



Product:	UDPIOServer
Product version:	V1.6
Document ID:	PM-UDPIOServer
Doc revision:	A
Written/Appr.:	KD / RE
Date:	11/06/2008

Industrial Control Design AS



UDPIOServer V1.6

Programmer Manual

The content of this document is confidential information not to be published without the consent of Industrial Control Design AS.

Industrial Control Design AS, www.icd.no, support@icd.no, forum.icd.no

Contents

1. INTRODUCTION.....	3
2. INSTALLATION.....	4
2.1. About.....	4
2.2. Preparing your application for linking with the UDPIOServer library.....	4
2.3. How to add the UDPIOServer component to a CDP Application.....	4
3. CUSTOMIZATION.....	5
3.1. Packet validation.....	5

1. Introduction

This document describes how the UDPIOServer library component works, and how to compile and link with it.

The UDPIOServer library has the following programmer-related features:

- Incoming packets may be verified by overriding the virtual method `ValidatePacket()` in the `UDPIOServer` class. The default implementation only checks that the packet size is correct.

For details on features, configuration and usage of UDPIOServer component, please refer to the User Manual)

2. Installation

2.1. About

This chapter describes how to link the UDPIOServerLib into a CDP application project.

2.2. Preparing your application for linking with the UDPIOServer library

Copy the example UDPIOServer.xml component file that came with the distribution into your project's Components folder, and the UDPIOServer.xml model.xml file into the Models folder.

Copy the UDPIOServerLib folder (source for the UDPIOServer library) either into your project, to a location for standard libraries, or copy the precompiled UDPIOServerLib_<config>.lib to a location where your linker can find it.

Include the UDPIOServerLib.h header file in your application's Libraries.h file:

```
#include <UDPIOServerLib.h>
```

Add the path to where the lib source code was unpacked to:

- CDP_Application -> Properties
 - C/C++ -> General -> Additional Include Directories

Link the UDPIOServer library into your application by putting UDPIOServer_<config>.lib in a folder included in your linker path. For the CDP_Application project:

- CDP_Application->Properties
 - Linker -> Input -> Additional dependencies

2.3. How to add the UDPIOServer component to a CDP Application

Add the following to your project's Application.xml:

Inside the **<Components>** element, add an instance of an UDPIOServer component:

```
<Component Name="UDPIOServer" src="Components/UDPIOServer.xml"></Component>
```

Or, inside the **<Subcomponents>** element, add:

```
<Subcomponent Name="UDPIOServer" Model="UDPIOServer" src="Components/UDPIOServer.xml"></Subcomponent>
```

This will tell CDP to initialize a component named “UDPIOServer” from a component file located at “Components/UDPIOServer.xml”.

Make sure that your Models\ folder contains an UDPIOServer.xml model file, or the component will not be initialized correctly. It should also contain the SignalTwoWay_*.xml model files.

3. Customization

The UDPIOServer component can be customized in C++. Here is a list of the most common customizations, and how they are accomplished.

3.1. Packet validation

Incoming packets may be verified by overriding **ValidatePacket()** in **UDPIOServer** class. The default implementation only checks that the packet size is correct.

Subclass the **UDPIOServer** class, override the virtual method **ValidatePacket()** adding your specific packet validation while making sure to call the superclass version as well. Provide your new component by adding a LibBuilder for it and instantiate it instead of UDPIOServer in your CDP application configuration.