: integer value (0-3) that allows to control amount of information send to ArchestrA SMC Logger console. Whenever something is changed in

configuration, the system sends logs to SMC console. It is strongly recommended to open SMC log console and read logs from GSM-Calendar Object, this is the way which system uses to communicate with operator. If “LogMessage” level is set to 0 – no logs will be send to SMC console except errors and warnings.

Save/Read Config: Described above definition for Users, groups and members can be saved/read to/from the XML file which location is defined in UDA - “ConfigurationFilePath” (default value is “c:\GSMCalendar_config.xml”).

Schedulers tab

The “Schedulers” tab allow to add/update/remove definitions of schedulers. System allows to define up to 200 schedulers shown on 20 pages. The single scheduler contains the group name (the same as scheduler name) and date-time information. It is possible to use 4 different types of schedulers:

1. **Date** - with exact date and time for start and end. This scheduler lasts constantly for the whole time between defined start/end date-time. Look on the picture below at scheduler No. 1.
2. **Day of Week** - with exact start and end time described by days of the week. This scheduler is very similar to the first one, but instead of exact date here is the day of the week. Look on the picture below at scheduler No. 2.
3. **Recur Date** - this is a recurring date scheduler, that starts every day (between start/end date) from start time to end time. On HMI panel such schedulers are marked with additional ‘R’ inside the cycle image. Look on the picture below at scheduler No. 3.
4. **Recur Day of Week** - same as **Recur Date**, but instead of exact date here is the day of the week. Look on the picture below at scheduler No. 4

Each week starts on Monday, the 24-hours format is used for time and the date format is YYYY/MM/DD. The start date-time must be before the end date-time, exception is allowed when days of the week are used. When start date is e.g. Friday and end date is Monday, then it means that scheduler lasts constantly to the end of week and then to Monday (all together Friday-Saturday-Sunday-Monday). It is possible to select any of existing scheduler and change its type or date-time and update the existing one or add a new one or delete it.

Save/Read Schedulers: Described above definition for schedulers can be saved/read to/from the XML file, which location is defined in UDA - “schedulerFilePath” (default value is “c:\GSMCalendar_schedulers.xml”).

Members **Schedulers** Runtime About

Definition of Schedulers

	Group name	Start	End	
1:	TeamA	19:00 2011/11/27	19:00 2011/11/29	
2:	TeamB	19:00 Sunday	19:00 Tuesday	
3:	TeamC	19:00 2011/11/27	(R) 23:00 2011/11/29	
4:	TeamD	19:00 Sunday	(R) 23:00 Tuesday	
5:	Night	19:00 Monday	(R) 06:00 Monday	
6:	Emergency	08:00 Monday	(R) 11:00 Friday	
7:	Management	14:00 Tuesday	(R) 17:00 Thursday	
8:	HelpDesk	08:00 2011/12/02	08:00 2011/12/31	
9:				
10:				

Page: 1

Read Schedulers

Save Schedulers

Group name	Start	End
HelpDesk	08:00 2011/12/02	08:00 2011/12/31

24H

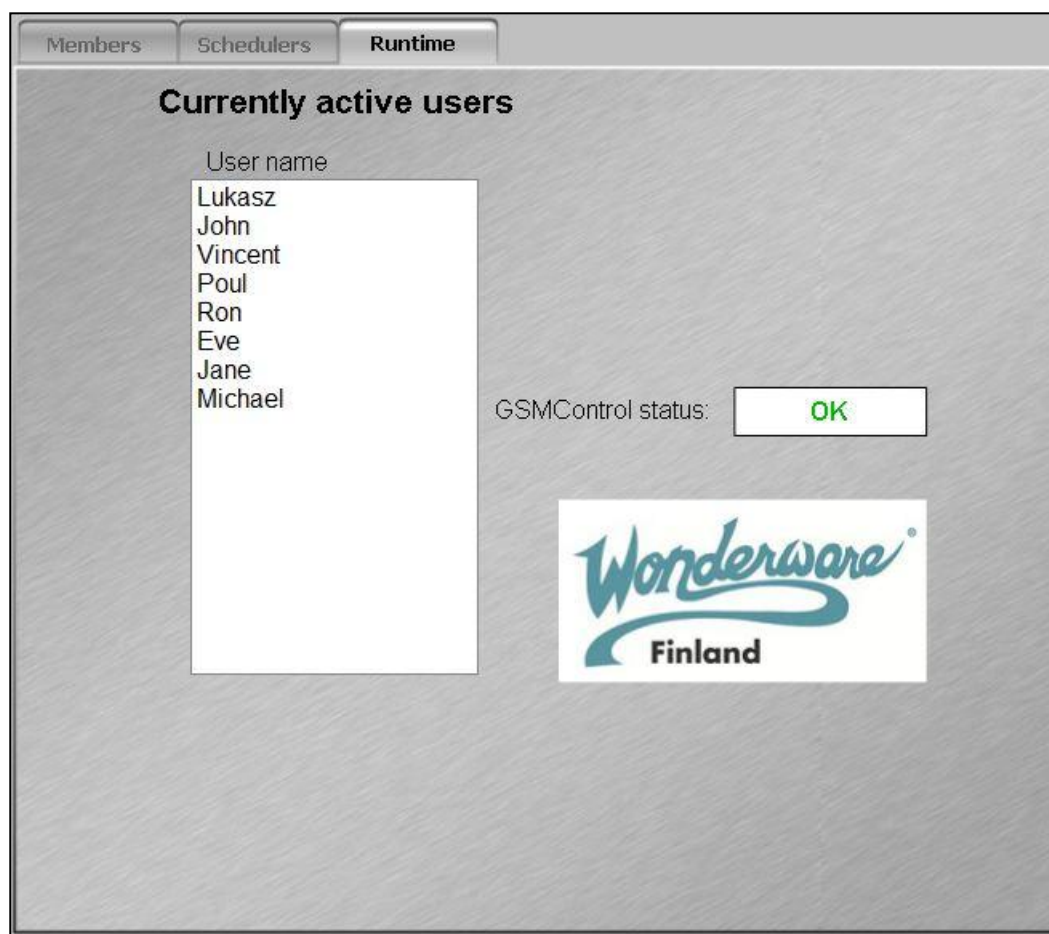
Curr. date: 10:33 2011/11/28, Monday

☒ Date
 ☐ Week
 ☐ Recur Date
 ☐ Recur Week

Add Update Remove

Runtime tab

The “Runtime” tab contains information which “Users” are enabled to receive/transmit SMSs. The GSM-Calendar Object based on its users/groups/members configuration and added schedulers automatically enable/disable Users in GSM-Control Object. The Users that are allowed to receive/transmit SMSs are shown on this list:.



GSM-Calendar Object configuration in details

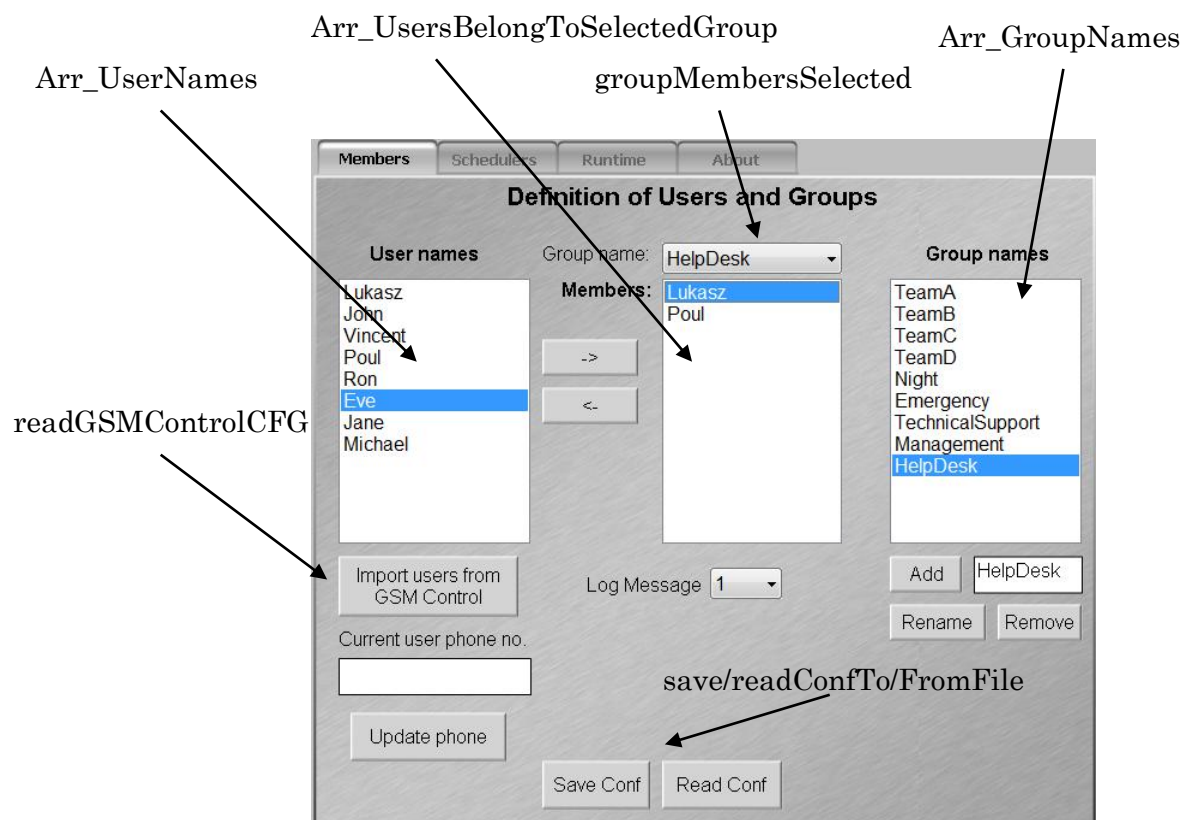
This section explains how the GSM-Calendar Object is built. That section should be read by anyone who would like to extend/adjust a functionality of GSM-Calendar Object to customer requirements.

