OFS Parameter Tuning

Training Manual

OFS v3.40

Schneider Electric

DISCLAIMER

Schneider Electric Limited makes no representations or warranties with respect to this manual and, to the maximum extent permitted by law, expressly limits its liability for breach of any warranty that may be implied to the replacement of this manual with another. Furthermore, Schneider Electric Limited reserves the right to revise this publication at any time without incurring an obligation to notify any person of the revision.

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric Limited nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information that is contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric Limited software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

OFS Parameter Tuning Training Manual

INTRODUCTION AND LEGAL NOTICE

Your purchase of this official OFS Parameter Tuning Training Manual entitles you to undertake the OFS Parameter Tuning training course.

Satisfactory completion of the course evaluation is mandatory for you to obtain a Schneider Electric Limited certificate of completion of the training course.

Schneider Electric Limited will not accept any liability for action taken in reliance on this training manual.

TRADEMARKS (OFS Server)

Schneider Electric Limited has made every effort to supply trademark information about company names, products and services mentioned in this manual. Trademarks shown below were derived from various sources.

OFS Server is a trademark owned by Schneider Electric or its affiliated companies. All other trademarks are the property of their respective owners.

Microsoft Windows®, Windows® XP, Windows® 7 and Excel are either registered trademarks or trademarks of Microsoft® Corporation in the United States and/or other countries.

General Notice:

Some product names used in this manual are used for identification purposes only and may be trademarks of their respective companies.

Validity Note

The present documentation is intended for qualified technical personnel responsible for the implementation, operation and maintenance of the products described. It contains information necessary for the proper use of the products.

About Us

Members of Schneider Electric's team of Instructional Designers have tertiary qualifications in Education, Educational Course Development and are also experienced Instructors. Currently, the team is supporting a range of Schneider Electric courses in multiple languages and multiple software environments.

Authors

Antony Saunders

August 2015 Edition for OFS v3.40
Manual Release 1.0

Before the Course Begins

Course Instructor

Please write the name of the Instructor here: ☞ _____

The course Instructor will be spending the next One Day with the class, and will guide the students through this training course. The Instructor is an experienced OFS Server user so please feel free to ask questions.

Scope of this Training Manual

This training manual is a supplement to the authorised training. In order to make proper use of the software students should also refer to the documentation that has been provided with the product such as the Help Files, User Guides or Knowledge Base.

The graphics displaying screen captures were taken in Windows® XP and Windows® 7 operating system(s) using Classic mode display properties. If students are running a different version of Windows® then screen images may differ slightly from those shown in the training manual.

Some screen captures may have been taken from beta versions of the software and may vary slightly from release screen captures.

This document is intended to provide guidelines and practical tips to analyze the performance of an existing system, and then configure some of the OFS parameters to optimise the overall performance.

From a system perspective, it is difficult to provide any figures on this improved level of performance, as the number of options and possible configurations on the PLC side are numerous.

This training manual does not include any aspects on the use or configuration of the **OFS Configuration Tool** which has a User Guide included within on-line Help.

Course Overview

Course Objectives

By the completion of this training course the student will be able to:

- Understand OFS Server architecture
- Know how to use OFS Server
- Be able to configure aspects of OFS Server to enhance interfacing
- Use on-line changes

Target Audience

The OFS Parameter Tuning training course is an integral part of the complete Schneider Electric curriculum. This specific course is designed for:

- Users who are new to OFS Server
- Users who have no previous OFS Server programming experience
- Experienced programmers who are new to the SoCollaborative tools as a guide to the differences between these and other OPC interfaces

Further Training:

This course is designed specifically for users with no previous OFS Server programming experience; therefore, the content is set at a level that is most suited to inexperienced programmers.

Experienced OFS Server programmers may find they are able to complete the content in this course before the designated duration of One Day.

Enquire with the Instructor with regards to further customised OFS Server training.

Prerequisite Knowledge

It is expected that trainees will:

- Be familiar with the concepts of PLCs
- Be familiar with the concepts of Industrial Automation
- Be familiar with the concept of Ethernet Networking
- Be familiar with Microsoft Windows®

Course Overview (cont.)

Course Program

The training course will take One Day to complete. The following program outlines the topics that will be covered:

Day	Topics
1	➢ What is OFS? ➢ OFS Network Architecture ➢ OFS Network Configuration ➢ OFS Client Configuration ➢ OFS Client Performance

Support

If support or additional information about any concepts or products in the course is required, students should ask the Instructor who will either address the question or obtain additional technical assistance as required.

Conventions Used in this Manual

Objectives

These are the skills to be achieved by the end of each chapter. An overview providing a brief synopsis of the topic begins each section. Often, examples are given to illustrate the conceptual overview.

Example -

The configuration environment consists of several toolbars, browser windows and programming editors. This chapter introduces the user to the configuration environment using an example project with pre-defined elements.

This Chapter Covers These Topics:

Exercises

After a concept is explained students will be given exercises that practice the skills just learned. These exercises begin by explaining the general concept of each exercise and then step-by-step procedures are listed to guide students through each exercise.

Example -

Paste an object from a library onto a test page called **Utility**.

1 Run the Milk_Upgrade project then trigger and view some alarms.

i. Use the following template settings:

User Input

Whenever information is to be typed into a field or dialog box it will be written in this font:

```
KETTLE_TEMP/25
```

Note that some exercises will show a fragment of information already typed into a OFS Server screen and then ask students to add extra lines of configuration. In this instance, the previously entered material will be given to the student as light grey italic text.

KETTLE_TEMP/25

```
OVEN_TEMP/5
```

Conventions Used in this Manual (cont.)

Hints & Tips

This heading will provide students with useful or helpful information that will make configuring the project easier.

Example -

Hints & Tips:

To go to the next field, use the mouse cursor or press the **TAB** key.

Note

A note will refer to a feature which may not be obvious at first glance but something that should always be kept in mind.

Example -

Note:

Any events named **GLOBAL** are enabled automatically when events are enabled.

Menus and Menu Options

Text separated by the double arrow symbol "**»**" indicates that students are to select a menu.

Example -

File » New…

Open a menu "**File**" then select the menu option "**New…**"

Horizontal and Vertical Tabs

Text written this way indicates the **Horizontal** then the **(Vertical)** tab is to be selected.

Example -

Appearance (General)

Conventions Used in this Manual (cont.)

See Also

Text written in this way indicates further references about the current topic.

Example -

 **See Also:**

For further information about **Templates,** see O*FS Server Help - Using Page Templates*.

Further Training

This heading describes topics that are covered in more advanced courses.

Example -

 **Further Training:**

Trend Table Maths is a topic in the **Customisation and Design Course**.

Table of Contents

Table of Instructions

Chapter 1: OFS Parameter Tuning

Overview

Introduction

This manual is intended to provide guidelines and practical tips to help users configure some of the OFS Server parameters on an existing system, based on simple features or programming rules, to optimise system interfaces.

These features are independent of each other, but are best combined together to improve system efficiency depending on the needs of each user.

Most of the items within this document apply to OFS Server software platforms. It will be mentioned when it is not the case.

Chapter Objectives

By the completion of this chapter you will be able to:

- Understand the function of OFS Server and OFS Client
- Configure aspects of both in order to improve interface communications on an existing system

This Chapter Covers These Topics:

What is OFS?

OFS Definition

OFS is an acronym for **O**PC **F**actory **S**erver, where **OPC** is also an acronym for **O**pen **P**latform **C**ommunications.

- **OPC** was originally called **O**LE (**O**bject **Li**nking and **E**mbedding) for **P**rocess **C**ontrol. However, the OPC Foundation renamed it in 2011 as the applications of OPC technology expanded into non Process Control markets and beyond its original OLE platform into other data transportation methods.
- **OFS** (**O**PC **F**actory **S**erver) is Schneider Electrics' implementation of OPC for automated systems.

OFS Uses and Functions

Schneider Electric's OPC Factory Server (OFS) software allows 'client' software applications, such as SCADA/Supervisors and customised interfaces, to access the data of Schneider Electric automation systems and electrical distribution devices connected to networks or fieldbuses, in real time.

It also enables communication to third-party devices supporting Modbus and Modbus/TCP protocols.

It is a component part of the SoCollaborative software suite which also includes **UnityPro** and **VijeoCitect** under the banner of **PlantStruxure.**

OFS v3.40 integrates the following:

- **OPC-DA** (OPC Data Access)
- **.NET API** Interface
- **OPC XML-DA v1.0** (OPC XML Data Access)
- **OPC-UA** (OPC Unified Architecture)

There are two license package options for OFS Server:

- **OFS Small:** Data server for up to 1000 items supporting the OPC-DA and OPC-UA protocols only
- **OFS Large:** Data server with an unlimited number of items supporting the OPC-DA, OPC XML-DA and OPC-UA protocols

What is OFS? (cont.)

OFS Supported Devices and Protocols

OFS Server software is a multi-device data server which allows simultaneous use of several communication protocols.

It provides client applications with a set of services for accessing the following control system items that may be local or remote:

Supported Devices:

- Modicon Quantum, Premium, M340, M580, Micro, Compact and Momentum PLC's
- TSX Series 7 and April Series 1000 PLC's
- Modbus serial devices connected via Schneider Electric gateways : TSXETG10**, EGX** ranges
- Uni-Telway serial devices connected via Schneider Electric gateways (TSXETG1010)

Supported Networks and Protocols:

- **Modbus:** Modbus serial, Modbus Plus and Modbus TCP/IP
- **X-Way/Uni-TE:** Uni-Telway, Flipway, ISAway and PCIway

OFS Access Modes

The **OFS Server** software allows for four access modes:

