# Unity Application Generator 3.3 SP4

## a SoCollaborative software Quick Start

09/2014

33003222.13

www.schneider-electric.com

Schneider Electric

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 nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information 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 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.

# Table of Contents

# Safety Information

## Important Information

### NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

**⚠ DANGER**

**DANGER** indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

**⚠ WARNING**

**WARNING** indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

**⚠ CAUTION**

**CAUTION** indicates a hazardous situation which, if not avoided, **could result** in minor or moderate injury.

***NOTICE***

***NOTICE*** is used to address practices not related to physical injury.

## PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

### ⚠ WARNING

**UNINTENDED EQUIPMENT OPERATION**

Test and verify the operation of applications before release.

Perform safety analysis and tests appropriate to the application. This software package provides a means to quickly design an application. System programming and interlocks are required to make sure the safety of the resulting applications.

Any applicable regulatory requirements must be met before releasing an application for general use.

**Failure to follow these instructions can result in death, serious injury, or equipment damage.**

### ⚠ WARNING

**LOSS OF CONTROL**

The designer of any control scheme must consider the potential detected failure modes of control paths and, for certain critical control functions, provide a means to achieve a safe state during and after a detected failure in the path. Examples of critical control functions are emergency stop and overtravel stop.

Separate or redundant control paths must be provided for critical control functions.

System control paths may include communication links. Consideration must be given to the implications of unanticipated transmission delays or detected failures of the link.

Each application must be individually and thoroughly tested for proper operation before being placed into service.

**Failure to follow these instructions can result in death, serious injury, or equipment damage.**

# ⚠ CAUTION

**UNPREDICTABLE APPLICATION**

Do not use a tool such as Unity Pro to edit segments of the application generated by UAG. For communications links and properties to be correctly maintained, any changes to these segments must be made within the UAG environment.

**Failure to follow these instructions can result in injury or equipment damage.**

# About the Book

## At a Glance

### Document Scope

This manual contains a quick start for Unity Application Generator (UAG).

### Validity Note

This document applies to

- Unity Application Generator 3.3 SP4
- Microsoft Windows XP Professional, Windows Vista Business and Windows 7 Professional (32/64 bit)

### Related Documents

| Title of Documentation | Reference Number |
|---|---|
| Unity Application Generator Basic Manual | 33002830 |
| Unity Application Generator Extended Manual | 33003669 |
| SCoD Editor Manual | 33002608 |
| Unity Application Generator (UAG) ArchestrA Generator User Manual | 33004144 |
| Unity Pro Software Reference Manual | 35006144 |

You can download these technical publications and other technical information from our website at www.schneider-electric.com.

# Chapter 1
# General Information

## Overview

This chapter provides general information about Unity Application Generator and installation hints for the additional programs like Unity Pro and Vijeo Citect.

## What Is in This Chapter?

This chapter contains the following topics:

# Introduction

## Aim of the Document

This documentation helps the user to generate an application with Unity Application Generator for the PLC programming software Unity Pro and the HMI software Vijeo Citect, Monitor Pro, Generic HMI or iFIX.

**NOTE:** For iFIX a different sample application is used. The information regarding the iFIX demo application can be found in Appendix A *(see page 55)* at the end of this document.

## Structure of the Document

The document is divided into different parts.

- General information and installation hints.
- Start of the delivered demo program. Explanation of the settings which have to be done by the user to start the application.
- