

DDE client for OpcDbGateway and SAEAUT UNIVERSAL OPC Server

User's guide

by SAE - Automation, s.r.o., (Ltd.), www.saeautom.sk

OpcDbGateway is universal platform enabling integration of complex or easy applications preferably by configuring instead of more laborious programming. SAEAUT UNIVERSAL OPC Server is tool for easy development of dedicated OPC servers. Both products use common configuring and programming practices.

The DDE client for these products enables to integrate functionality of different applications with built-in DDE servers to a complex integrated application.

DDE client driver software consists of two parts - the runtime and the configuring one. The configuring part enables to configure applications although with enormous amount of DDE items easily and fast.



DDE client for OpcDbGateway and SAEAUT Universal OPC Server

Copyright © 2012 SAE - Automation, s.r.o. (Ltd.) All rights reserved. www.saeautom.sk

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: august 2012 in SAE - Automation, s.r.o. (Ltd.), Slovakia, Nová Dubnica, Trenčiansk 19, www.saeautom.sk, sae-automation@saeautom.sk

Publisher

SAE - Automation, s.r.o. (Ltd.)

Managing Editor

Vladimír Palacka.

Technical Editors

Tomáš Smolek

Cover Designer

Vladimír Palacka..

Team Coordinator

Vladimír Palacka

Production

Vladimír Palacka

Table of Contents

Foreword	0
Part I Introduction	3
Part II Features	5
Part III Usage	5
1 Installation	5
2 Demo mode and activation	6
3 First steps	7
4 After installation	8
5 Connecting DDE Servers	9
6 Defining and mapping of DDE items	10
7 Standard behaviour	10
Part IV The user interface (Overview)	11
1 DDE servers	12
Choose server	13
Generate items	14
Advanced settings	15
Memory operands used as DDE server interface.....	15
Other advanced settings.....	17
2 DDE item	17
3 External DII's	18
4 Memory operands	19
5 Synchronous controller	20
6 OPC items	21
7 Logging	22
Part V Easy DDE items defining	23
1 Browsing DDE servers	24
2 Choosing indexes and index ranges	26
3 Choosing MS Excel ranges	28
Part VI Easy DDE items features changing	31
Part VII Easy DDE items mapping	32
Part VIII Advanced configuring	33
1 Function blocks	34
DDE_PokeAllOutputsAsyn	35

DDE_RequestAllInputsAsyn	36
DDE_ExecuteMessageAsyn	37
DDE_Reconnect	38
DDE_PokeOneOutput	39
DDE_RequestOneInput	40
DDE_StartAdvLoopOneItem	41
DDE_StopAdvLoopOneItem	43
2 Memory operands for DDE servers and items	44
3 Constants used for DDE configuring	44
4 Logging DDE client functionality	45
Part IX Examples	46
1 System variables	46
2 DDE test book	47
Index	50

1 Introduction

[OpcDbGateway](#) is universal platform enabling integration of complex or easy applications preferably by configuring instead of more laborious programming. [SAEUT UNIVERSAL OPC Server](#) is tool for easy development of dedicated OPC servers. Both products use common configuring and programming practices. The DDE client driver for these products enables to integrate functionality of different applications with built-in DDE servers to a complex integrated application.

Using DDE client driver together with SAEUT UNIVERSAL OPC Server enables to use it as an **DDE client OPC Server** and to integrate legacy devices with DDE server to different SCADA systems with built-in OPC client.

Using it with OpcDbGateway has even much broader usage – connection to databases, a DDE server/OPC client communication, a data visualisation and calculations in Excel (activated by OpcDbGateway as DDE client)...

DDE client driver software consists of two parts - the runtime and the configuring one. The configuring part enables to configure applications although with enormous amount of DDE items easily and fast.

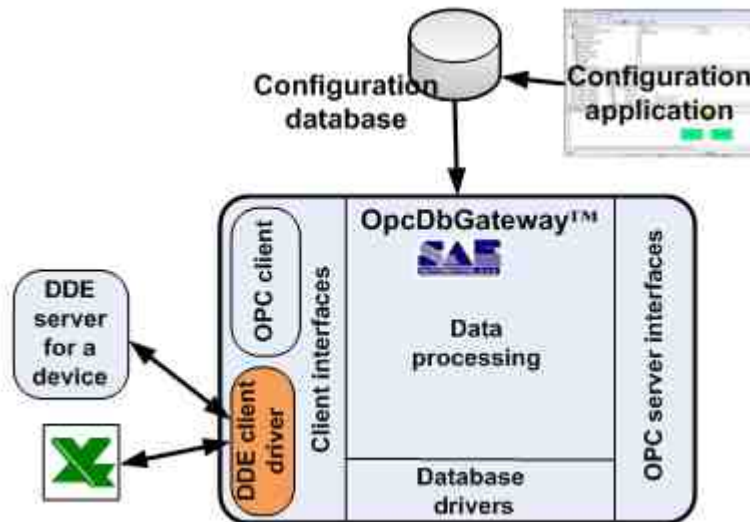


Figure 1: using of DDE client driver with OpcDbGateway

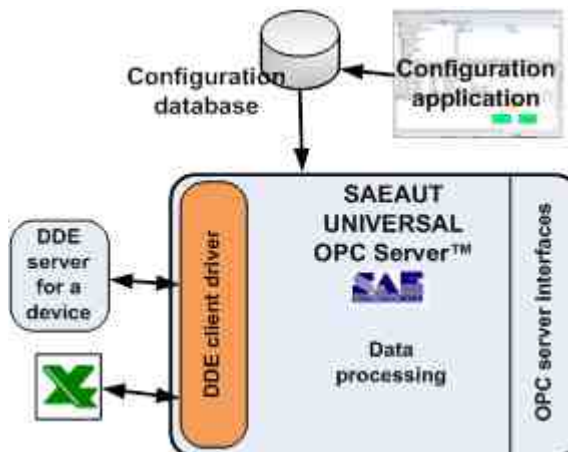


Figure 2: using of DDE client driver with SAEAUT UNIVERSAL OPC server

[DDE](#) stands for Dynamic Data Exchange. DDE is a method for exchanging information between applications running on the Windows operating system, all the way back to Windows 2 .0. As the first Windows-based HMI software came to market between 1989 and 1991, the developers chose to use DDE to connect the [HMI software](#) to their device drivers. The device driver that connected to a PLCs or process equipment ran as a separate application called a DDE Server. The HMI software was a DDE client application that requested and received data from the DDE Server. By having the DDE Server as a separate piece, it was very easy for 3rd parties to create drivers that would work with any HMI or other software application that used DDE. There are still many DDE servers for different devices in use. MS Excel can function as DDE server as well.

A server primarily waits for other Windows applications (clients) to send commands and then acts upon that command, which can instruct the server to perform a specific function or send data back to the client.

A client initiates conversations with the server by asking the server to perform a task. For example, if you need to send data acquired from your OpcDbGateway runtime to a report created in Microsoft Excel, your application acts as the client and Excel acts as the server.

If you communicate to a third-party software package through DDE, you must know what command strings the software understands. This DDE client provides generic DDE functions that you can use to send, receive, and execute commands. For information on the actual command strings and format of the data, consult the user manuals of the third-party software package.

Both products enable data exchange between different data sources over shared memory area - Process Image Memory (PIM) consisting of Memory Operands (MO) which enable to contain different basic data types.

Data exchange over MO can be configured using configuration application (see figures above) using configurable commands which are organised in function blocks. Function blocks are either called from special function block Main or started as configurable events. Events are initiated by configurable triggers which can be of the type time or value. Using this principles can be configured also very sophisticated data exchange scenarios.

Except of this configurable functionality OpcDbGateway can be enhanced also with programmed modules implemented as enhancing dynamically linked libraries (Dll). Also the DDE client driver itself is implemented as such enhancing Dll. Details can be found in the [OpcDbGateway help file](#) and in white papers:

[Enhancement Dll's for OpcDbGateway](#)

[Integration of applications effectively.OpcDbGateway – configuring and programming, overview.](#)

[Programming of OpcDbGateway and SAEAUT UNIVERSAL OPC Server](#)

However, not always it is necessary to understand all details of the configuring using above mentioned approach. The most often data exchange scenarios used for communication with DDE servers (as described in [Standard behaviour](#)^[10]) are supported by **enhancements of the standard configuring application** and so **user need not always to know all details about using of configurable commands.**

2 Features

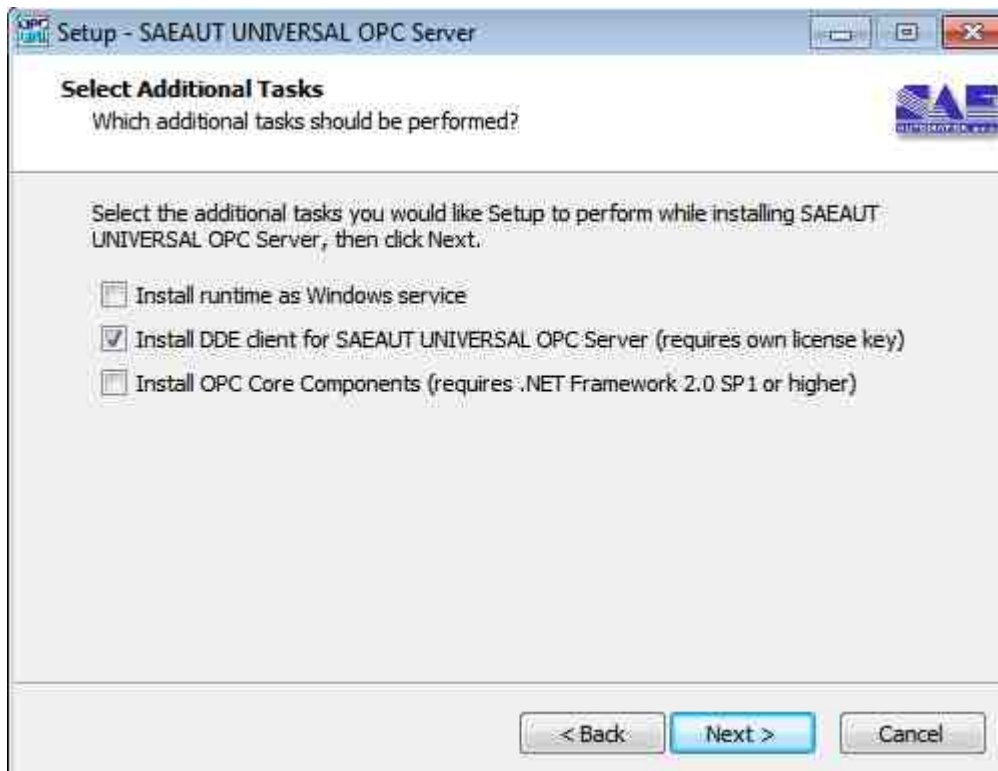
- The DDE Client Driver is capable of communicating with any server that supports the standard "CF TEXT" DDE data format. Addressing uses familiar "Application Name| TopicName ! ItemName" method for DDE server data.
- It can be used to connect with one or more DDE servers
- The driver will automatically reconnect when DDE servers are brought back on-line.
- A "DDEreconnect" tag is provided to aid in the resolution of DDE communication failures.
- This driver uses programmable/configurable ways of communication:
 - ["Hot links"](#),^[41] where a DDE server passes data to the driver only when the data value has changed. This provides a significant performance advantage over the alternative request/response loop approach since each DDE transaction is relatively time consuming
 - [reading of preconfigured group of items](#)^[36] from DDE server using request loop approach
 - [writing of preconfigured group of items to DDE server using poke loop](#)^[10]
 - [reading of individual item from DDE server using request](#)^[40] – only when the integrated application needs it
 - [writing of individual item to DDE server using request](#)^[39] – only when the integrated application needs it
 - [execute DDE commands](#)^[37]
- All major data types are supported
- High productivity by configuring although thousands of DDE items and, in the same time, high flexibility - different types of DDE functionality can be configured
- [High productivity](#)^[23] by configuring is provided by automatic mapping of DDE items to OPC items and to memory operands (used to interconnect different data sources) and also by possibility to use one of three ways of configuring:
 - Browsing of DDE server for available DDE items (if server has this functionality available)
 - Defining index ranges for DDE items (e.g. choosing of [1-1000] enables defining thousand DDE items in one step)
 - Specifically for MS Excel - by choosing of cell's range in a data sheet
- [High flexibility](#)^[33] by configuring - different types of DDE functionality are preconfigured as standard function blocks. It enables to combine DDE functionality with all other configurable or programmable functionality available in [OpcDbGateway](#) or [SAEAUT UNIVERSAL OPC Server](#)
- DDE functionality can be logged to file (in both) or database (only in OpcDbGateway) using standard built-in configurable logging functionality of the [OpcDbGateway](#) or [SAEAUT UNIVERSAL OPC Server](#)

3 Usage

This paragraph describes practical usage of the product - from installation and activation, through first steps by working with product till easy methods to configure functionality of the product and description of standard behaviour provided after applying easy configuration methods.

3.1 Installation

DDE client driver can be installed **within installation package of the container application** OpcDbGateway or SAEAUT UNIVERSAL OPC Server by choosing a related check box in dialog box (see below please).



Please download OpcDbGateway or SAEAUT UNIVERSAL OPC Server from following locations:

OpcDbGateway - x86

[Download URL 1](#) or [Download URL 2](#)

OpcDbGateway - x64

[Download URL 1](#) or [Download URL 2](#)

SAEAUT UNIVERSAL OPC Server - x86

[Download URL 1](#) or [Download URL 2](#)

SAEAUT UNIVERSAL OPC Server - x64

[Download URL 1](#) or [Download URL 2](#)

[OpcDbGateway or SAEAUT UNIVERSAL OPC Server - on line help](#)

[DDE client for OpcDbGateway and SAEAUT UNIVERSAL OPC Server - on line help](#)

3.2 Demo mode and activation

After installation, it is possible to use DDE client driver immediately in demo mode. It can run continuously with full functionality one hour. After that, to continue with testing, you have to restart the whole application. To remove this limitation, it is necessary to activate the DDE client driver.

The container application and DDE client driver are licensed separately. There are two distinct warning dialog boxes when running in demo mode - one for the container application and other for the DDE client driver itself. You can use those dialog boxes to activate a container application and/or DDE client driver.

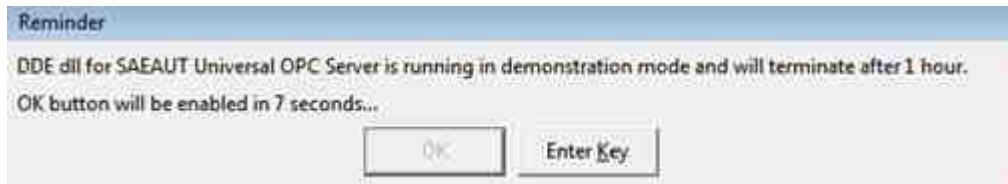


Figure 1: Dialog box for activation of DDE client driver.

In order to activate your software product to full version, you will need to enter a name and a license key. Please start the product you want to activate and click the button "Enter key" in the activation dialog box. You will see the dialog window that shows hardware fingerprint for your computer. Please send us this fingerprint together with a name under which you'd like to activate your product by e-mail to the address sae-automation@saeautom.sk. Based on this information, we will send you back the license key that can be used to activate your product.

3.3 First steps

1. Start the configuration application of the container application from Start menu (e.g. "OpcDbGateway Configurator" as in the Figure 1) The description what you should see is in paragraph [After installation](#)^[8].
2. Please learn basics of working with container application in [OpcDbGateway help file](#) first.
3. To test and learn the functionality of the DDE client driver, two demo applications are installed. Both applications use MS Excel as DDE server (which you need to have installed on your computer to use them). Applications are described in [Examples](#)^[46]. (Of course, you can use whatever DDE server application instead, to start with creating of own application from scratch.)
4. In the Program start menu under container application (Figure 1), you can find the directory "DDE client" with links on XLS sheets used by demo applications:
 - XLS server - System.variables - for [System variables](#)^[46] demo application
 - XLS server - Testing of DDE - for [DDE test book](#)^[47] demo application

To start whatever of demo applications, please click on related link to start the MS Excel application first.

5. Basic steps to create own applications communicating with DDE servers:
 - a) chose DDE servers you want to communicate with as described in [Connecting DDE Servers](#)^[9]
 - b) chose DDE items on the servers and map them to memory operands and OPC items
 - c) use standard functionality of the DDE client as described in [Standard behaviour](#)^[10] or configure required special functionality for working with DDE topics as described in [Advanced configuring](#)^[33]

Just after installation, the configuration **...\\Users\\a_user\\Documents\\OpcDbGateway\\SystemToExcel.odg** is chosen. This application enables transferring of system variables of the container application to the Excel sheet.

The second demo configuration ...\\Users\\a_user\\Documents\\OpcDbGateway\\DDETestBook.ODG enables to study all features offered by DDEclient driver.

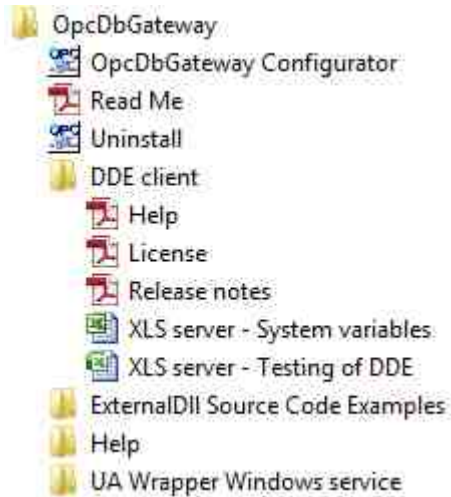


Figure 1: Start menu of the container application

3.4 After installation

When you start a configuration application of the container application for the first time, the configuration SystemToExcel.ODG (used for demo application [System variables](#)^[46]) will be automatically opened and checked. If the installation has overrun correctly, you will see a message "0 errors, 0 warnigs in the Checker Output window (Figure 1). In case that there are some errors click on a concrete error message and you will be navigated in the tree view to the place which caused the error.

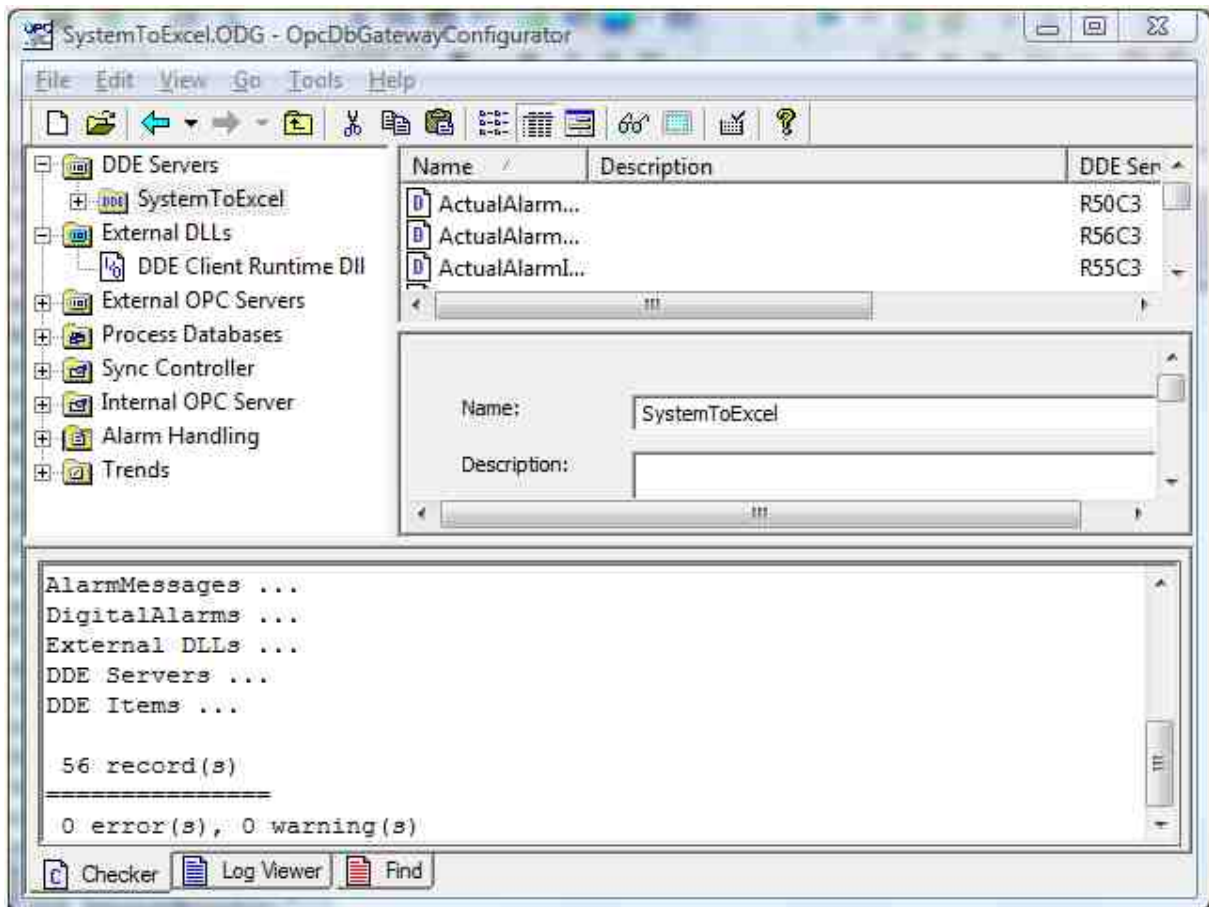


Figure 1 Checking of the configuration

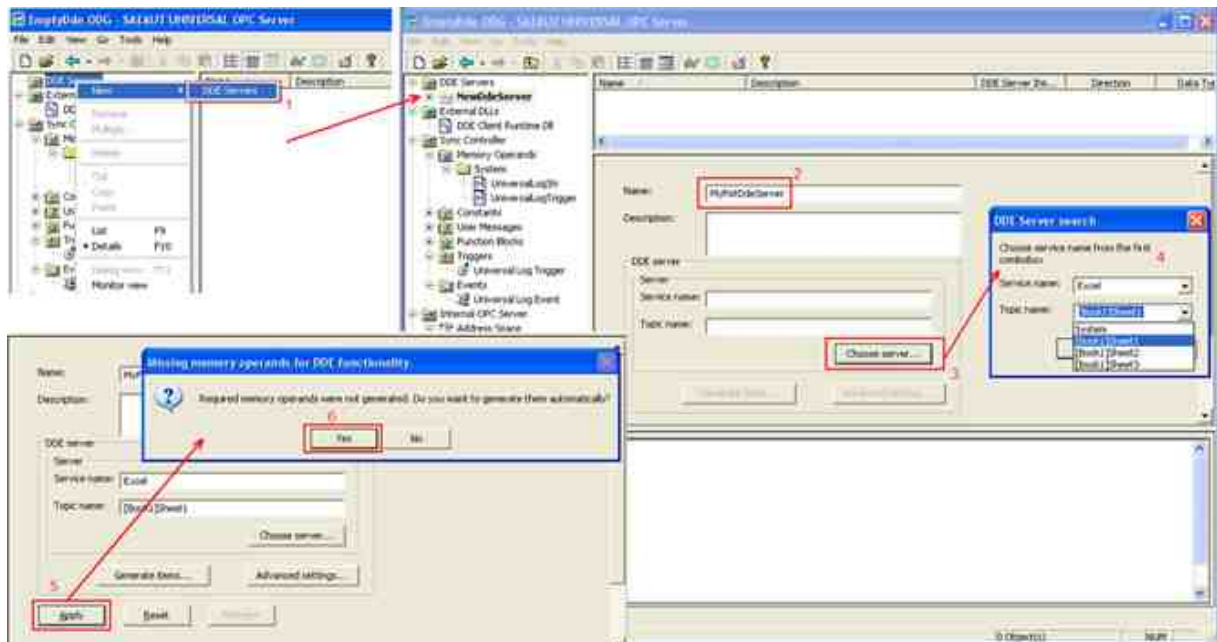
If you compare configuration application with and without installed DDE client driver you can see 2 specific items in tree view:

- one new item "DDE Servers" in root directory of the tree view
- one specific item in the folder External DLLs - DDE ClientRuntime Dll.

3.5 Connecting DDE Servers

Steps to setup connected DDE Servers according to the figure bellow:

1. Push the right mouse button on the "DDE Servers" folder and create new DDE Server element.
2. Write a name of the DDE server element.
3. Write a service name and a topic name of the DDE Server directly or open a dialog box "DDE Server search".
4. Choose your server (Service name and Topic name) from combo boxes. There are all currently running (discoverable) DDE Servers.
5. Push the Apply button to add your DDE server to configuration.
6. Generate required memory operands (MO) for the DDE server. Using these MO are mediated



3.6 Defining and mapping of DDE items

DDE items can be configured using one of three ways as described in [Easy configuring](#)^[23]. After that, it is possible to use standard functionality of the DDE client driver as described in [Standard behaviour](#)^[10]. To use all functionality of the DDE client driver, you can use [Advanced configuring](#)^[33].

3.7 Standard behaviour

By configuring DDE servers and DDE items within the [Easy configuring](#)^[23] process, the configuration wizard creates all commands, function blocks, triggers and events to provide a standard behaviour of the DDE client driver.

Standard behaviour consists of two basic activities:

- reading of values for a defined set of OPC items from address space of the Internal OPC server (or other data sources) of the container application (OpcDbGateway or SAEAUT UNIVERSAL OPC Server) to memory operands (MO) and writing them to the DDE items on a DDE server (Figure 1). MO's are the source and DDE items destination in this case. DDE dll provides watching of changes in MO's within a loop and changed values are transferred (poked) to the DDE server. Information about changed values can be written to log file if required.
- watching the notifications about value changes of DDE items set on the DDE server (within advice loop) and actualizing related OPC items.(Figure 2). DDE items are source and MO's destination of the data transfer.

Other types of communication with DDE servers can be configured according an integrated application needs as described in [Advanced configuring](#)^[33]

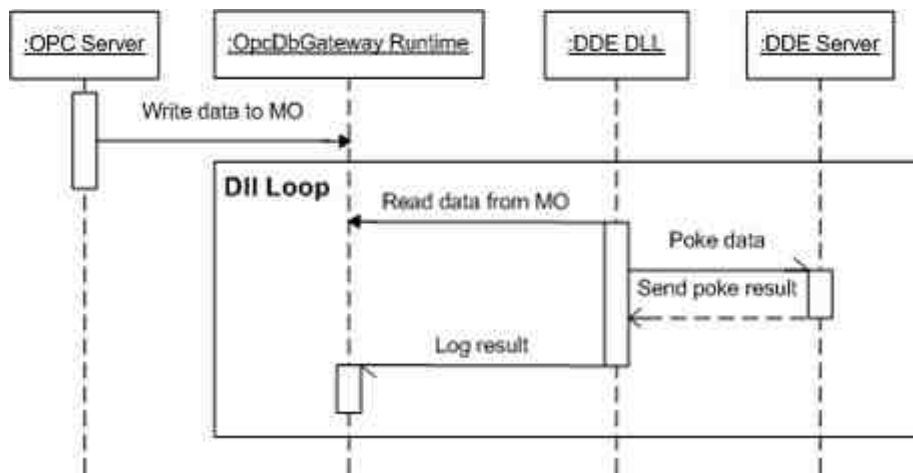


Figure 1: Data transfer MO ->DDE server

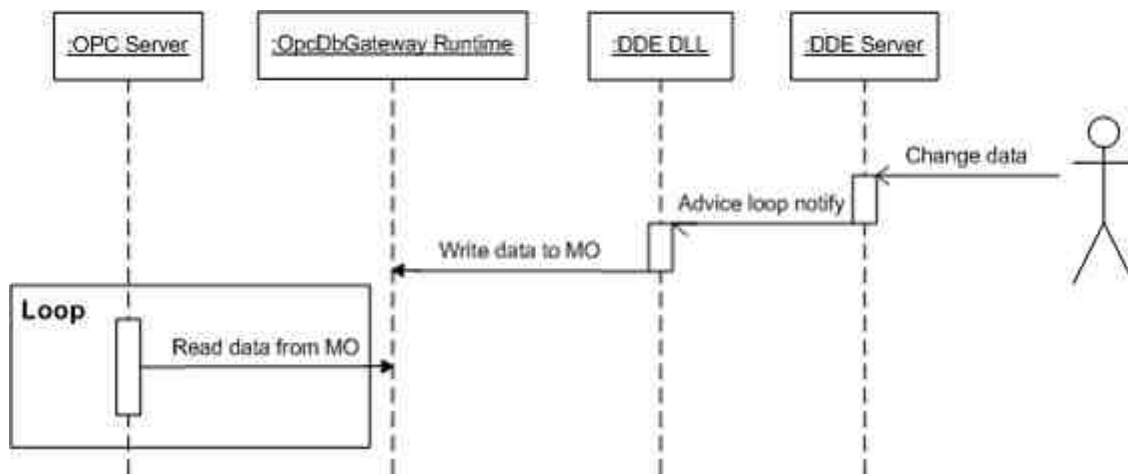
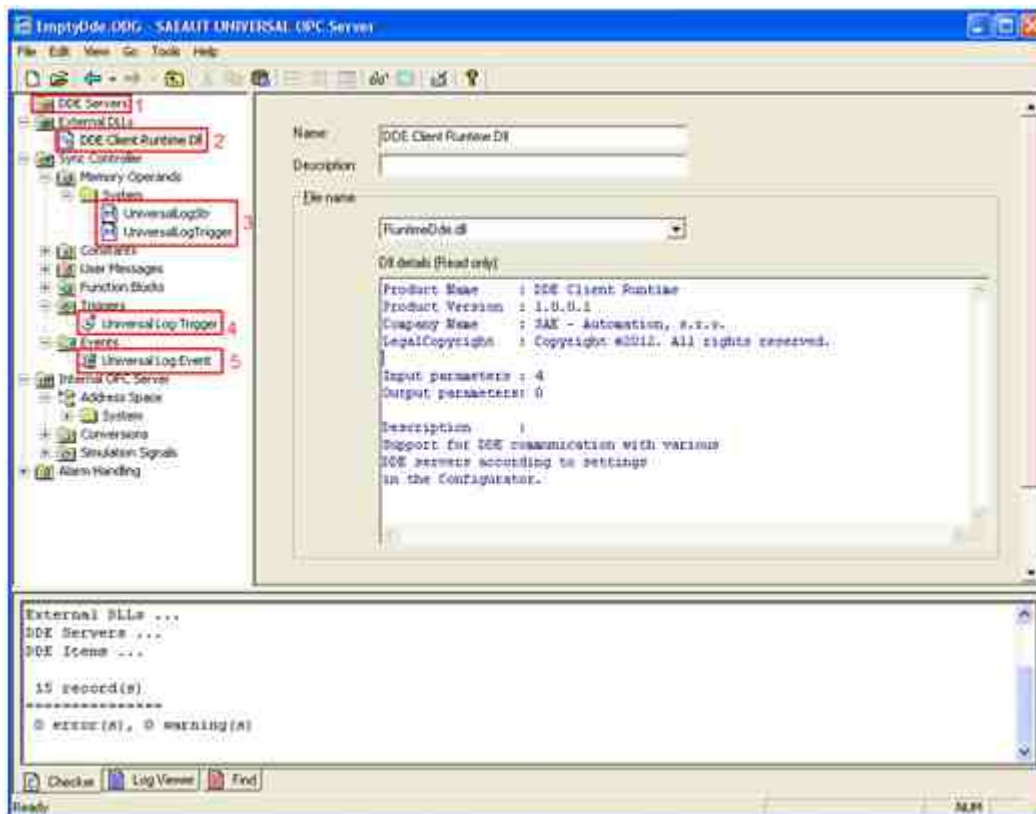


Figure 2: Data transfer DDE server ->MO using advice loop notify

4 The user interface (Overview)

To configure functionality of DDE client driver for [OpcDbGateway](#) or [SAEUT UNIVERSAL OPC Server](#) their configuration applications with following enhancements in the tree view according to the Figure bellow are used:

1. The folder for your DDE servers (It is the only for the DDE client driver specific enhancement in the root directory).
2. The definition for the enhancing runtime DDE dll. (As the DDE client driver is implemented as enhancing Dll you can see here the description of related runtime DLL but you need change nothing there.)
3. New system memory operands, which are necessary to control DDE DLL functionality and transfer data and for logging issues from DDE Runtime.
4. New trigger, that is necessary for log from DDE Runtime.
5. New event, that is necessary for log from DDE Runtime.



Remark

Figures in the "User interface" paragraph are from the example configuration [DDE test book](#)⁴⁷.

4.1 DDE servers

The folder DDE Servers in tree view (Figure 1) contains all DDE servers used as data sources or destinations in an integrated application.

As a DDE server in the configuration, a pair - Application| Topic is identified. For instance: Application - Excel, Topic: [DDETestBook.xlsx]TEST. Meaning of the term Topic in different applications differs. E.g. in MS Excel it consists of the pair [xls/file name]sheet name. In MS word topic is doc-file name.

Description of items in dialogbox below:

Name Descriptive DDE server name which can be chosen by user

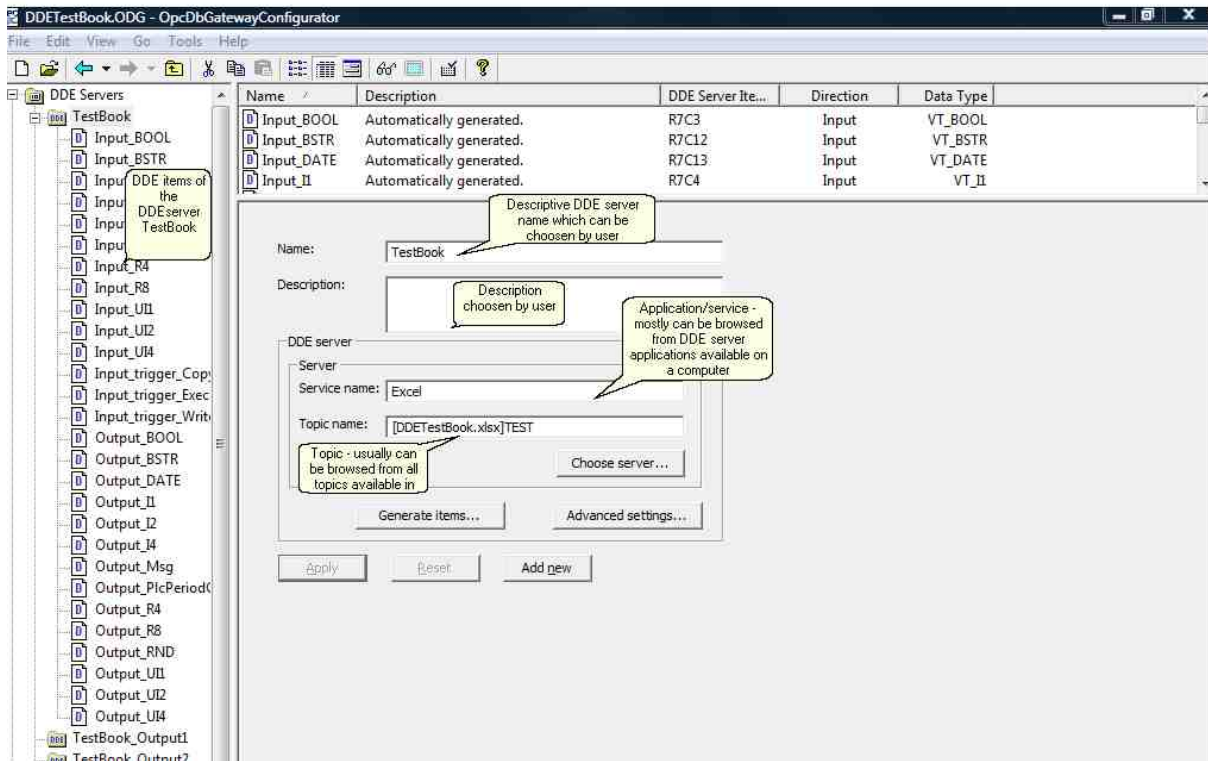
Description Text to describe

Section *DDE server*:
Service name
Topic name

Button *Choose server...* see in [Choose server](#)^[13] and [Browsing DDE servers](#)^[24]

Button *Generate items* see in [Easy DDE items defining](#)^[23]

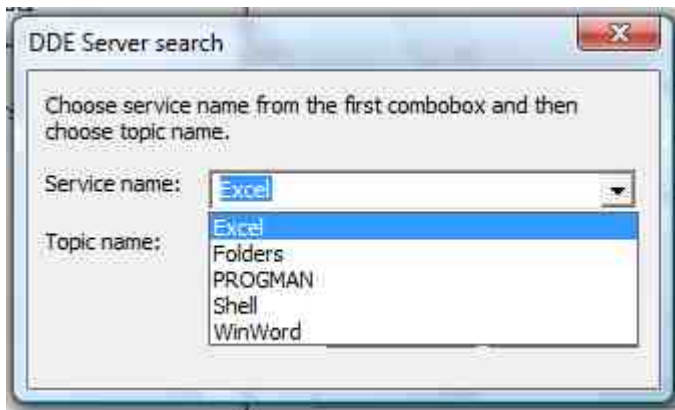
Button *Advanced settings* see in [Advanced settings](#)^[15]



4.1.1 Choose server

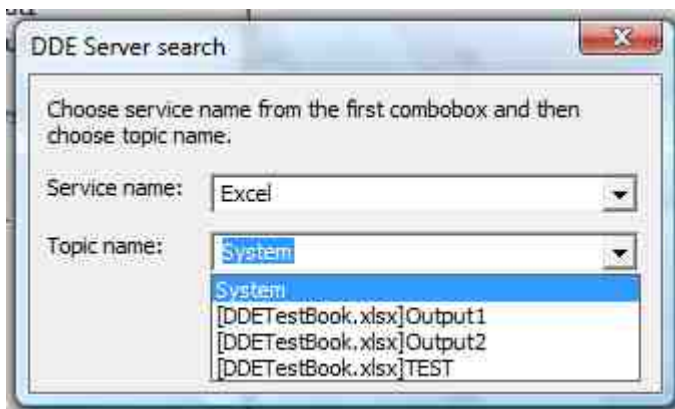
Service name can be usually browsed from all applications just running on a computer and offering a DDE server functionality.