General information

The code for GSM-Calendar Object is shared in two places: inside scripts of the object and in many objects (scripts, buttons, properties) on graphical HMI panel. The main functionality like: reading/saving configuration/schedulers, validation of configuration and schedulers, control of GSM-Control tasks are placed inside scripts of GSM-Calendar template, while adding/removing/updating/selecting any user/group/member/scheduler is controlled by HMI items like: buttons, lists, selectors, etc. Many data are stored in arrays (the UDA names start with “Arr_” letters), which length is assign dynamically with the exception for schedulers arrays. The general rule for arrays is that they are always filled from top – the new entry that must be saved in array always goes to the first free place in array from the top. If any item from array is deleted then all items below it are moved up.

UDAs on graphical panel

UDAs attached to the graphical panels helps to understand better the concept and architecture of GSM-Calendar Object:



Arr_ScheduleStartDate
Arr_ScheduleStartWeekDayName

Arr_ScheduleNames

Arr_ScheduleEndDate
Arr_ScheduleEndWeekDayName

Arr_ScheduleActive

save/readSchedulerTo/FromFile

Arr_ScheduleType

	Group name	Start	End
1:	TeamA	19:00 2011/11/27	19:00 2011/11/29
2:	TeamB	19:00 Sunday	19:00 Tuesday
3:	TeamC	19:00 2011/11/27 (R)	23:00 2011/11/29
4:	TeamD	19:00 Sunday (R)	23:00 Tuesday
5:	Night	19:00 Monday (R)	06:00 Monday
6:	Emergency	08:00 Monday (R)	11:00 Friday
7:	Management	14:00 Tuesday (R)	17:00 Thursday
8:	HelpDesk	08:00 2011/12/02	08:00 2011/12/31
9:			
10:			

Arr_ListOfActiveUsers
Arr_UserWillReceiveMessage

User name

- Lukasz
- John
- Vincent
- Poul
- Ron
- Eve
- Jane
- Michael

GSMControl status: OK

Wonderware
Finland

General Attributes

Attribute	Data Type	Access Type	Description
Arr_DaysOfWeek	String[]	R	Defines the order and names for days of week. The names of week must be the same as the current system uses. Method <code>date.toString("dddd")</code> must return the same names as defined in this array. Order of items inside this array can be changed to start the week on Sunday. The default order is Monday-Sunday.
Arr_GroupMembers	String[]	R/W	Defines the members (Users) assigned to the groups. Format: Group: User1, User2, ..., UserN. Users attached to the group cannot be duplicated (unique in one line), Group: cannot be duplicated inside the whole array.
Arr_GroupNames	String[]	R/W	Unique names of groups
Arr_ListOfActiveUsers	String[]	R	Unique names of Users that are enabled to receive/transmit SMSs. The order and array id is not equal to order and id inside the "Arr_UserNames" array.
Arr_ScheduleActive	Boolean[]	R	The length of array is 200. Each item in this array references to individual schedule definition. This array cannot change its size dynamically. The "True" value of the item in this array means that based on current date-time and start/end date-time defined for that scheduler this scheduler is active.
Arr_ScheduleEndDate	DateTime[]	R/W	The length of array is 200. Each item in this array references to individual schedule definition. This array cannot change its size dynamically. Contains the end date-time for each scheduler.
Arr_ScheduleEndWeekDay Name	String[]	R	The length of array is 200. Each item in this array references to individual schedule definition. This array cannot change its size dynamically. Contains the day of the week that refers to "Arr_ScheduleEndDate" array. This

			string is used when “Day of Week” type of scheduler is selected. The value for this UDA is automatically returned via method: date.toString("dddd")
Arr_ScheduleNames	String[]	R/W	The length of array is 200. Each item in this array references to individual schedule definition. This array cannot change its size dynamically. The array keeps the names of schedulers (the name of group which was selected during the definition of scheduler).
Arr_ScheduleStartDate	DateTime[]	R/W	The same as “Arr_ScheduleEndDate” but for start date-time
Arr_ScheduleStartWeekDayName	String[]	R	The same as “Arr_ScheduleEndWeekDayName” but for start date-time
Arr_ScheduleType	Integer[]	R/W	The length of array is 200. Each item in this array references to individual schedule definition. This array represents the type of each scheduler. Value = 0 means the scheduler type is based on exact date-time, Value=1 means the scheduler type is based on the days of week.
Arr_UserNames	String[]	R/W	Unique names of Users. This array must contain the same Users (not necessary in the same order) as GSM-Control Object has.
Arr_UsersBelongToSelectedGroup	String[]	R	The list of unique Users belonging to currently selected group (from drop down list on HMI)
Arr_UserWillReceiveMessage	Boolean[]	R	The array of Boolean values indicating if the certain User (inside “Arr_UserNames” array) is allowed to receive/transmit SMSs.
ConfigurationFilePath	String	R/W	Path to XML file that is used in reading/saving configuration data. Default destination is “c:\GSMCalendar_config.xml”
configurationHasChanged	Boolean	W	Boolean UDA that triggers “ConfigurationWasChanged” script. This

			script is called to validate many data in GSM-Calendar Object
currentTime	DateTime	R	Current time used to enable/disable each scheduler
groupMembersSelected	String	R	UDA that keeps information which group operator selects from the drop down list on HMI.
increaseLengthOfGroups	Boolean	R/W	Boolean UDA that works together with “resizeLengthOfGroups” UDA. This UDA shows the direction of resizing, “True” means “increase”.
LogLevel	Integer	R/W	Value from 0 to 3 determines how many information must be send to SMC log console. 0 means nothing is send, but errors and warnings are always send.
readConfFromFile	Boolean	W	Boolean UDA that triggers “ReadConfFromFile” script to start the reading configuration for XML file.
readGSMControlCFG	Boolean	W	Boolean UDA that triggers “ReadConfFromGSMControl” script to import already defined Users from GSM-Control Object. That trigger is set to true every time when GSM-Calendar Object is deployed.
readSchedulerFromFile	Boolean	W	Boolean UDA that triggers “ReadSchedFromFile” script to start reading schedulers for XML file.
resizeLengthOfGroups	Boolean	W	Boolean UDA that works together with “increaseLengthOfGroups” UDA. This UDA triggers “ChangeSizeOfGroups” script to resize the length of group arrays.
saveConfToFile	Boolean	W	Boolean UDA that triggers “SaveConfToFile” script to save configuration in XML file.
saveSchedulerToFile	Boolean	W	Boolean UDA that triggers “SaveSchedToFile” script to save schedulers in XML file.

schedulerFilePath	String	R/W	Path toe XML file that is used in reading/saving schedulers data. Default destination is “c:\GSMCalendar_schedulers.xml”

Scripts

The description of script is available at the beginning of all scripts with many comments inside the code.

Revision History

WONDERWARE FINLAND & BALTICS GSM-CONTROL OBJECT

Revision History

Oct 2011	Rev 1.0	First Release
Nov 2011	Rev 1.1	“GSM-Control Object configuration in details” added. “Description of GSM-Control attributes” added. “Error and warning messages” added.
Dec 2011	Rev 1.2	“GSM-Calendar Object” added.
Apr 2011	Rev 1.3	“Licensing agreements”, “Files” and “Installation steps” sections modified in “Installing the GSM-Control Object” section. “Getting and enabling license” section modified. “License tab” section modified.