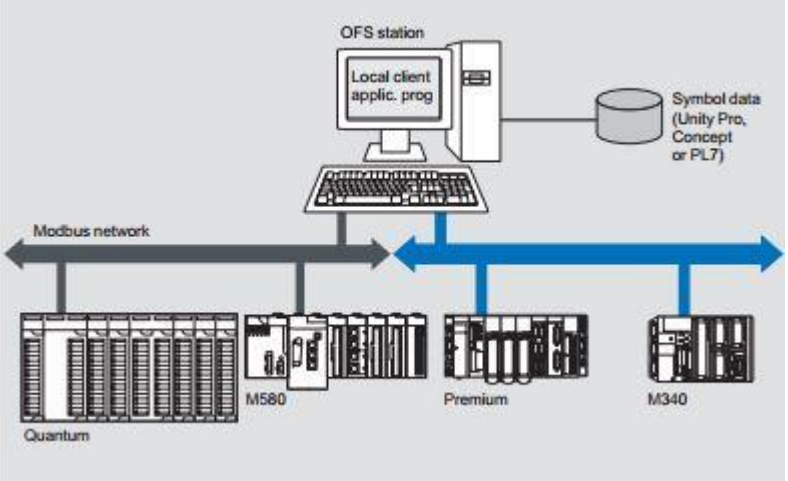
- **Local Mode**
- **Remote Access** from an **OPC-DA** client
- **Remote Access** from an **OPC.NET** client
- **Remote Access** from an **OPC XML-DA** or **OPC-UA** client

The following sections explain each mode and their corresponding network architectures:

OFS Network Architectures

OFS Local Mode

The simplest method of using OFS Server to retrieve OPC data from various sources is to place these sources on the same network as the OFS Server and run the OFS Client application from within the main OFS Station.



The picture above shows a range of Schneider Electric PLCs acting as the OPC data sources for the local OFS Station running the OFS Client application.

As fieldbuses such as Modbus and X-way are local physical connections then data sources on these networks can be read in Local Mode.

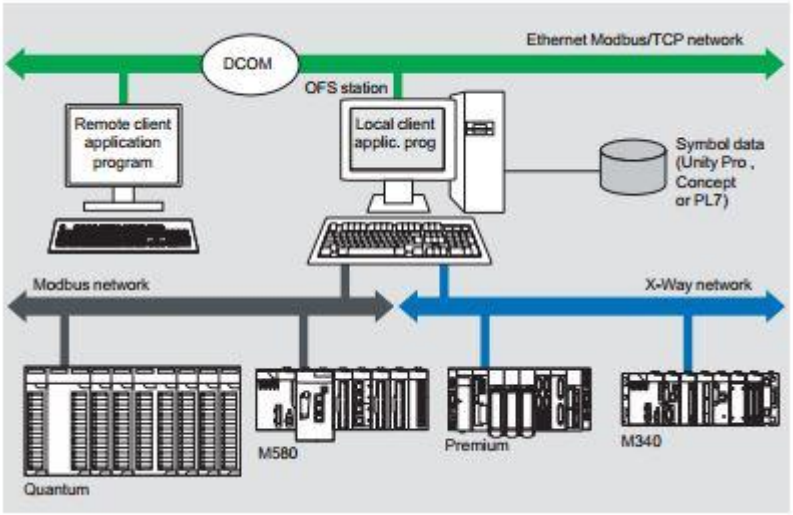
Note:

A simple explanation would be that **OFS Local Mode** is used for direct connections only.

OFS Network Architecture (cont.)

OFS Remote Access via OPC-DA

A more common method of OPC data retrieval using OFS Server is when the sources are on a remote network from the OFS Server and a remote OFS Client application is run as well the local OFS Client application from within the main OFS Station.



The picture above shows a range of Schneider Electric PLCs acting as the OPC data sources for the Local OFS Station running the Local OFS Client application and a Remote OFS Client gathering data from another source.

This remote client communicates to the OFS Server via the DCOM layer (Microsoft) of an Ethernet Modbus TCP/IP Network using the OPC-DA protocol.

Note:

A simple explanation would be that **OFS-DA Remote Mode** is used for single and simple remote connections only.

OFS Network Architecture (cont.)

OFS Remote Access via OPC.NET

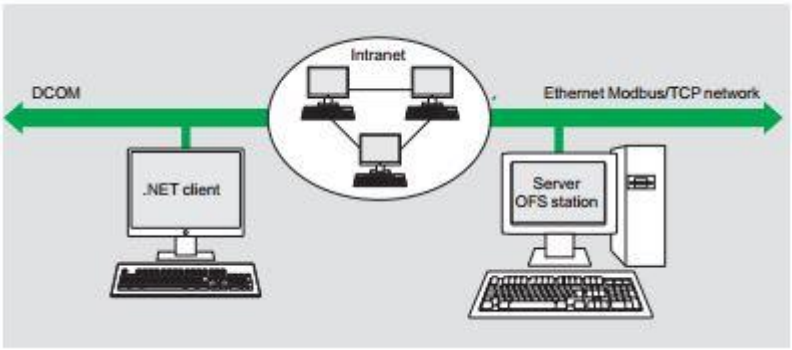
Gathering OPC data via OPC-DA protocol through the DCOM layer of Ethernet is possible over a simple remote connection. However, this method of retrieval becomes more difficult as the network becomes more complex and diverse.

For instance data is often required to be recovered through an existing business Intranet or Local Area Network (LAN).

Using a remote OPC.NET client enables the use of the OPC-DA protocol through the DCOM layer of the Intranet.

The .NET (Microsoft) compatibility of the OFS Server has been developed to allow an OPC.NET client to access OFS Server items on an Intranet network via the OPC .NET API interface.

This interface provides interoperability between existing OPC applications and applications developed in the standard .NET environment.



The picture above shows a .NET Client connecting through an Intranet (LAN) to an OFS Server Station.

This remote .NET client communicates to the OFS Server on an Ethernet TCP/IP network via the DCOM layer (Microsoft) of the Intranet using the OPC-DA protocol.

Note:

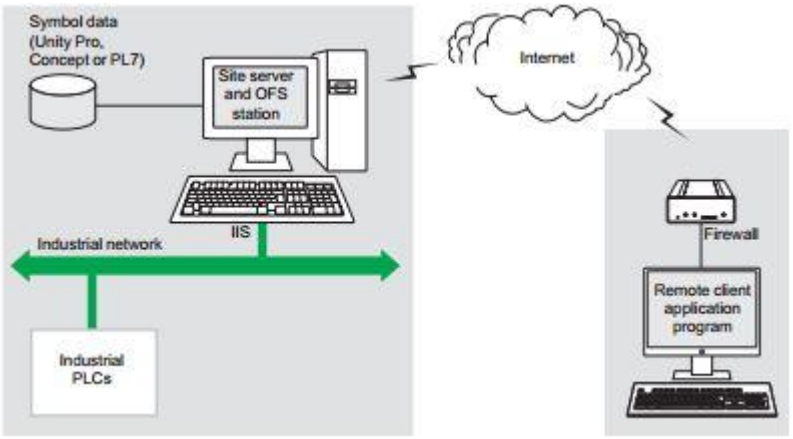
A simple explanation would be that **OFS.NET Remote Mode** is used for more complex physical remote connections (Intranet/LAN's).

OFS Network Architecture (cont.)

OFS Remote Access via OPC XML-DA

The final scale of OPC data retrieval is that required over Wide Area Networks (WAN's) or the Internet.

Using a remote SOAP (*) client enables the use of the OPC XML-DA (*) protocol over the Internet, where the client application program and the OFS Server are both remote stations.



This above picture shows the use of the SOAP (*) protocol to communicate via the Internet in conformity with the OPC XML-DA or OPC UA specifications (*) of the OPC Foundation.

These two protocols are designed to overcome the limitations of COM/DCOM connections, thus providing:

- A standardised interface for Windows and non-Windows client applications
- Remote access via the Internet through firewalls, beyond the Intranet perimeter

Note:

A simple explanation of this would be that **OFS XML-DA Remote Mode** is used for wide area complex remote connections (Internet/WAN's).

OFS Network Architecture (cont.)

OPC XML-DA (cont.)

The OPC XML-DA specification is based on Web Services standards such as SOAP, XML and WSDL (*).

* **SOAP**: **S**imple **Ob**ject **A**ccess **P**rotocol

XML: e**X**tensible **M**arkup **L**anguage

WSDL: **W**eb **S**ervices **D**escription **L**anguage

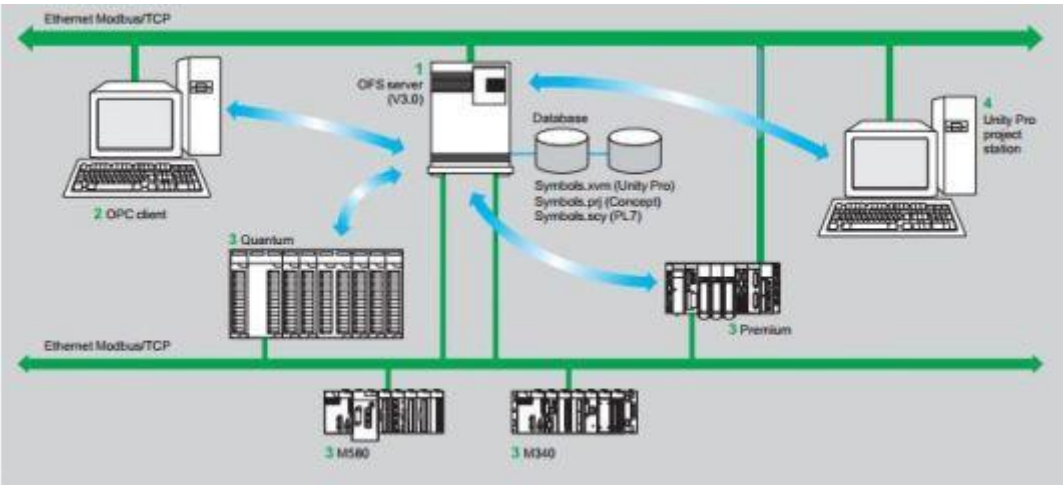
Note:

OPC XML-DA is only available with the **Large** licensed version of OFS Server.

OFS Network Configuration

OFS Server Network

The picture below shows the component parts of a simple physical network with a typical **OFS Server** retrieving **OPC** data.



The **OFS Server (1)** is at the centre of the data exchange. It makes variables exchanged between the **OFS Client (2)** and the **PLCs (3)** in one of three ways:

1 The variables database is either the **Unity Pro Project** [Symbols.xvm] **(4)**, or the **Concept Project** [Symbols.prj]. In both these cases, Unity Pro or Concept needs to installed on the **OFS Server Station**

2 The variables database is an export file [.scy for PL7, .xvm for Unity Pro]. In these cases neither PL7 or Unity Pro is required to be installed on to the **OFS Server Station**

3 Or the variables database is the **PLC** itself. In this case neither **Unity Pro** nor an export file is required. This does not however apply to **Momentum** and **TSX Micro PLCs**.