In case that the application/service is not running you can write known service name manually



When service name has been correctly defined and the service is running, topics available within this service can be browsed.

In case that the application/service is not running you can write known topic name manually.

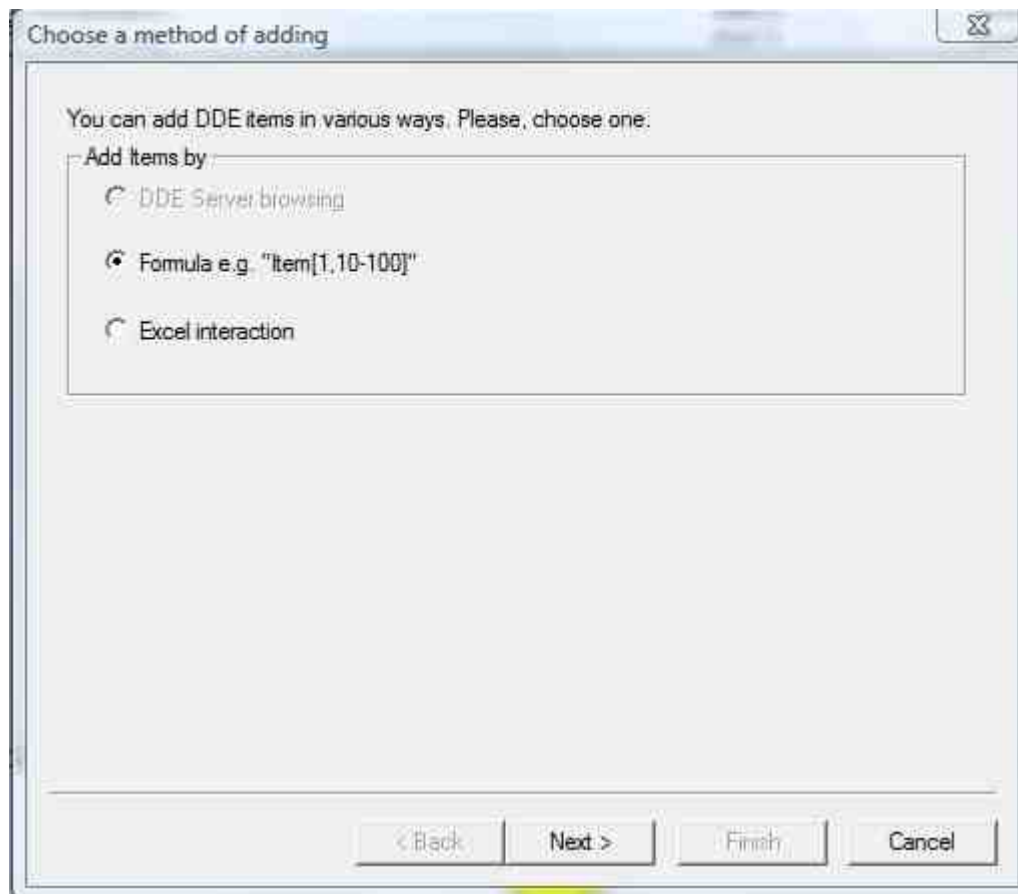


4.1.2 Generate items

There is a start dialog box of the wizard in the figure below to define although big amounts of DDE items efficiently.

Different ways to create DDE items in a DDE server are described in paragraph

[Easy DDE items configuring](#)^[23].



4.1.3 Advanced settings

It is called from the dialog box [DDE servers](#)^[12]. It contains tabs [Memory operands](#)^[15] to define memory operands to control communication with a DDE server and [Other](#)^[17] to define logging level, reconnecting and decimal symbol used in real numbers for a DDE server.

4.1.3.1 Memory operands used as DDE server interface

Memory operands below are used as interface between DDE runtime DLL and runtime of a container application.

Every DDE server has its own set of interface memory operands.

Main instruction switch

Enables choosing of functionality which have to be executed within DDE runtime DLL

Execute message

string for execute functionality of the DDE server

Reconnect trigger

setting this value to -1 forces reconnecting of the DDE server

Log message

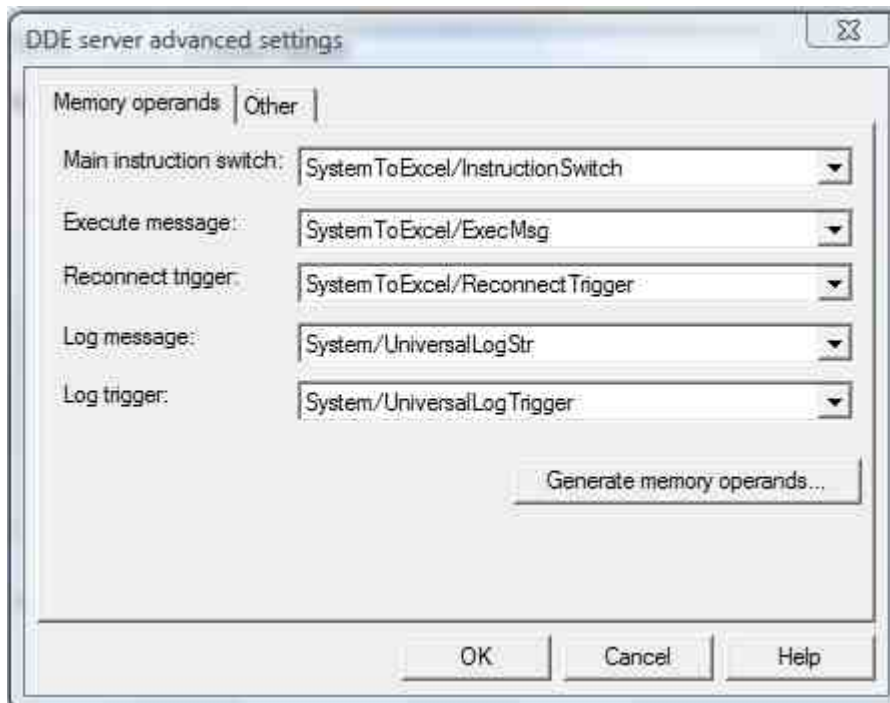
System memory operand used for placing of log messages from DDE runtime DLL.

These messages are then processed using standard logging functionality of the container application

runtime

Log trigger

System memory operand used as Trigger variable to initiate or disable logging functionality from DDE runtime DLL.



After click on button Generate memory operands following dialog box will be displayed:

Folder

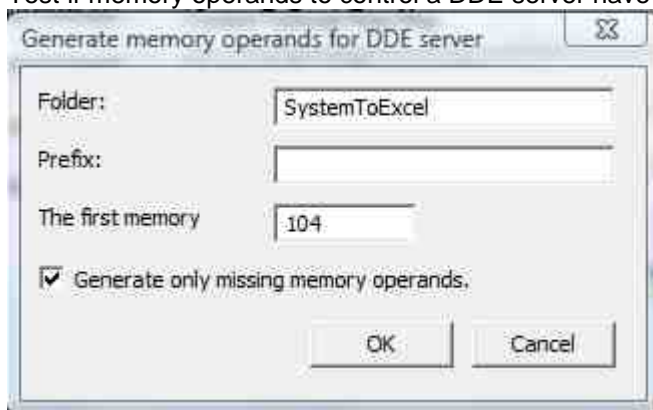
Sub folder under Memory operands folder used to place main instruction switch, execute message and reconnect trigger for a DDE server.

The first memory

Index to PIM where above mentioned three memory operands will be placed.

Generate only missing memory operands

Test if memory operands to control a DDE server have already been created if checked in.



4.1.3.2 Other advanced settings

Logging settings

to set [log level for DDE servers](#)⁴⁵

Reconnect settings

Reconnect timeout

timeout for trying to reconnect

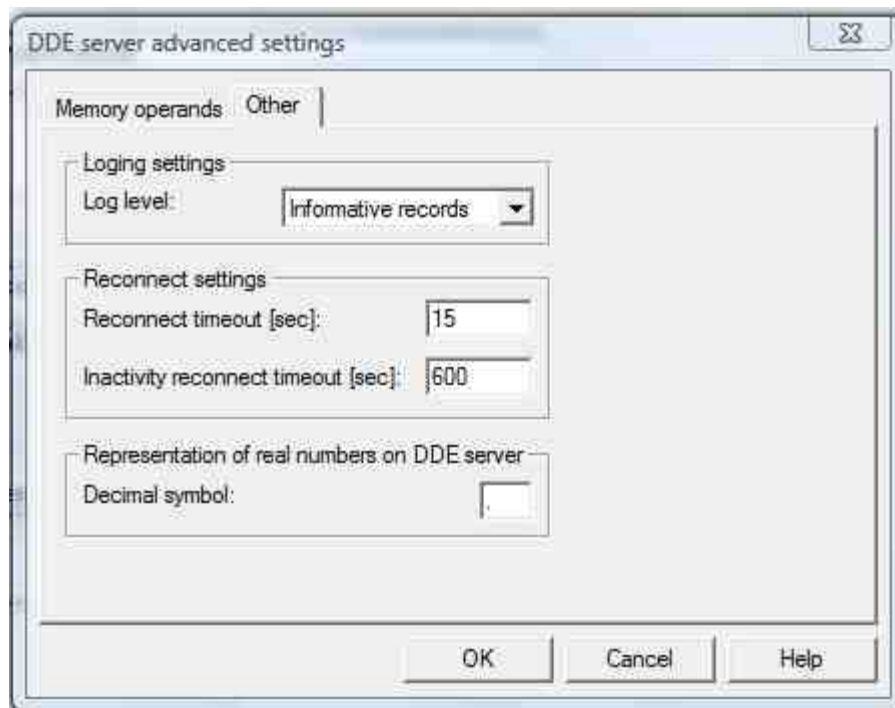
Inactivity reconnect timeout

time to retry reconnecting

Representation of real numbers on DDE server

Decimal symbol

it must be set according to really used one by concrete DDE server



4.2 DDE item

DDE items can be created using one of ways of [easy DDE items defining](#)²³ or manually using dialog box bellow.

If you would like to create new DDE item for a DDE server chose New in context menu of this DDE server in tree view.

The lower part of the dialog box bellow enables reading and writing DDE item values from/to a DDE server if it is on line. It can be used for diagnostic purposes.

Description of items in dialog box bellow:

Name Descriptive DDE item name which can be chosen by

user

<i>Description</i>	Text to describe DDE item
<i>Memory operand</i>	MO to map DDE item to
<i>Name on server</i>	Name to access to DDE item on DDE server
Direction	Communication direction: Output - from MO to DDE server,
Input - DDE server to MO, Both	
Section <i>Item value on server:</i>	Value to write or read to DDE server when it is online
-	Edit box to read value of DDE item
Button Get	read value from DDE server
Button Set	write value to DDE server

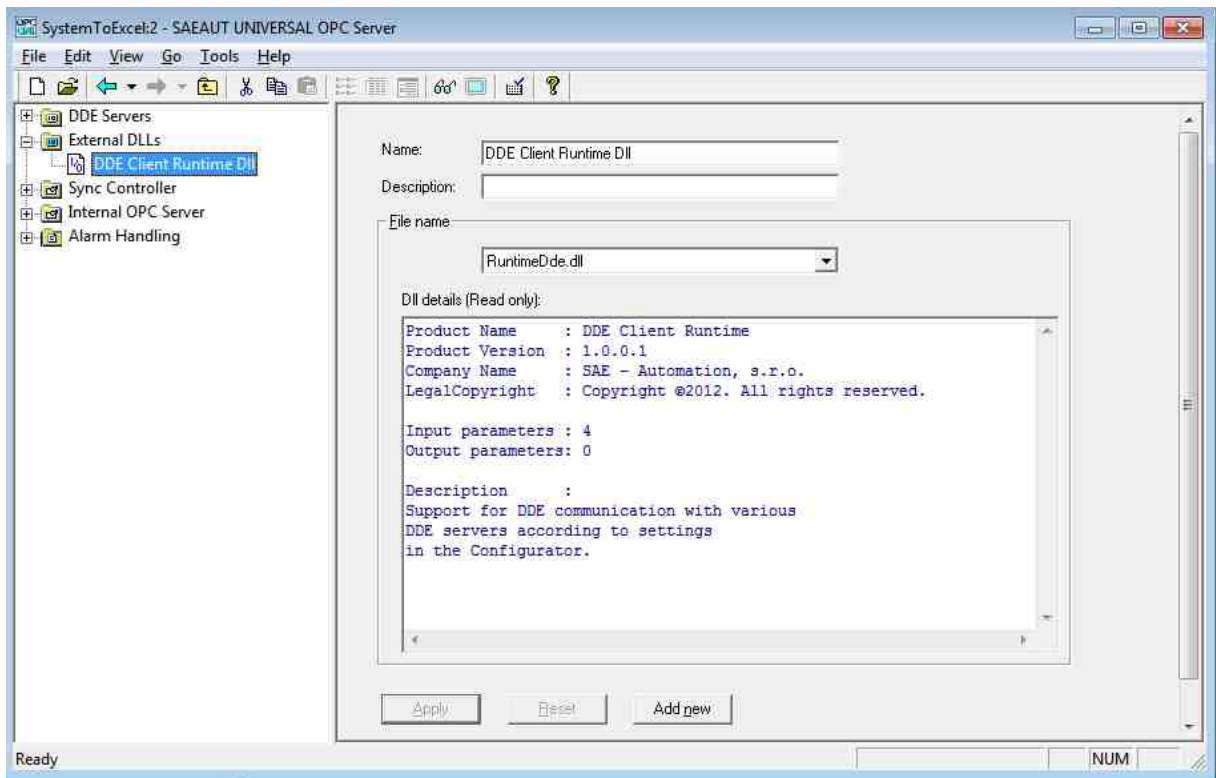
The screenshot shows a configuration window for a DDE client. It includes the following fields and controls:

- Name:** PlcCycle
- Description:** (empty text area)
- Memory operand:** System/PlcCycle (with a dropdown arrow and a help icon)
- Name on server:** R9C3
- Direction:** Output (dropdown menu)
- Item type:** LONG (dropdown menu)
- Buttons:** Apply, Reset, Add new
- Item value on server (It's not part of configuration.):** 1 (edit box)
- Buttons:** Get, Set

4.3 External DII's

DDE client functionality is implemented within OpcDbGateway and SAEAUT UNIVERSAL OPC Server as a **standard enhancing DLL** as described in [their documentation](#). Therefore you can use many advanced configuring features as described in [Advanced configuring](#)³³.

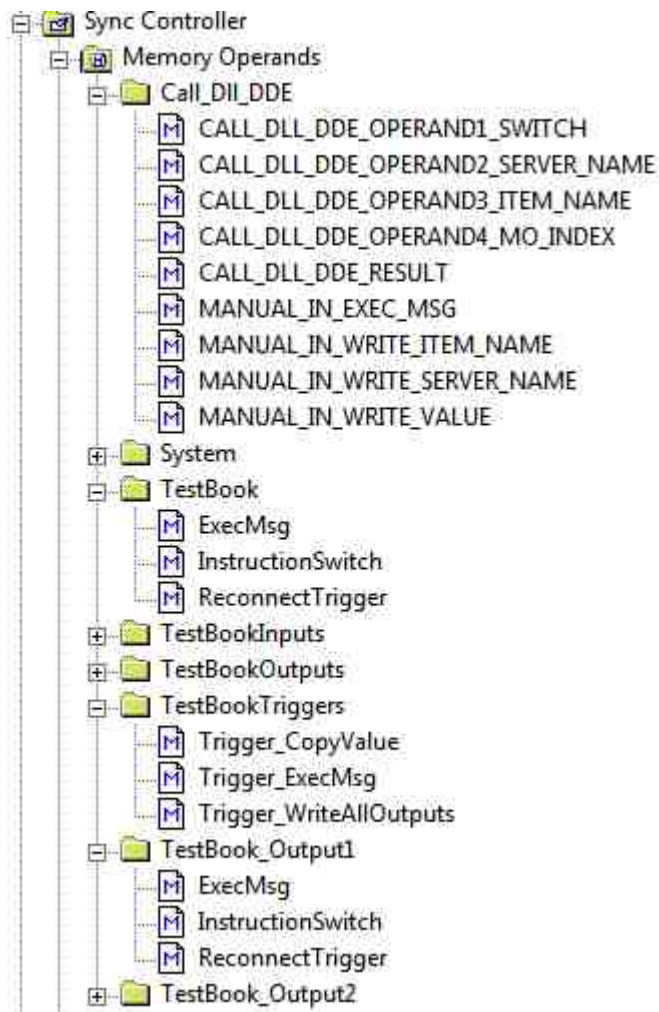
As every enhancing DLL, it is shown also in configuration tree as displayed bellow.



4.4 Memory operands

All DDE items and also some control variables from runtime DDE DLL are mapped to memory operands.

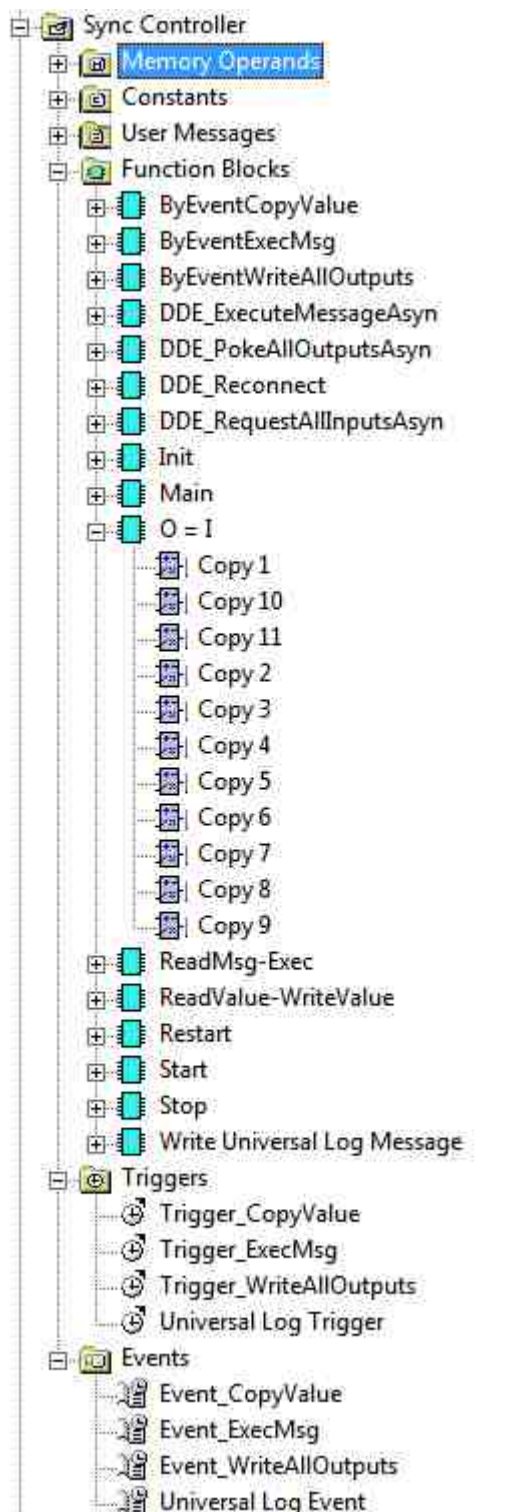
There are MO's for the [DDE test book](#)^[47] in the figure below. You can see folders for input and output variables of the data transfer within folder TestBookInputs and TestBookOutputs. Memory operands of this type are mostly configured automatically within [Easy configuring](#)^[23] process. There are also MO's MANUAL_IN_EXEC message used for specific applications configured within [Advanced configuring](#)^[33] process.



4.5 Synchronous controller

Under this folder all function blocks, memory operands, constants, triggers and events used by specific integrated applications are organised. The configuring of that is described in [OpcDbGateway help file](#) and in [Advanced configuring](#)^[33].

In the figure below, you can see a configuration for the example [DDE test book](#)^[47].

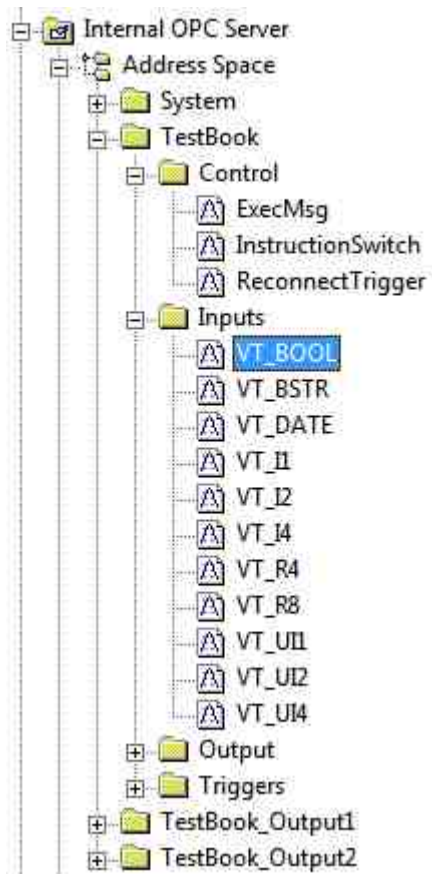


4.6 OPC items

OPC items of the internal OPC server in a container application are one of configurable sources or destinations for data transfers with DDE servers.

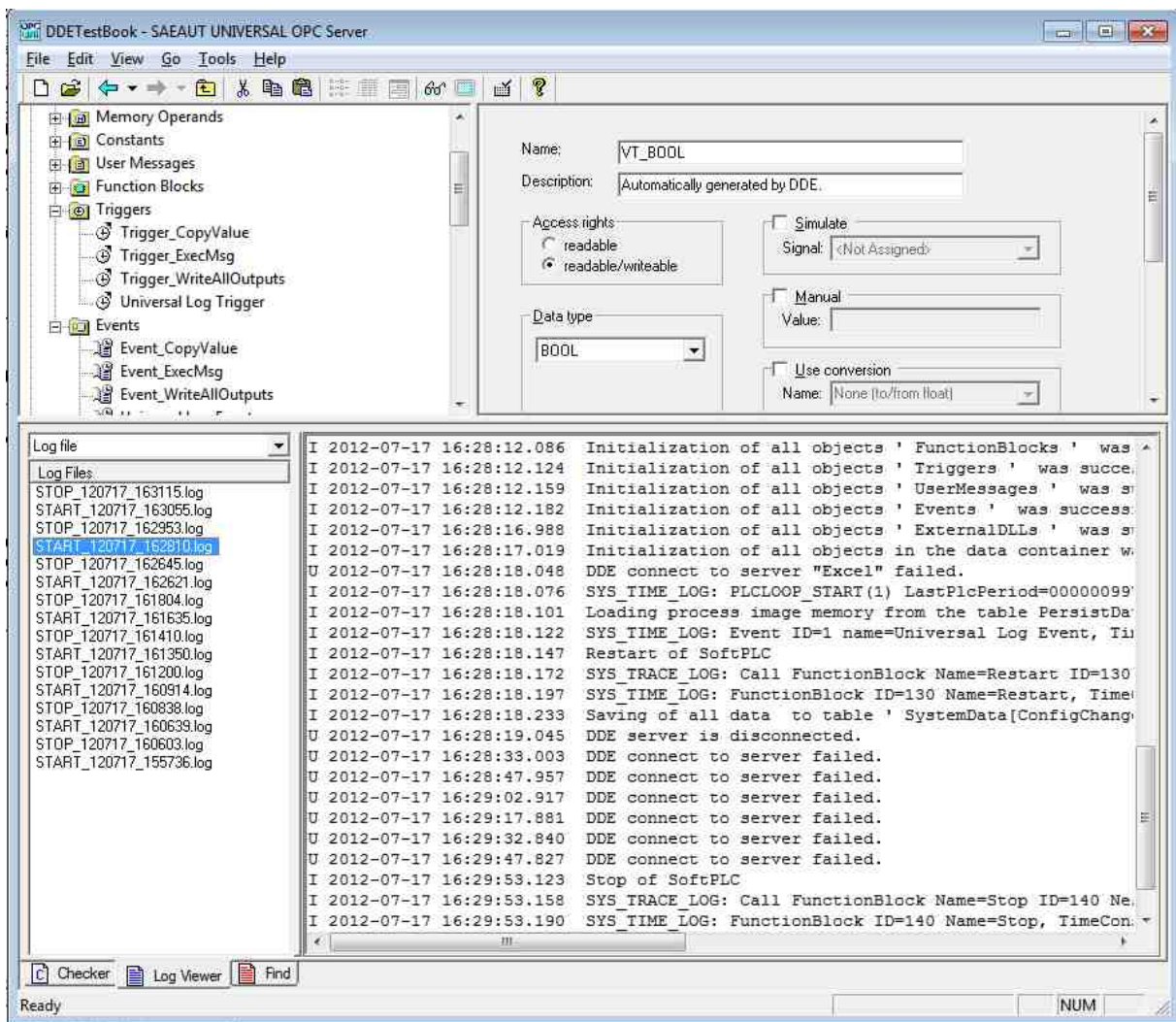
They can be created automatically within [Easy DDE items mapping](#)^[32] process. This functionality is important mainly when many source/destination DDE data points are used.

In the figure below, there is address space of the internal OPC server for the example application [DDE test book](#)^[47].



4.7 Logging

There is possibility to provide logging of DDE client driver activity within standard logging functionality of the container application. Types of messages please see in [Logging DDE client functionality](#)^[45]

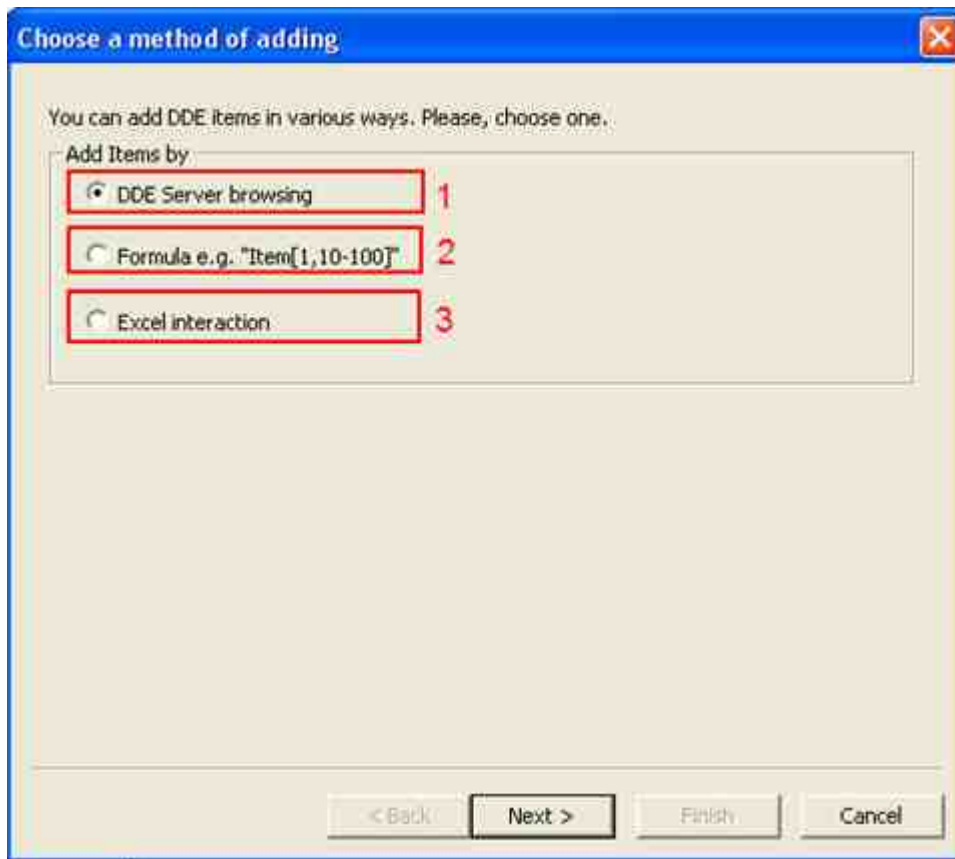


5 Easy DDE items defining

When basic properties of the DDE server are set, you can determine the relationships between elements of the DDE server (e.g. Excel sheet) and DDE client (OpcDbGateway).

There are various ways in the configuring application to add new items automatically. The first step in all ways is to push the "Generate items..." button, which is on the dialog of a DDE server configuration (see in [DDE servers](#)^[12]). Then you can see dialog with radio button selection.

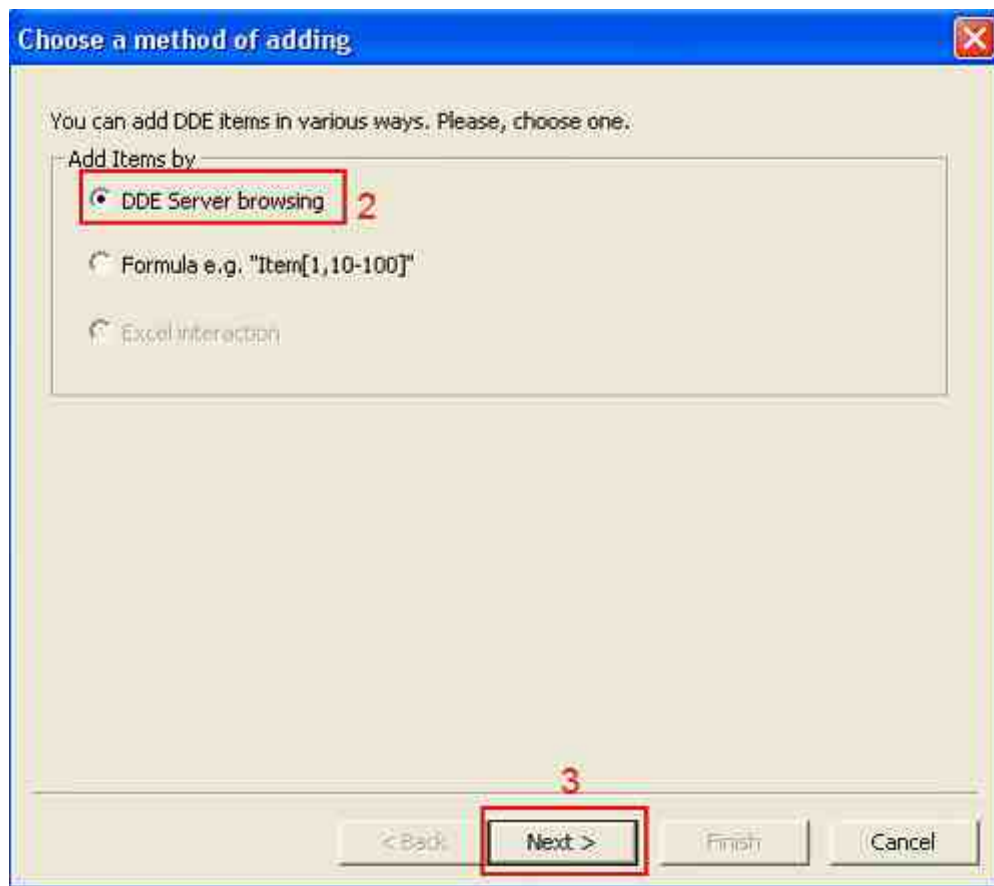
1. Add new items by browsing DDE server. DDE server can return in the item "TopicItem" name of supported items in topic. But, not all DDE servers support this feature.
2. To add a number of similar items with different index as the extension, you can use a formula. E.g. ItemX1-ItemX1000.
3. You can add new items by special functionality for Excel. User can select items directly in an Excel sheet.

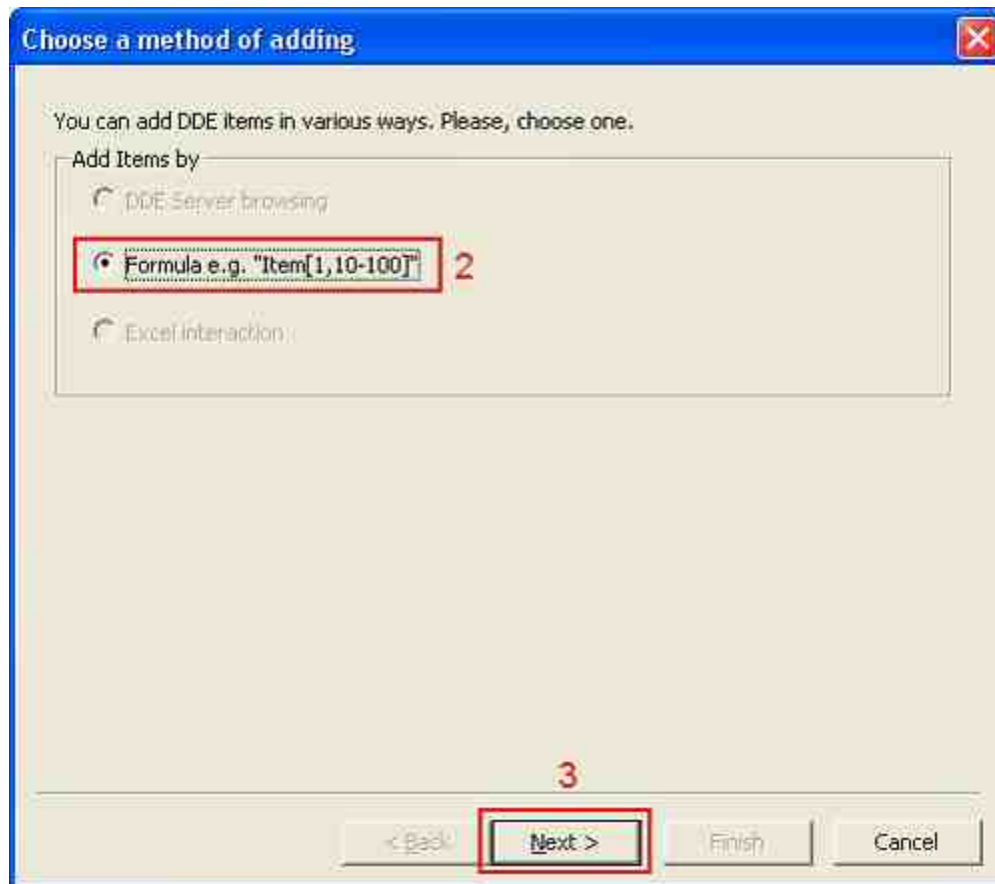


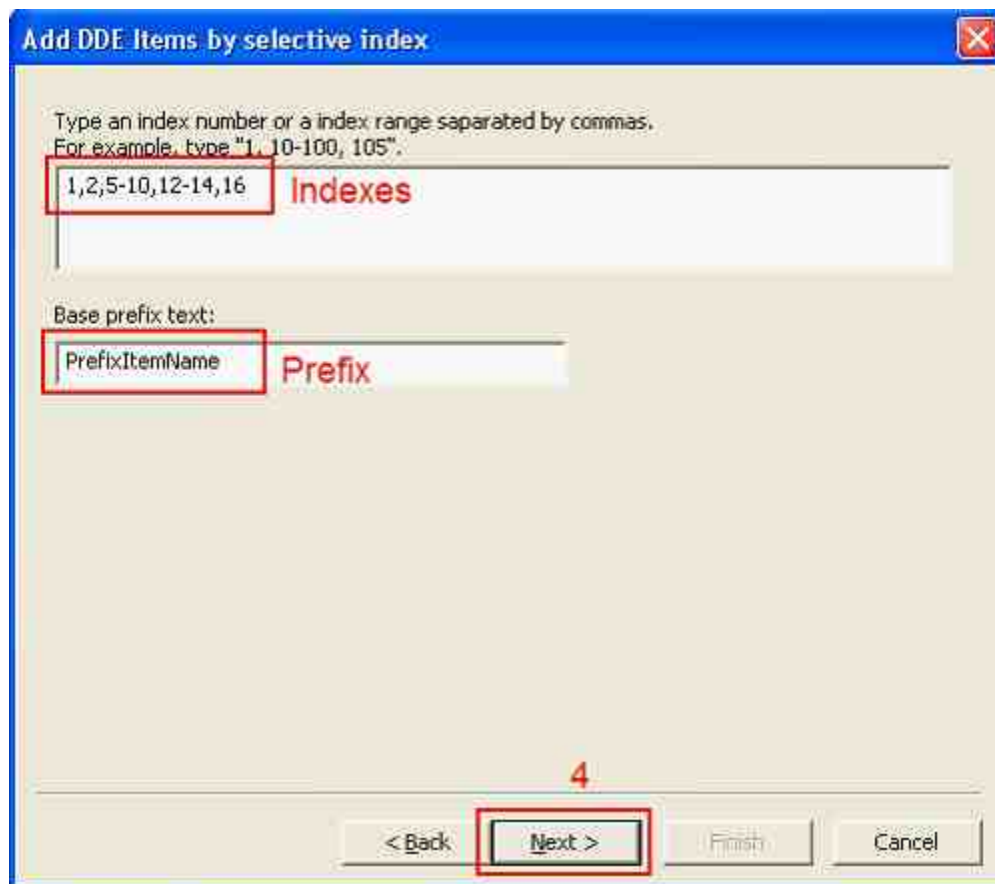
5.1 Browsing DDE servers

You can add items by browsing of the DDE server. Follow these steps:

1. Set a service and a topic name of the DDE server you want to use and then push the "Generate items..." button (see in [DDE servers](#)^[12]).
2. Select DDE server browsing.
3. Push the "Next >" button.
4. In the next window are items with values from the DDE server. Select items, that you want to add to configuration and push the "Next >" button.
5. Next steps are described in paragraphs [Easy DDE items features changing](#)^[31] and [Easy DDE items mapping](#)^[32]

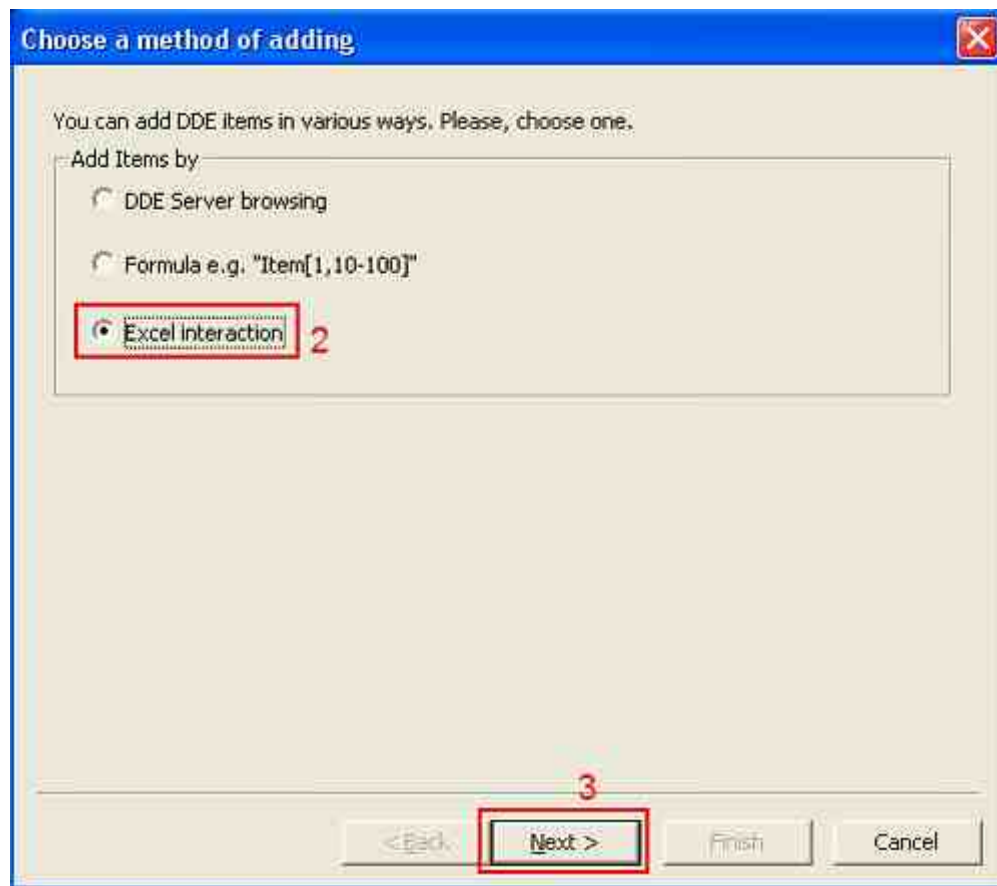


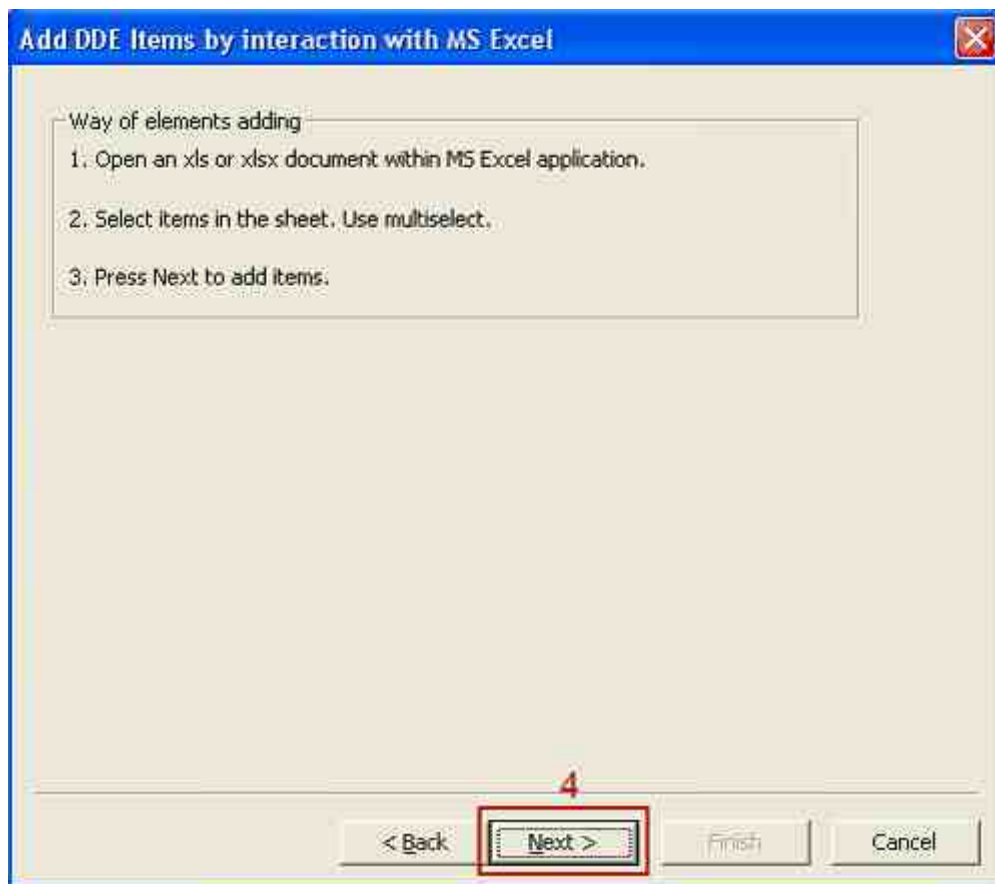


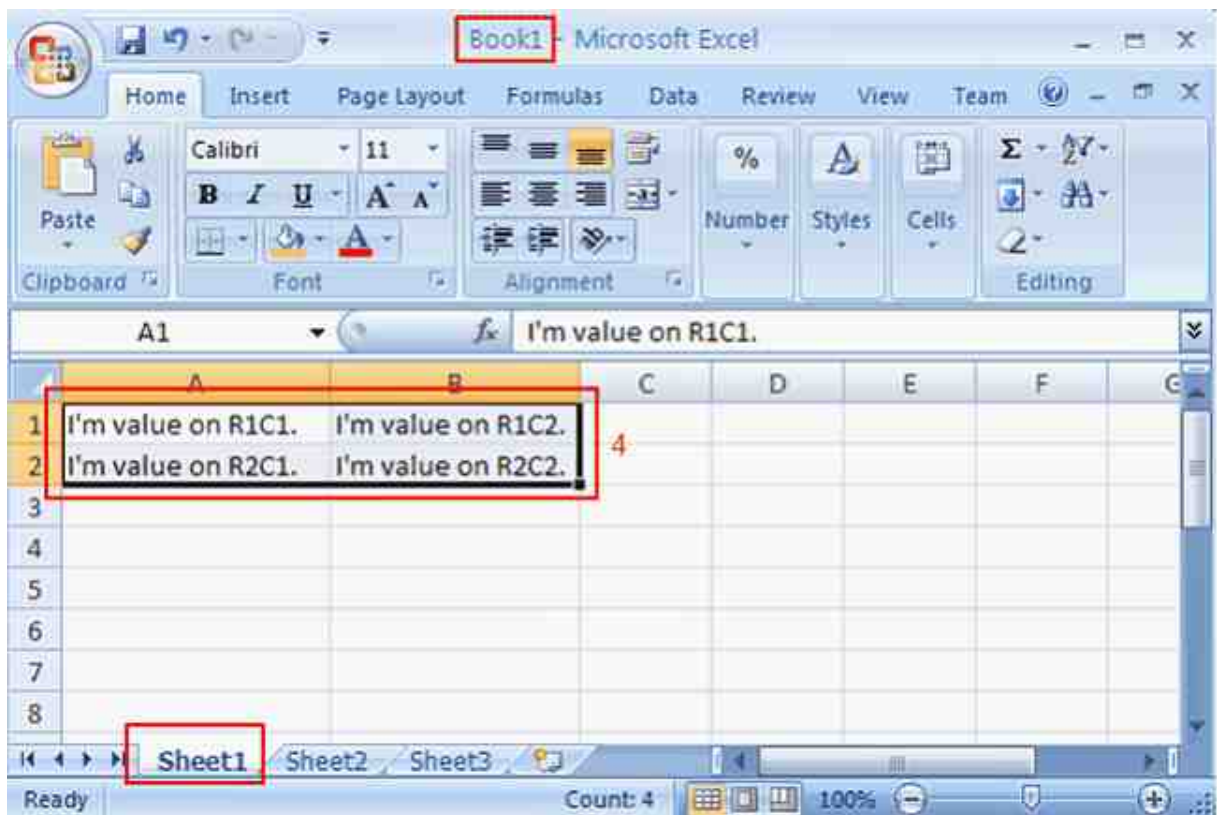


5.3 Chosing MS Excel ranges

1. Set "Excel" as service name and topic like as "[Book1]Sheet1". Push the "Generate items..." button- (see in [DDE servers](#)^[12])
2. Select Excel interaction.
3. Push the "Next >" button.
4. In a corresponding Excel sheet select required items. You can use multi select. Then push the "Next >" button in configuring wizard.
5. Next steps are described in paragraphs [Easy DDE items features changing](#)^[31] and [Easy DDE items mapping](#)^[32]

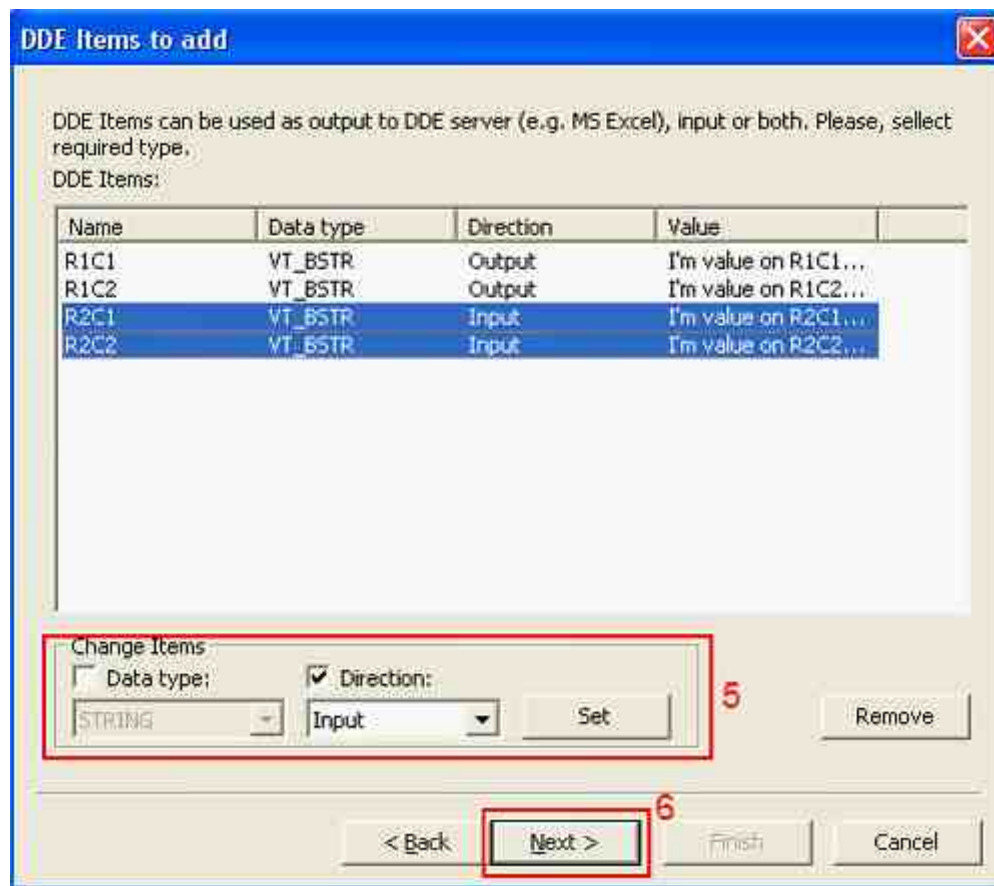






6 Easy DDE items features changing

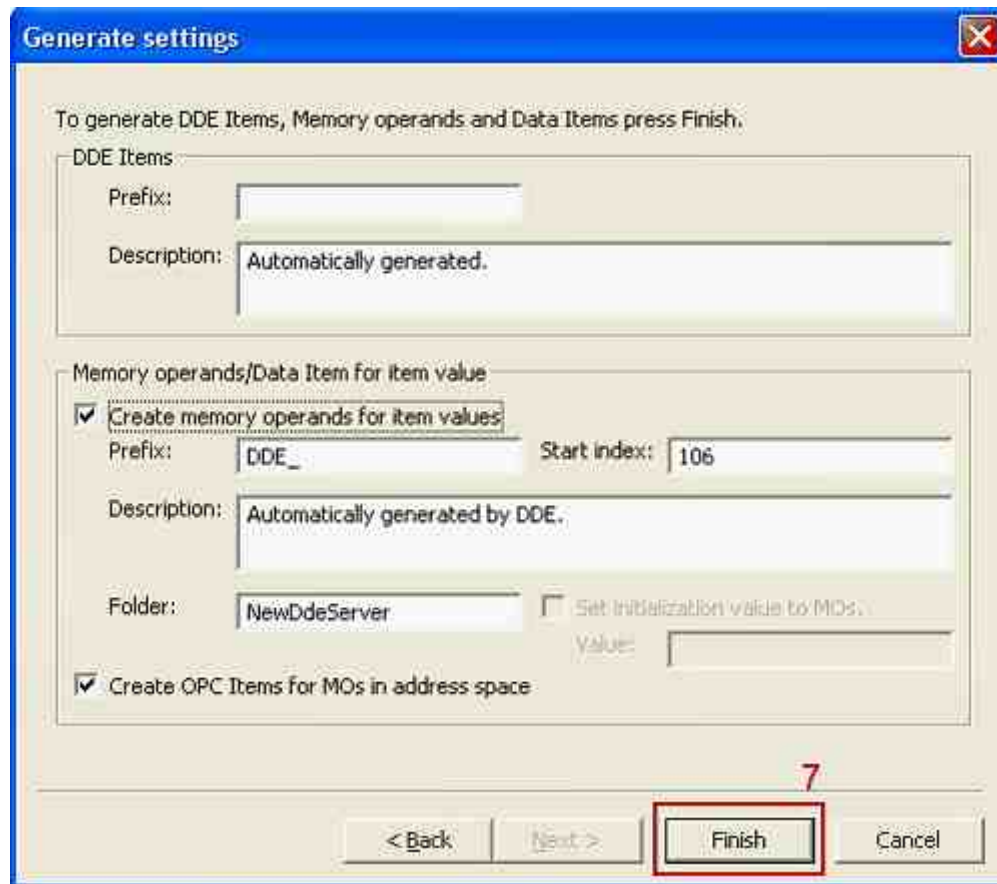
After defining DDE items by one of three possible ways as described in paragraph [Easy DDE items defining](#)^[10] the list box with all defined DDE items is shown. If you want, you can change the direction and the data type for selected items - it is Step 5 of the DDE items configuring. To go to step 6 - (described in)



7 Easy DDE items mapping

After defining DDE items as described in paragraph [Defining and mapping of DDE items](#) ¹⁰

6. Then push the "Next >" button.
7. If you want, you can change predefined values. Push the "Finish" button to generate all necessary elements (DDE Items, Memory operands and OPC Items).
4. necessary elements (DDE Items, Memory operands and OPC Items).



8 Advanced configuring

Functionality of the DDE client is implemented as standard enhancing DLL.

There are two possible ways of using the functionality implemented in enhancing DLL by the OpcDbGateway runtime. The first possibility is using of a function implemented within DLL by using *configurable command CALL DLL*. The second possibility is running of activities in DLL continuously in independent thread and providing synchronization with the OpcDbGateway runtime through shared memory operands (MO). DDE client for OpcDbgateway and SAEAUT UNIVERSAL OPC Server use both methods to be able to use full functionality of different DDE servers.

To enable [Standard behaviour](#)^[10] as e.g. using of advice loops notifications from DDE server for big amounts of DDE items the second method is used. To be able to call standard DDE client functions one by one, read/write DDE items, activate/deactivate advice loop for individual DDE items the second method is used.

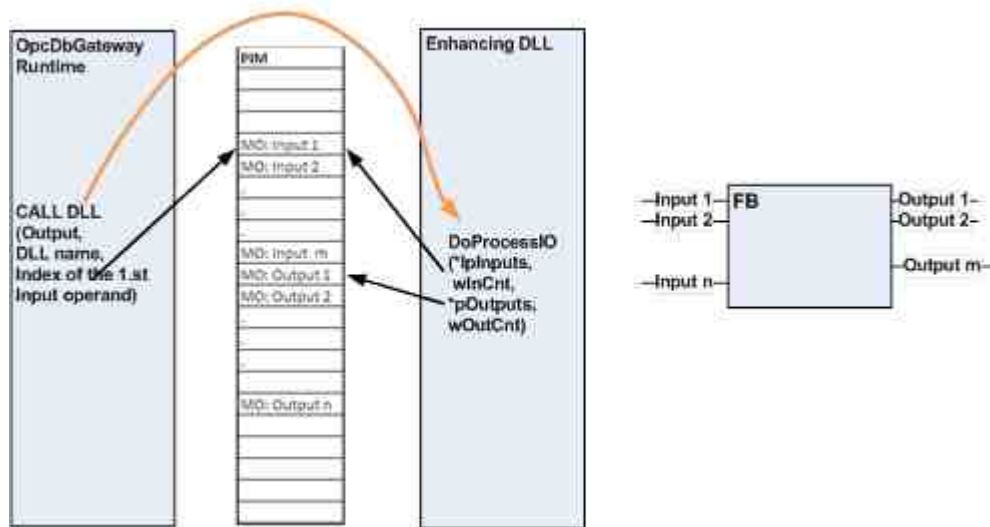


Figure 1 - cooperation with enhancing DLL using configurable command CALL DLL

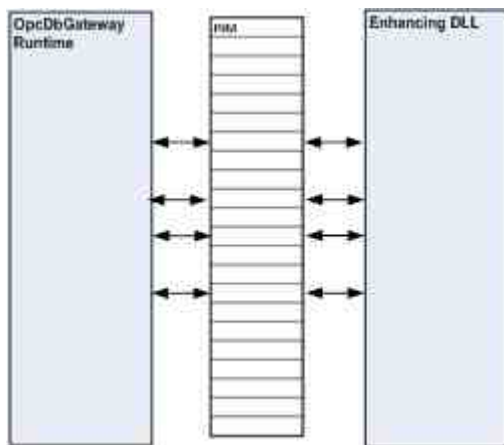


Figure 2 - coordination of the runtime functionality and enhancing DLL functionality by shared memory operands

However, both types of the functionality are provided within function blocks as described in paragraph [Function blocks](#) ^[34].

8.1 Function blocks

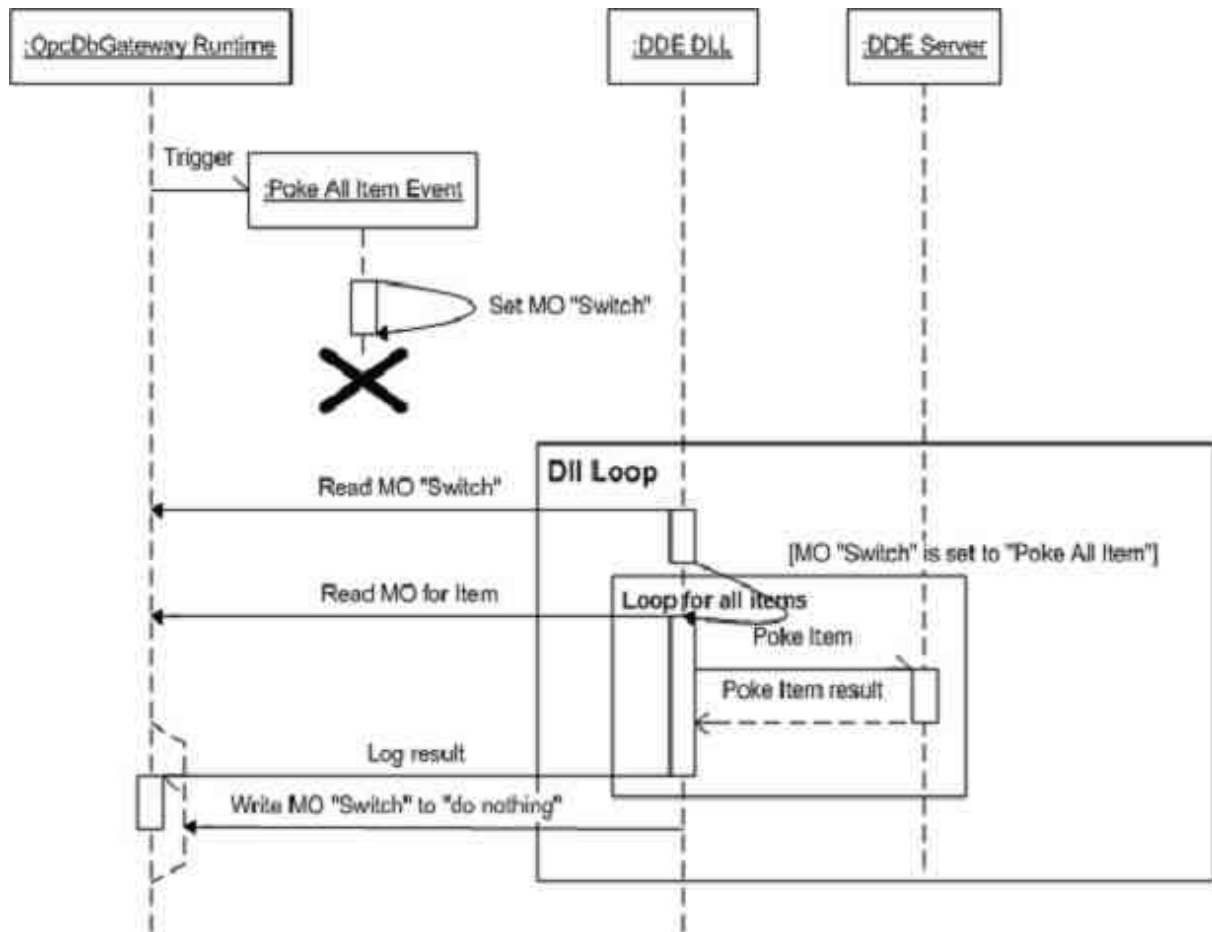
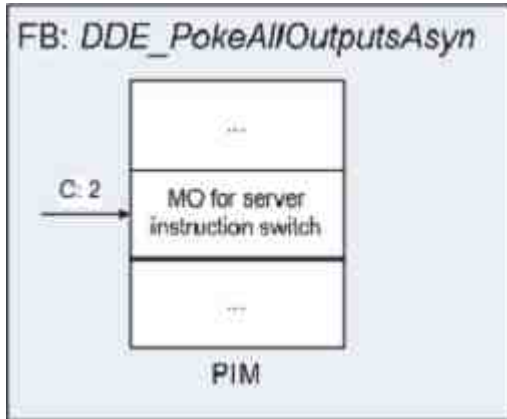
Templates for function blocks described below you can find in demo configuration [DDE test book](#) ^[47].

Function blocks consists of configurable commands. There are none DDE client specific configurable commands.

Within configurable commands, [DDE specific memory operands](#) ^[44] and [constants](#) ^[44] can be used.

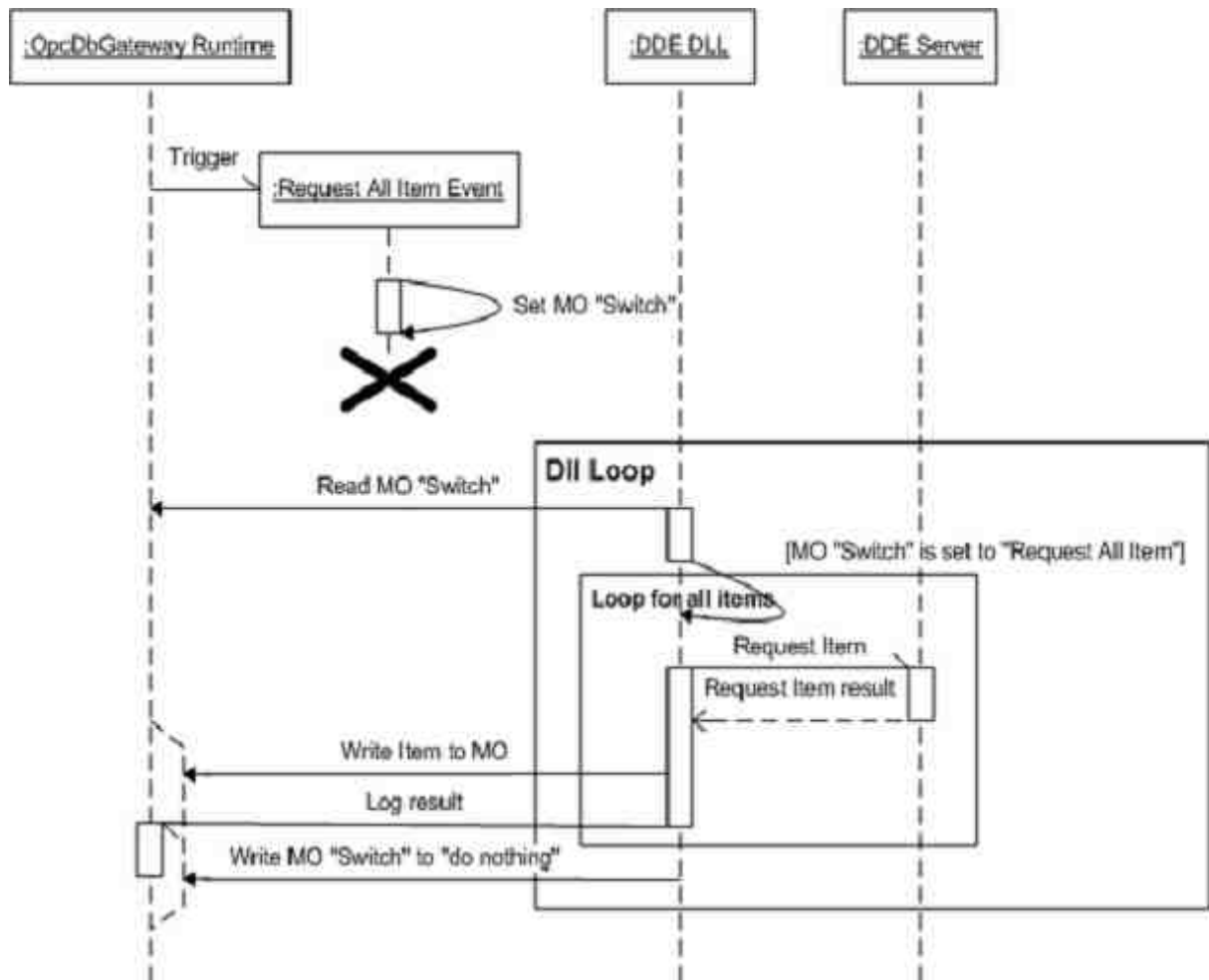
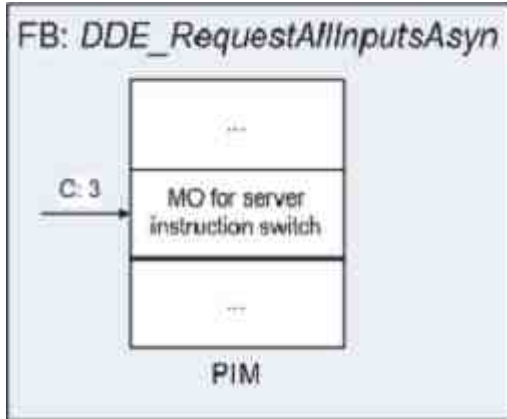
8.1.1 DDE_PokeAllOutputsAsyn

This functional block will force write all output items to DDE server. One command only changes value of memory operand "Main instruction switch" of DDE Server to constant DDE_INSTRUCTION_SWITCH_WRITE_ALL_OUTPUTS (value 2) for one time.



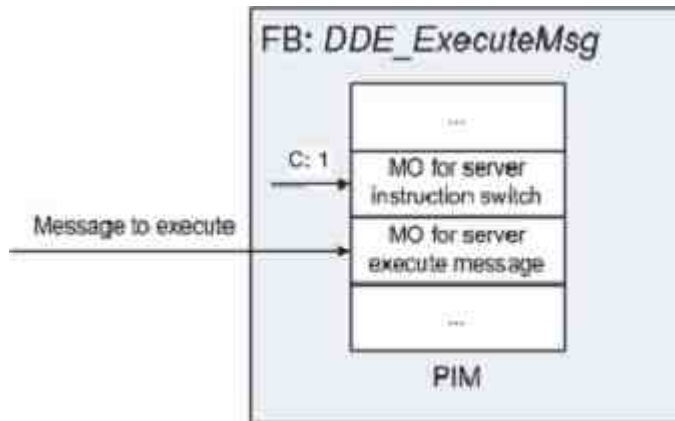
8.1.2 DDE_RequestAllInputsAsyn

This functional block will force read all input items from DDE server. One command only changes value of memory operand "Main instruction switch" of DDE Server to constant DDE_INSTRUCTION_SWITCH_READ_ACTUAL_INPUTS (value 3) for one time.



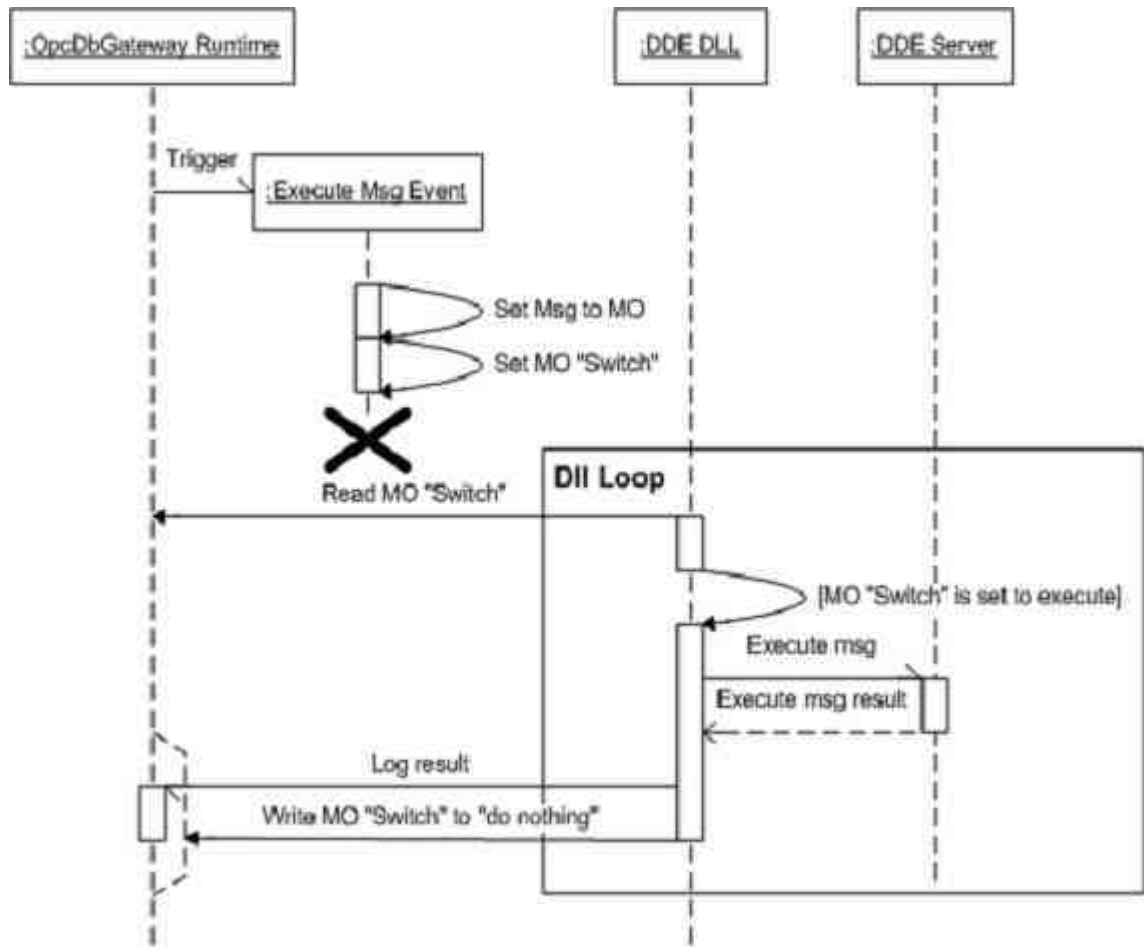
8.1.3 DDE_ExecuteMessageAsyn

Functional block "DDE_ExecuteMessageAsyn" send text command to DDE server. For example for Excel can be set message "[FONT.PROPERTIES(,"Bold")]".



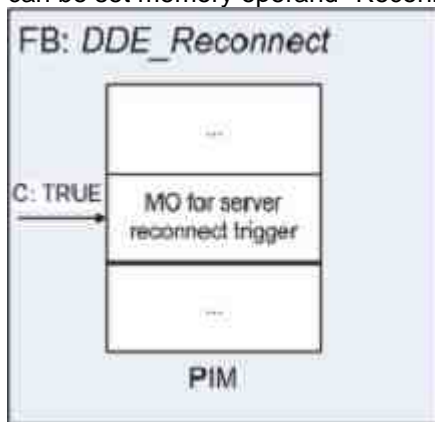
Required:

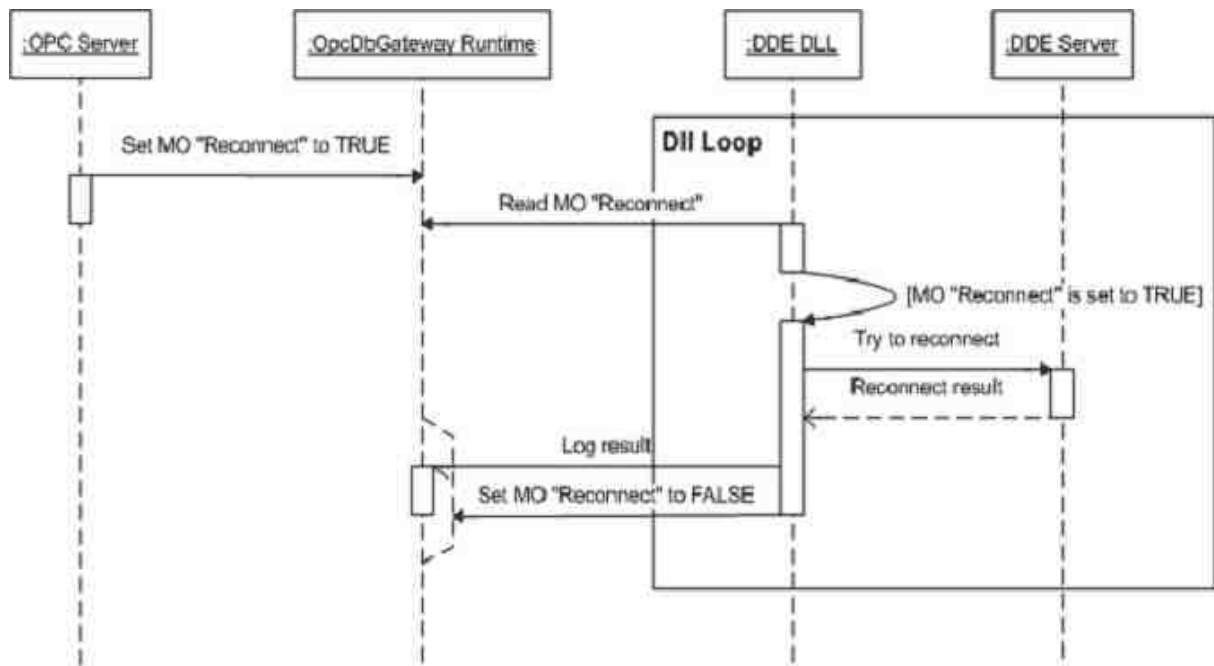
- Set text command to memory operand used for execute message on DDE server.
- Set constant `DDE_INSTRUCTION_SWITCH_EXECUTE_MSG` (value 1) to memory operand "Main instruction switch" of DDE server.