Hints & Tips

For point 3: If an inconsistency is detected (for example, following online modification of the PLC program), then **OFS** resynchronises itself as a background task, without breaking communications between the **PLC** and the **OFS Client**.

Note:

For this function the following minimum versions are required:

OFS v3.35

Unity Pro v6.0

Modicon M340 v2.3, **M580** v1.0, **Premium** v2.9 and **Quantum** v3.0 PLCs

OFS Network Configuration (cont.)

Target Ethernet Modules

The first step towards configuring an **OPC** interface is to configure the **Ethernet Network Module (NOE)** of the data targets i.e. The **PLC's.**

The following section highlights the need to do this for a **140NOE771xx** communication module incorporated into a **Quantum PLC**.

Quantum 140NOE771xx Communication Module

Before configuring the **OFS Server** software, any PLC Ethernet Communication Modules that are to act as data sources need to be configured to the best of their capabilities. The firmware of the CPU and of the Ethernet module is also required to be updated to the latest versions.

This is particularly important with the 140NOE771xx communication module used in Quantum PLCs, as communication performances have recently been improved in latest versions.

Hints & Tips

For example, **v4.7** of the **140NOE771xx** accepts packets of up to 1024 bytes instead of only 256 in previous versions. Hence, this version also allows for 12 concurrent requests instead of only 4 previously.

Upgrading the NOE to this version of firmware therefore increases the theoretical Ethernet port throughput by 12 times.

Note:

This is only true with **>v2.80** of the **CPU** module.

OFS Client Configuration

How To: Launch OFS Client and Server

- The following steps explain how to Launch **OFS Server** with an **OFS Client** using aliases already created.

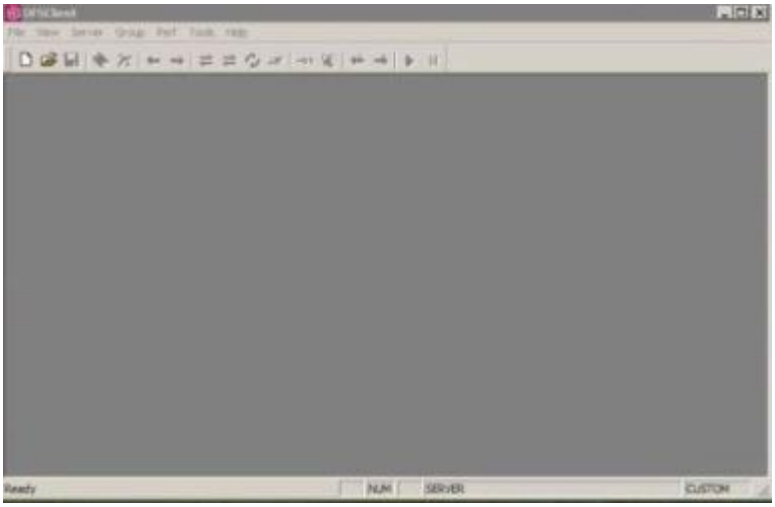
OFS Server is launched automatically when **OFS Client** is started. Hence the following procedure only explains how to launch **OFS Client.**

---

1 **OFS Clients** are started from this desktop shortcut 

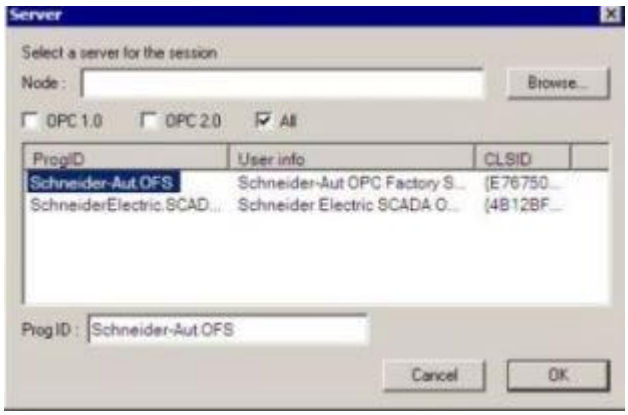
OR from the following location in **Windows XP:**
`Start»All_Programs»Schneider Electric»SoCollaborative»OFS»OFS_Testing Clients»OFS_Client`.

2 The following **OFS Client** Window then appears:



---

3 The following **OFS Server** Window also appears, layered over the **OFS Client** Window**:**



OFS Client Configuration (cont.)

Launching OFS (cont.)

4 In the **Node** field enter the name, or **Browse** for, the computer that hosts the **OFS Server**

Note:

This field can be left blank if the **OFS Server** is on the same computer as the **OFS Client**.

5 If it hasn't already been selected; **Select** the **Schneider-Aut.OFS** Server for **Prog ID**

6 And tick '**All**' for **OPC** version support

7 Click **OK** to connect the **OFS Server**

OFS Client Configuration (cont.)

Launching OFS (cont.)

8 The **OFS Server Status** Window then appears:

The **Status** field near the bottom of this window should read **Running.**

Note:

If any changes are made in the **OFS Configuration Tool** then **OFS Server** must be restarted before the modifications are taken into account.

OFS Client Performance

Task 1: Set and Test OFS Request Generation

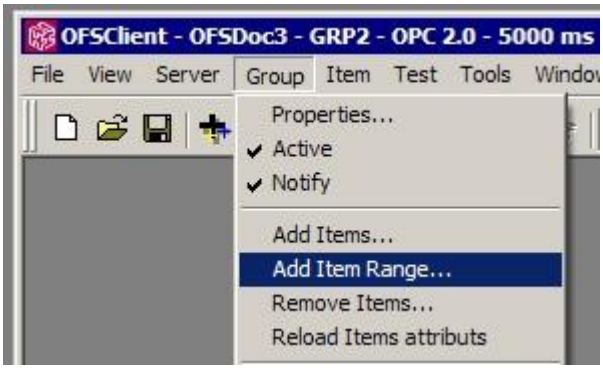
➢ The following procedure explains how to **Set** the **OFS Request Generation** and **Test** it to have the correct number of requests to efficiently scan a **PLC.**

Hints & Tips

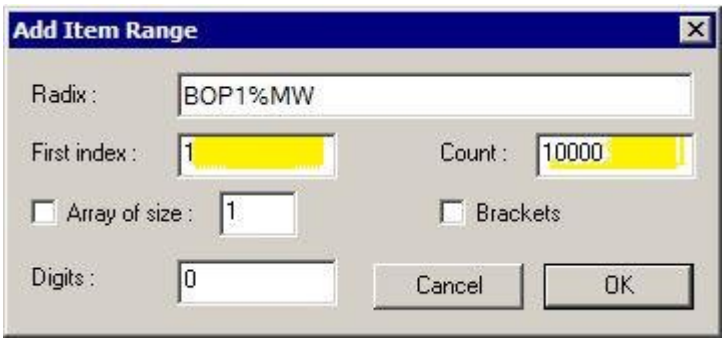
The following **Tasks 1, 2 & 3** should be carried out in order as configuration data is cascaded through each task.

1 Run **OFS Client** and add as many items as are needed in the SCADA project

By adding **Item Range** as **%MW1 to %MW10000.**



➢ **Select: `Group » Add Item Range...`**

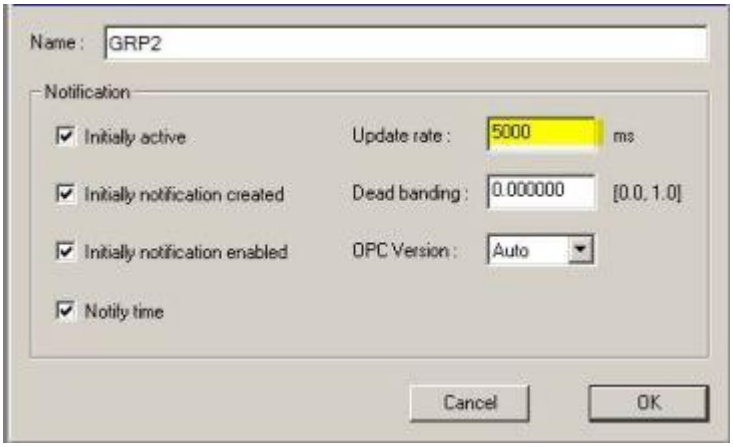


➢ **First index: `1`** **Count: `1000`**

Task 1: OFS Request Generation (cont.)

Set and Test (cont.)

1 Configure a **New** or existing **Group** with a slow **OFS Client Group Update Rate** of **5 Seconds** (e.g. **5000ms**).



- **Update rate:** `5000`

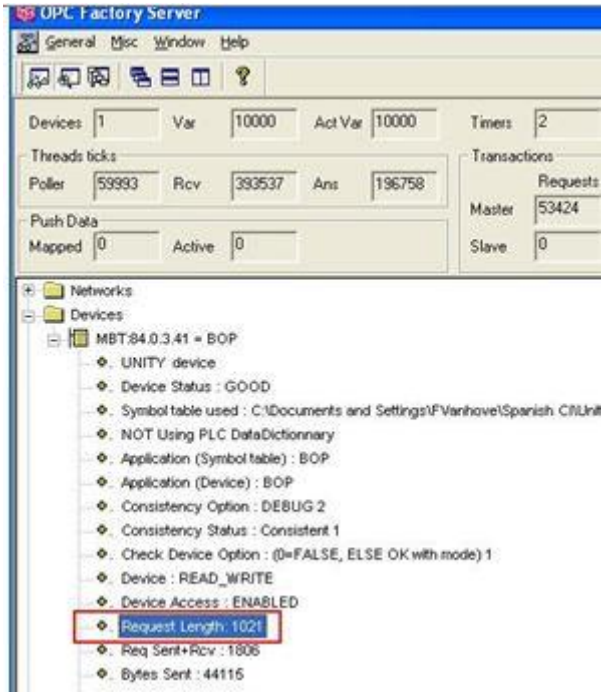
1 Open the **Network Window** in **OFS Server**.

- **Select:** `General » Network Window`

Task 1: OFS Request Generation (cont.)

Set and Test (cont.)

1 Review the **Request Length** supported by the nominated **Target Ethernet** port.



- **Request Length: = 1024 bytes**

Task 1: OFS Request Generation (cont.)

Set and Test (cont.)

1 Also review the number of requests required to read all of the subscribed OPC items = **Nb Varman Req**.

Nb Varman Req. is found by expanding **Freq Desc Nb** and then the **Group Update Rate**, which in this example is **5000ms**.

- **Freq Desc Nb » 5000ms » Nb Varman Req**.

The statistics shown above in the **OFS Server Network Window** for **Group 2** with the update rate of **5000ms** are as follows:

Request Length = `1021 bytes` *****

Nb Varman Req. = `20`

Note:

* This will only be correct, if the **NOE** firmware is upgraded to **v4.7**.

The further **Tasks** in this section will confirm if these statistics show that the communication port on the **PLC** is correctly configured or not.

They will also enable a calculation to be made of how quickly **OFS Server** will theoretically be able to refresh the entire **OPC** group containing all of the desired items.

Task 1: OFS Request Generation (cont.)

Set and Test (cont.)

Note:

In the test case above the **SCADA** project has about **10,000** items.

Hence, adding **%MW1 to %MW10000** in **OFS Client**, with a **Group Update Rate** of **5 seconds (5000ms)**, is required.

Hints & Tips

It should also be remembered that in a real **SCADA** project, the number of requests **[Nb Varman Req.]** is likely to be significantly higher.

This is because the required addresses will not be contiguous.

Hence **OFS Server** will need to be set to send more requests.

OFS Client Performance (cont.)

Task 2: Test and Set OFS Group Scan Time

- The following procedure explains how to **Test** the **OFS Group Scan Time** and then **Set** it to the most efficient scan time for a **PLC.**

1 Allow the currently open **OFS Client** to run for a few minutes**.**

2 Then check the **Average Access Time** of the **Target Ethernet Module** in the **OFS Server Network Window.**

The **Average Access Time** is found further down the list of parameters under the **Request Length**:

Hints & Tips

It is also recommended that the **Worse Access Time** is checked, which is found just above the **Average Access Time**.

The number of requests next to **Worse Access Time** (shown in brackets), is an indication of the health and stability of the network.

Note:

These access times are directly linked to the **PLC Task Time**, **CPU Load** and the **Network Bandwidth**.

3 To calculate the time needed to scan all the active items in a given group:

Multiply the number of **Varman** requests from the previous task by the **Average Access Time**.

- *[Worse Access Time=125ms]*
- *Nb Varman Req. = 20*

Average Access Time = **32ms**

Therefore the time needed to scan the group is: **20*32ms= 640 ms**

Task 2: OFS Group Scan Time (cont.)

How To: Test and Set (cont.)

Furthermore, **OFS Server** is able to send multiple requests in a parallel fashion, so the above result needs to be divided by the number of parallel requests.

The number of requests sent in parallel to a device is configured by **Max Channel** and **Max Pending** parameters.

- **Max Channel** is the Number of Channel (TCP/IP Connection) opened on the device.
- **Max Pending** is the Number of Requests sent by the **OFS Server** to a device waiting for a response.

By default **OFS Server** sets **Max Channel** to **4** and **Max Pending** to **0.**

Note:

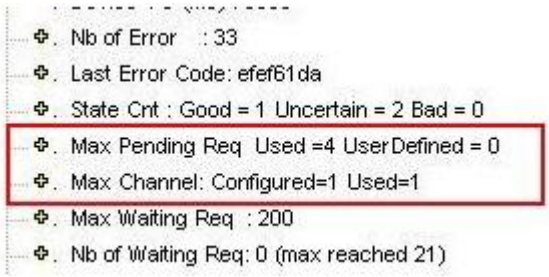
Max Pending = **0** means that the user lets the **OFS Server** identify the **PLC's** target communication port and decide how many parallel requests may be sent at the same time.

OFS Server will base its decision as per a predefined table listing the type and reference of the communication module (NOE/COPRO/TSY …).

Hints & Tips

It is recommended to configure **Max Pending**, either at **0**, or at the same value as **Max Channel**.

The number of **Max Pending** requests, as decided by **OFS Server,** may be checked from the **Network Window** of **OFS Server**.



Hints & Tips

This value may also be read directly from the **OFS Client** as: **<alias name>!#NbrMaxPending.**

Task 2: OFS Group Scan Time (cont.)

How To: Test and Set (cont.)

The actual time needed to scan the entire **OPC Group** is therefore:

- (Nb Varman Req * Average Access Time) / Used Max Pending

  = (20 * 32) / 4 = 160ms

In the example used, **OFS Server** identifies the communication port to be a 140NOE 771, which is able to process up to 12 parallel requests in a single cycle, with firmware v4.7.

However, for consistency reasons, and in order not to consume all communication bandwidth available in the PLC, **OFS Server** still identifies the NOE as allowing only up to **4 Max Pending**.

Therefore the value of **Used Max Pending** is decided by **OFS Server** to be **4.**

Hints & Tips

If we wanted to maximise communication performance **Max Pending** *could*, in this specific case, be manually overridden to **12**.

But in our case, we will leave **MaxPending** to **0** (i.e.: **OFS Server** will set it to **4)**.

Task 2: OFS Group Scan Time (cont.)