8.1.4 DDE_Reconnect

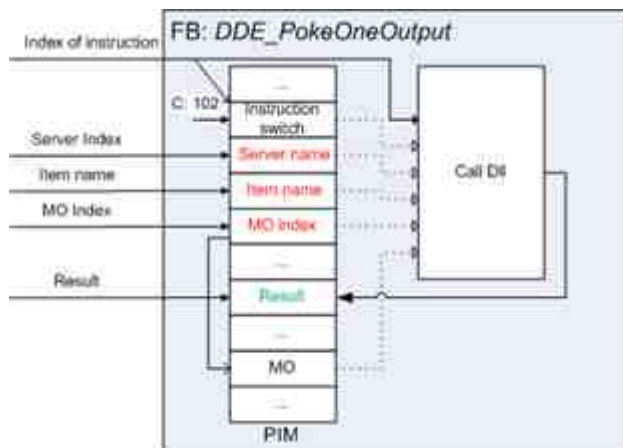
If the DDE conversation doesn't work, client can try to reconnect immediately. From external source can be set memory operand "Reconnect trigger" of DDE server to TRUE by corresponding OPC Item.





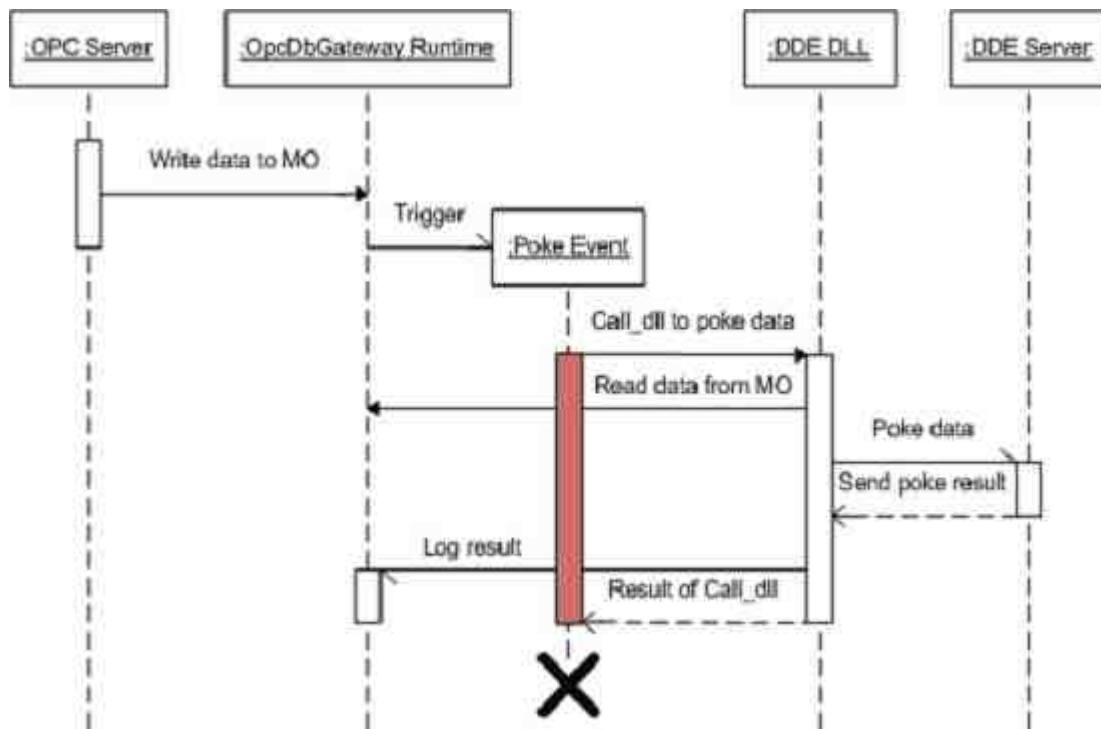
8.1.5 DDE_PokeOneOutput

You can call this functional block, if you want to write only one output item to DDE server. The output item may not be in set of output items, which are updated automatically. This is synchronous function and in result you can find success of function.



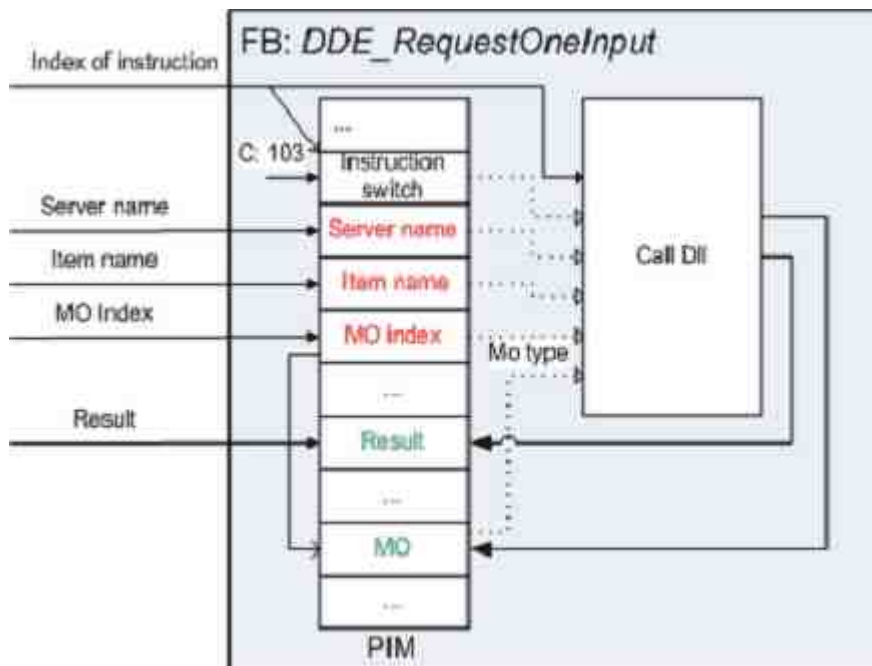
Required:

- Set four input memory operands. First memory operand can have user-defined memory address. Second, third and fourth go after it.
 - "Call_DLL_instruction_switch" (DWORD) – constant DDE_INSTRUCTION_SWITCH_WRITE_ONE_ITEM_OUTPUT (value 102).
 - "Server_name" (String) - server name in configuration to write.
 - "Item_name" (String) - item name to write.
 - "MO_index" (DWORD) - index of memory operand to write.
- Call command "Call DDE" with the first input memory operand set to "Call_DLL_instruction_switch".



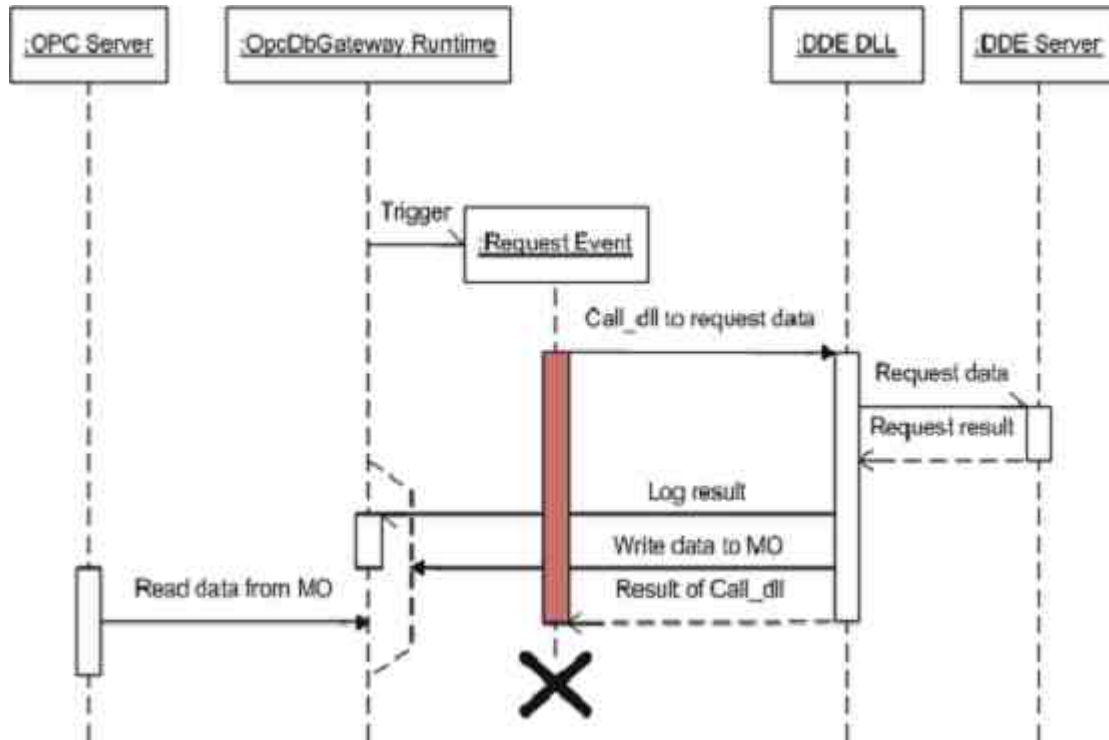
8.1.6 DDE_RequestOneInput

You can call this functional block, if you want to read one input item from DDE server. The input item may not be in set of input items, which are updated automatically. This is synchronous function and in result you can find success of function.



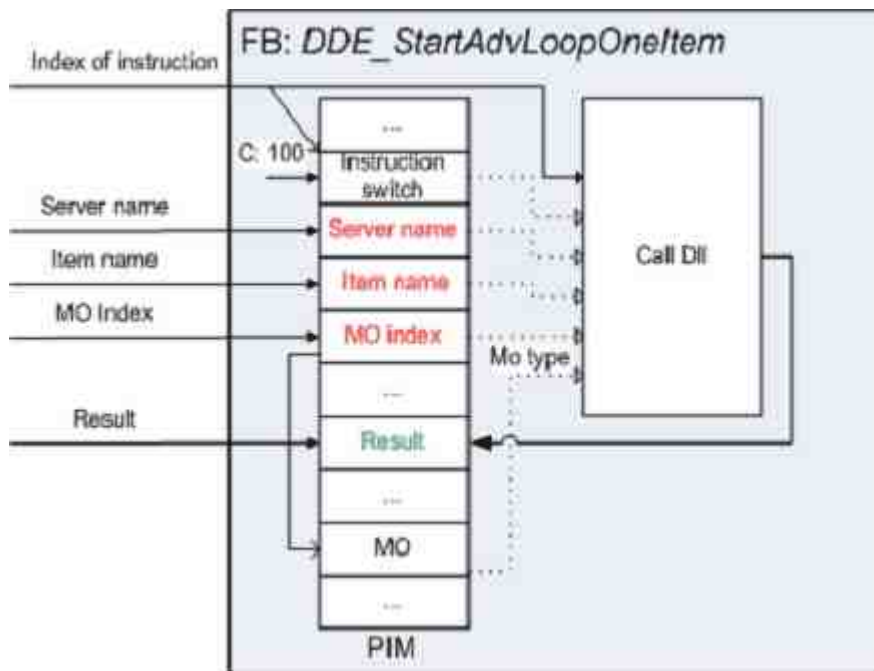
Required:

- To set four input memory operands. First memory operand can have user-defined memory address. Second, third and fourth go after it.
 - "Call_Dll_instruction_switch" (DWORD) - constant DDE_INSTRUCTION_SWITCH_READ_ONE_ITEM_INPUT (value 103).
 - "Server_name" (String) - server name in configuration to write.
 - "Item_name" (String) - item name to read.
 - "MO_index" (DWORD) - index of memory operand to store value.
- -Call command "Call Dll" with the first input memory operand set to "Call_Dll_instruction_switch".



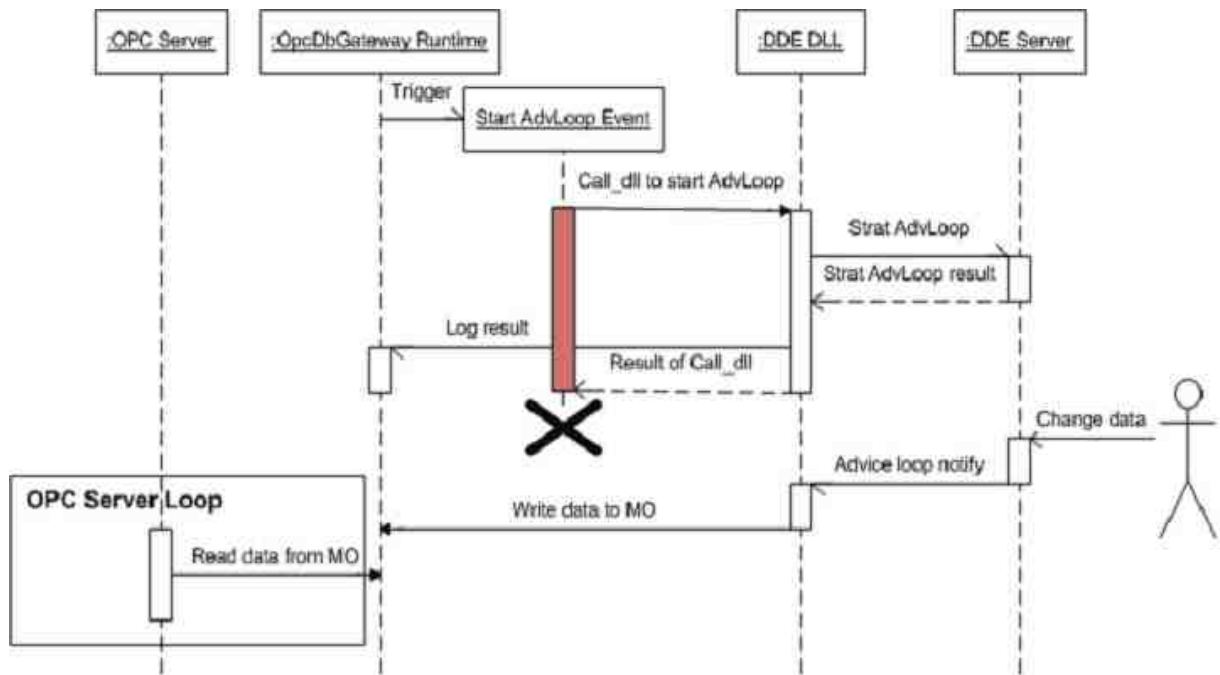
8.1.7 DDE_StartAdvLoopOneItem

You can call this functional block, if you want to start advice loop for one specific input item on DDE server. This is synchronous function and in result you can find success of function.



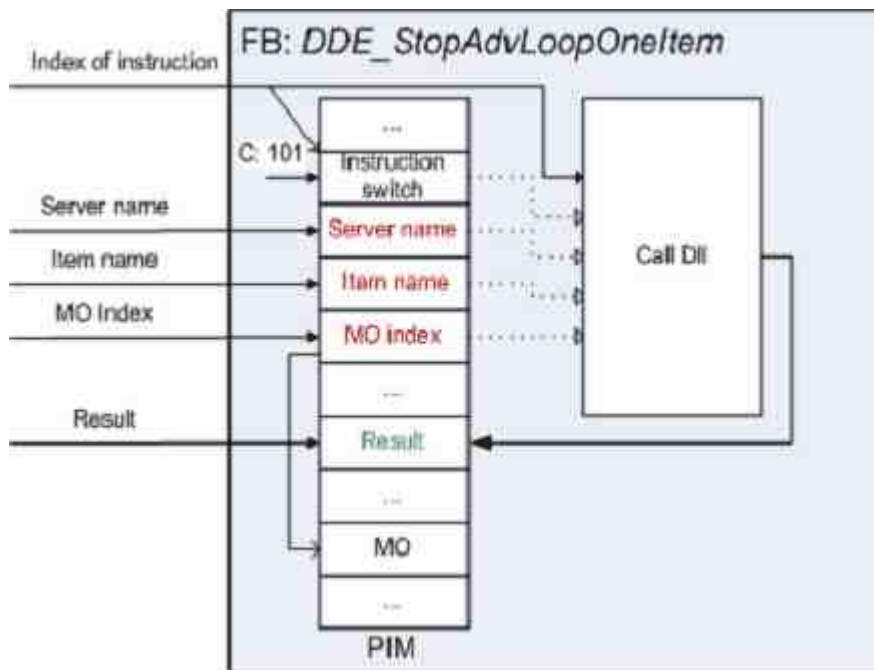
Required:

- To set four input memory operands. First memory operand can have user-defined memory address. Second, third and fourth go after it.
 - "Call_Dll_instruction_switch" (DWORD) - constant DDE_INSTRUCTION_SWITCH_START_ADV_LOOP_FOR_ONE_ITEM (value 100).
 - "Server_name" (String) - server name in configuration to write.
 - "Item_name" (String) - item name, which will be checked.
 - "MO_index" (DWORD) - index of memory operand to writing values from advice loop.
- Call command "Call Dll" with the first input memory operand set to "Call_Dll_instruction_switch".