Multiple Vijeo Citect Servers

As the time to scan the 10,000 items = 20 * 32 / 4 = **160ms**

Then consideration needs to be made regarding the total number of connections to this **Ethernet** port: if two **Vijeo Citect Redundant Servers** are running (i.e. 2 x **OFS Server** instances in parallel).

A situation may occur, for instance during redundancy switch over, when the two instances of **OFS Server** will send their maximum number of parallel requests to the **PLC.**

This is **2 x Max Pending.**

In this case the **PLC** will not be able to service the requests in one cycle, so it will have to buffer the requests and service them in several cycles.

 **Note:**

This is only a transient state though, as during normal and steady states, only one server will be active and sending requests.

OFS Client Performance (cont.)

Task 3: Test OFS Client Group Update Rate

- The following procedure explains how to Test the **OFS Client Group Update Rate** for a **PLC.**

Now that a figure for the **OFS Group Scan Time** has been calculated, the **OFS Client Group Update Rate** (**Or Subscription Rate**) can be reviewed.

1 As per **Task 1** the **Group Update Rate** was set to **5000ms**.

This can now be tested at a decreased level: approx 2 times the **Group Scan Time**.

Note:

This will test whether **OFS Server** is able to keep up with the performance required, and if the previous theoretical calculation works in practice.

2 Turn on the **Verbose Mode** in **OFS Server**

- **`Misc » Verbose Mode`**

1 Enable the **Diagnostics Window**

- **`General » Diagnostic Window`**

Note:

If some requests are not responded to before the next **OFS Server** poll cycle, a message will appear in the **OFS Server** Diagnostic Window saying **"Polling rate overrun for XXX".**

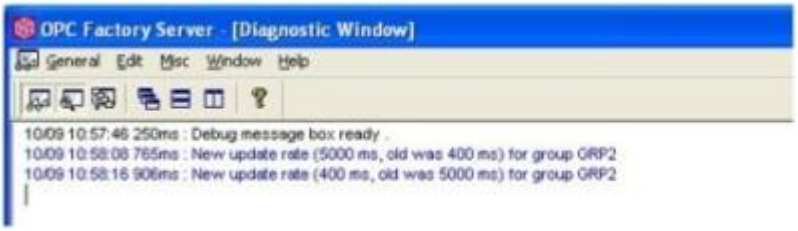
Task 3: OFS Client Update Rate (cont.)

How To: Test (cont.)

1 The calculated **Group Scan Time** is **160ms**.

Therefore decrease the **OFS Client Group Update Rate** to **400ms (380ms rounded up)**.

2 Monitor the **Diagnostic Window** in **Verbose** mode.



3 As expected **No message** appears**.**

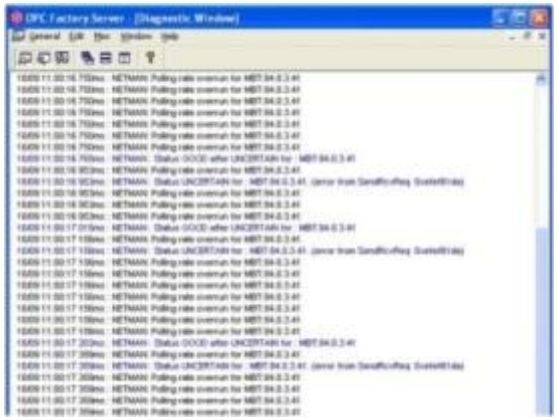
This confirms that **OFS Server** is able to send and receive all **20** requests, before the next **OFS Server** scan cycle (**400ms**).

Task 3: OFS Client Update Rate (cont.)

How To: Test (cont.)

4 Now reduce the **OFS Server Client Group Update Rate** down to **300ms**.

5 Monitor the **Diagnostic Window** in **Verbose** mode again.



6 This time **OFS Server** fails to keep up with the desired **OFS Client Group Update Rate**.

Note:

The quality of some of the above items are **'uncertain'.** This uncertain quality will be propagated up to the SCADA system, and therefore should be avoided**.**

Hints & Tips

A general rule therefore is to set the OFS Client Group Update Rate to 2 Times the calculated OFS Server Scan Time.

OFS Client Performance (cont.)

Task 4: Set OPC Client Group Update Rate

- The following explains how to **Set** the **OFS Client Group Update Rate** for an **OFS Client Application**.

1 Now that the testing of the **OFS Client** is finished follow the same steps in **Tasks 1 to 3** with a real **OPC Application.**

This will enable a realistic **Group Update Rate** to be specified in a **SCADA/OPC Client** and determine what performance can be expected**.**

Note:

As mentioned in the last **Hints & Tips** of **Task 1**, the number of **Varman** requests in a real **SCADA** project, is likely to be significantly higher.

This is because the required item addresses will not be contiguous, so **OFS Server** will need to send more requests.

Hints & Tips

The minimum **Group Update Rate** found by default in **OFS Server** could be used, which is the limit below which **OFS Server** will not accept **OPC** client requests.

But instead the **Group Update Rate** should be configured in the **OFS Client** directly.

For the **OFSOPC driver i**n **Vijeo Citect**, this is via the following **.ini** parameters:

```
[OFSOPC]

      Group1UpdateRate

      Group2UpdateRate

      Group3UpdateRate
```