8.1.8 DDE_StopAdvLoopOneItem

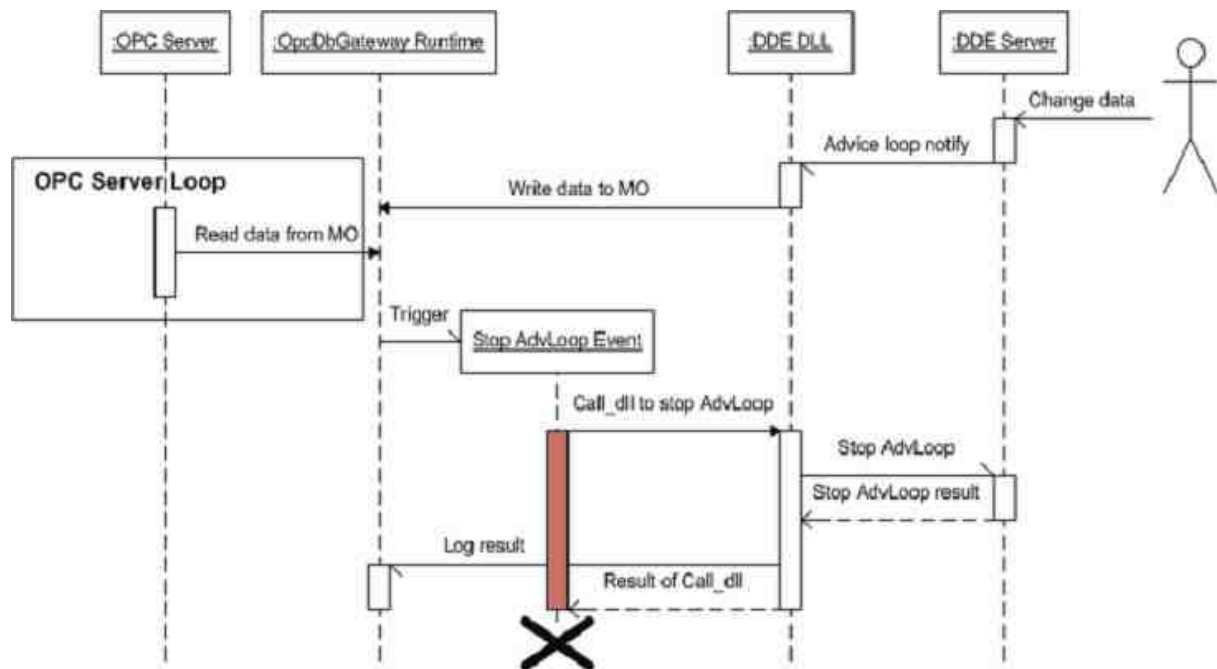
You can call this functional block, if you want to stop advice loop for one input item on DDE server. This is synchronous function and in result you can find success of function.



Required:

- To set three input memory operands. First memory operand can have user-defined memory address. Second and third go after it.

- "Call_Dll_instruction_switch" (DWORD) - constant DDE_INSTRUCTION_SWITCH_START_ADV_LOOP_FOR_ONE_ITEM (value 100).
- "Server_name" (String) - server name in configuration to write.
- "Item_name" (String) - item name, which will be checked.
- "MO_index" (DWORD) - index of memory operand to write values from advice loop.
- Call command "Call Dll" with the first input memory operand set to "Call_Dll_instruction_switch".



8.2 Memory operands for DDE servers and items

Every DDE server has a set of [memory operands to control its functionality](#)^[15].

There are also [memory operands for mapping of DDE items](#)^[32]. These can be created either by easy DDE items mapping or from [DDE item dialog box](#).^[17]

Above mentioned memory operands can be used as arguments in configurable commands.

8.3 Constants used for DDE configuring

To initialize memory operands or as arguments in configurable commands in functional blocks for DDE client driver following constant values are used. These constants you can see in the example configuration [DDE test book](#)^[47].

For asynchronous "Main instruction switch":

Constants for memory operand "Main instruction switch" of DDE server:

- DDE_INSTRUCTION_SWITCH_NOHING (value 0) - Client only does automatic functionalities. Read inputs from DDE server and write changed outputs to DDE server.
- DDE_INSTRUCTION_SWITCH_EXECUTE_MSG (value 1) - Constant to execute command.
- DDE_INSTRUCTION_SWITCH_WRITE_ALL_OUTPUTS (value 2) - Constant to write all outputs.
- DDE_INSTRUCTION_SWITCH_READ_ACTUAL_INPUTS (value 3) - Constant to read all actual

inputs.

When memory operand "Main instruction switch" of DDE server is changed, it's automatically changed to DDE_INSTRUCTION_SWITCH_NOTHING (value 0) after processing function.

For synchronous "Local instruction switch" in Call DII

- DDE_INSTRUCTION_SWITCH_START_ADV_LOOP_FOR_ONE_ITEM (value 100) - Constant to start advice loop for specific item.
- DDE_INSTRUCTION_SWITCH_STOP_ADV_LOOP_FOR_ONE_ITEM (value 101) - Constant to stop advice loop for specific item.
- DDE_INSTRUCTION_SWITCH_WRITE_ONE_ITEM_OUTPUT (value 102) - Constant to write one output item.
- DDE_INSTRUCTION_SWITCH_READ_ONE_ITEM_INPUT (value 103) - Constant to read one item input.

8.4 Logging DDE client functionality

In advanced settings of a DDE server you can set logging level for a DDE server.

Error level:

- "Inconsistent memory operand type for memory operand index %d."
- "DDE initialization fails."
- "Reading of memory operand index %d failed."
- "Internal error in DDE."
- "Not all required attributes for DDE Server found."
- "DDE connect to server "%s" failed."
- "Create DDE transaction on item "%s" failed."
- "Unsuccessful execute message "%s"."
- "DDE request item "%s" failed."
- "Send data to item "%s" on DDE server failed."
- "Inconsistent value for memory operand index %d."
- "Unsuccessful ending of DDE advice loop for item "%s"."
- "Writing to the system memory operand %d is not allowed."

Warning level:

- "Inconsistent memory operand type for memory operand index %d."
- "Execute message is empty."
- "DDE server is disconnected."
- "There is added item with same name "%s"."
- "DDE connect to server failed."
- **Informative level:**
- "DDE client connected."
- "Successful DDE transaction on item %s."
- "Successful execute message "%s"."
- "DDE advice loop for item "%s" ended successfully."

Diagnostic level:

- "Successful request item "%s"."
- "Successful send data to item "%s" on DDE server."

9 Examples

The first example "System variables" enables to read/write values of the system variables from/to memory operands from/to MS Excel sheet used as DDE server. The second one [DDE test book](#)⁴⁷ shows also advanced configuring of the DDE client driver usage.

9.1 System variables

This configuration (SsystemToExcel.ODG) is used to test the basic functionality of the DDE client in OpcDbGateway / SAEAUT UNIVERSAL OPC Server. Using it , you can monitor activity of the container application runtime in the Excel sheet „XLS server - System variables". All system memory operands of container application are displayed in the sheet. It's possible to change values of some items directly in the sheet (must be marked as „Read/Write").

Please. open the Excel document "OPC_Server.xls", before running of the OPC Server with this configuration.

	A	B	C	D	E
		Name	Value	Access	VarAddress
1		SyncQueueSize	0	Read/Write	1
2		AsyncQueueSize	0	Read/Write	2
3		TraceLog	0	Read/Write	3
4		TimeLog	0	Read/Write	4
5		PlcPeriod	1000	Read-only	5
6		PlcPeriodMeasured	1000	Read-only	6
7		PlcPeriodCounter	42	Read-only	7
8		PlcCycle	1	Read-only	8
9		PlcCycleMax	1	Read-only	9
10		PlcCycleMin	1	Read-only	10
11		PlcCycleLimitCounter	0	Read-only	11
12		PlcSystemTime	19.7.2012 12:36	Read-only	12
13		PlcCycleLimitPercent	0	Read-only	13
14		PlcUsedLogsSpace	0	Read-only	14
15		PlcUsedReportsSpace	0	Read-only	15
16		PlcAvailVirtualMemory	0	Read-only	16
17		PlcUsedProcessDBSpace	0	Read-only	17
18		PlcLogsFULL	0	Read-only	18
19		PlcReportsFULL	0	Read-only	19
20		PlcVirtualMemoryLOW	0	Read-only	20
21		PlcProcessDBFULL	0	Read-only	21
22		PlcACLLineStatus	1	Read-only	22
23		PlcBatteryFlag	128	Read-only	23
24		PlcBatteryLifePercent	255	Read-only	24
25		PlcBatteryLifeTime	-1	Read-only	25
26		PlcBatteryFullLifeTime	-1	Read-only	26
27		PlcStop	0	Read/Write	27
28		PlcRestart	0	Read/Write	28
29		PlcStatus	1	Read-only	29
30		PlcStatusDescription	Running	Read-only	30
31		PlcStopReason	0	Read-only	31
32		PlcStopReasonDescription		Read-only	32

9.2 DDE test book

This configuration is good for learning to work with functions of the DDE client in OpcDbGateway / SAEAUT UNIVERSAL OPC Server.

A necessary part of this configuration is the Excel document „XLS server - Testing of DDE“. In the document are presented all functions of DDE client.

Open the Excel document "DDETestBook.xlsx", before running of the OPC Server with this

configuration. Sheets TEST, Output1, Output2 in the document are configured as individual DDE Servers in OpcDbGateway.

1. When the DDE Client starts up, it will connect to the server (Excel sheet).

Message from the client:

2. On start the DDE Client read defined values from the DDE server to his memory. The server will notified the client, when values are changed.

Var Type:	VT_BOOL (TRUE -1; FALSE 0)	VT_I1 (-128 - 127)	VT_I2 (-32,768 - 32,767)	VT_I4 (-2,147,483,648 - 2,147,483,647)	VT_R4 (1.175494351e-38F - 3.402823466e+38F)	VT_R8 (2.2250738585072014e-308 - 1.7976931348623158e+308)	VT_UI1 (0 - 255)	VT_UI2 (0 - 65,535)	VT_UI4 (0 - 4,294,967,295)	VT_BSTR	VT_DATE
Server's values:	-1	13	14	45	-16	-2,5878000000000000E+02	155	12	568	asfsdvc	20.6.2012 8:47:18

3. The DDE client can write defined values on the server. There will be same values, that are on the server:

	-1	13	14	45	-1,6000000000000000E+01	-2,5878000000000000E+02	155	12	568	asfsdvc	20.6.2012 8:47:18
--	----	----	----	----	-------------------------	-------------------------	-----	----	-----	---------	-------------------

You can view system MOs, too. Eg. PlcPeriodCounter:

There will be random value from Internal OPC Server (0..10):

4. The DDE client can send message to execute on the server.

Write here message for the server: (FONT.PROPERTIES("Bold"))

For execute message write here "-1": 0

This event starts FBs "DDE_RequestOneInput" and "DDE_ExecuteMessageAsyn".

5. The DDE client can synchronously read items on event and write one item on defined DDE server to defined item.

Write here text to read: Something

Write here server configuration name. Eg. "TestBook_Output1, TestBook_Output2": TestBook_Output2

Write here item name. Eg. "RLC1": RLC1

For start event write here "-1": 0

This event starts FBs "DDE_RequestOneInput" and "DDE_PokeOneOutput".

6. The DDE client can force write all outputs on the DDE server.

For start event write here "-1": 0

Change some of the outputs on line 10 and then start the event!

This event starts FB "DDE_PokeAllOutputsAsyn".

7. The DDE client can force read all inputs from the DDE server.

Change some of the input values on the OPC server and then start this event!

You can not start this event from excel. When one item in excel is changed, then it send change notification (advice loop) to all items.

For start this event set OPC Item "TestBook\Triggers\Trigger_WriteAllOutputs" to TRUE.

This event starts FB "DDE_RequestAllInputsAsyn".

8. Reconnect on timeout.

Try to close the DDE server (Excel) and start it later again.

At first, be sure that in Advanced settings of DDE Server "Reconnect timeout" and "Inactivity reconnect timeout" are not 0.

9. Reconnect by external source.

Try to close the DDE server (Excel) and start it later again.

At first, set in Advanced settings of DDE Server "Reconnect timeout" and "Inactivity reconnect timeout" to 0.

Start excel. And then by external app change "TestBook\Control\ReconnectTrigger" on OPC Server to TRUE.

10. You can see the various messages from DDE in the log file.

Try to change Log Level in Advanced settings of DDE Server to see more or less messages.

Or try to make a mistake in the configuration of DDE.

Description of functionality

1. It's possible to write values to specific items on DDE Server. When the DDE client connects to the server, then it writes message, which is configured in the OPC Server, to cell C2.
2. The DDE Client can read values from defined items on DDE Server. Type to which convert value is defined in the configuration. If an user changes a value of a item on the DDE server, then it's changed a value of the memory operand corresponding to this item. You can see changes by an OPC client, where OPC items corresponding to memory operands.
3. To see conversion of input values from row 7 to defined data type, memory operands are copied to output memory operands. You can see output memory operands in row 10.

Output item can be every memory operand from OPC Server. In row 12 is system variable PlcPeriodCounter and random value.

4. One from important functionalities is executing command on the DDE Server. Commands are specific for every DDE Server. After entering the value -1 to the cell D16 is read a command from the cell D15, which is then sent to the DDE Server for execution.
5. The DDE client can work with items, which are not defined in the configuration. In functional block „ReadValue-WriteValue“ are read values from cells D20-D22 synchronous. In cell D21 should be a name of existing DDE Server in the configuration (TestBook, TestBook_Output1, TestBook_Output2). In the cell D22 is a name of an item on the DDE Server. If you put value -1 to the cell D23, then value is written by specific parameters.
6. The DDE client puts a new value to the DDE Server only if corresponding memory operand is changed. In some cases it's important to force write all defined output items to the DDE Server. You can try it, if you change same value in row 10 and then put value -1 to the cell D27. Functional block DDE_PokeAllOutputsAsyn will write all defined output items to the DDE Server.
7. Similar, it can be important to force read actual values from the DDE Server to OpcDbGateway. To test it, you need a OPC client. Change value of some OPC Item, which is defined as input DDE item a then set the trigger "TestBook\Triggers\Trigger_WriteAllOutputs" to TRUE. In the functional block DDE_RequestAllInputsAsyn will be read all input items from DDE Server to OPCDbGateway.
8. In the configurator is possible to set a time after which DDE client should try to connect again to the DDE Server, if previous connect was unsuccessful. DDE client try to reconnect if DDE Server was running, but later it turns off, too. Default time is set to 15 sec. When is set 0, no automatic reconnect is done. You can try turn on DDE Server later and wait to automatic reconnect.
9. There is another way to reconnect, if we don't want automatic. Set reconnect time to 0 sec in configuration and start OPC Server. Later start DDE Server and by OPC client set trigger "TestBook\Control\ReconnectTrigger" to TRUE. It starts functional block DDE_Reconnect, which will force reconnect.
10. All important parts in functionality are logged to the log file. For this, there is necessary trigger/event mechanism, which write message from defined memory operand to the log file. To understand see functional block „Write Universal Log Message“, trigger "Universal Log Trigger" and event "Universal Log Event".

Index

- " -

- "DDE Servers" folder 12
- "DDEreconnect" tag 5
- "Excel" as service 28
- "Main instruction switch" of DDE Server 35, 36, 37
- "Reconnect trigger" of DDE server 38

- (-

- (discoverable) DDE Servers 12

- [-

- [Book1]Sheet1 28

- A -

- activate/deactivate advice loop for individual DDE items 33
- add items 26
 - when server is offline 26
- add new items automatically 23
- address space of the internal OPC server 21
- advanced configuring features 18
- advice loop 10
- advice loops notifications 33
- arguments in configurable commands 44

- B -

- basic activities 10
- basic properties of the DDE server 23
- basics of working with container application 7
- big amounts of DDE items 33
- Both 17
- browsing DDE server 23
- Browsing of DDE server 5

- C -

- Call Dll 39, 40, 41, 43

- call standard DDE client functions one by one 33
- Call_Dll_instruction_switch 39, 40, 41, 43
- CF TEXT 5
- commands 10
- configurable command CALL DLL 33
- configuration wizard 10
- configuring wizard 28
- constants 20
- container application 5, 21
- control variables 19
- coordination of the runtime functionality and enhancing Dll functionality by shared memory operands 33

- D -

- DDE 3
- DDE client 5
- DDE client driver 3, 5
- DDE conversation 38
- DDE dll 11, 19
- DDE item 17
- DDE items 19
- DDE items are source and MO's destination 10
- DDE Runtime 11
- DDE server browsing 24
- DDE Server element 12
- DDE server to MO 17
- DDE_ExecuteMessageAsyn 37
- DDE_INSTRUCTION_SWITCH_EXECUTE_MSG 37, 44
- DDE_INSTRUCTION_SWITCH_NOHING 44
- DDE_INSTRUCTION_SWITCH_READ_ACTUAL_IN PUTS 36, 44
- DDE_INSTRUCTION_SWITCH_READ_ONE_ITEM_INPUT 44
- DDE_INSTRUCTION_SWITCH_START_ADV_LOOP_FOR_ONE_ITEM 44
- DDE_INSTRUCTION_SWITCH_STOP_ADV_LOOP_FOR_ONE_ITEM 44
- DDE_INSTRUCTION_SWITCH_WRITE_ALL_OUT PUTS 35, 44
- DDE_INSTRUCTION_SWITCH_WRITE_ONE_ITEM_OUTPUT 39, 44

Defining index ranges for DDE items 5
Diagnostic level 45
dialog box "DDE Server search" 12
dialog of a DDE server configuration 23
directory "DDE client" 7

- E -

Error level 45
event 11
events 10, 20
Excel interaction 28
execute DDE commands 5
execute message on DDE server 37

- F -

function blocks 10, 20
functional block 39, 40, 43
functions 3

- G -

Generate items 23, 24
generic DDE 3

- H -

Hot links 5
choosing of cell's range in a data sheet 5

- I -

index ranges 26
input and output variables of the data transfer 19
input item from DDE server 40
installation package of the container application 5
internal OPC server 21
Item_name 39, 41, 43

- L -

log file 10
logging functionality 22
logging of DDE client driver activity 22

- M -

Main instruction switch 44
mapped to memory operands 19
Memory Operands 3, 20
MO 3
MO to map DDE item to 17
MO_index 39, 41, 43
MO's are the source and DDE items destination 10
MS Excel as DDE server 7
multi select 28

- N -

Name to access to DDE item on DDE server 17

- O -

OPC server 5
Output - from MO to DDE server 17
output items 39

- P -

poke loop 5
prefix and indexes of items 26
Process Image Memory 3
Program start menu 7

- R -

read/write DDE items 33
reading of individual item from DDE server 5
reconnect 5
relationships between elements of the DDE server 23
request loop approach 5
request/response loop approach 5
resolution of DDE communication failures 5
running of activities in DLL continuously in independent thread 33

- S -

select items directly 23

send text command to DDE server 37
Server_name 39, 41, 43
service and a topic name of the DDE server 24
service name 12
set logging level for a DDE server 45
source/destination data points 21
sources or destinations for data transfers with DDE servers 21
special functionality for Excel 23
specific applications 19
standard enhancing dll 18
standard function blocks 5
synchronous function 39, 40, 43
system memory operands 11

- T -

TopicItemList 23
trigger 11
triggers 10, 20

- U -

use a formula 23
user-defined memory address 39

- W -

Warning level 45
ways to add items 28
writing DDE item values from/to a DDE server if it is on line 17
writing of individual item to DDE server 5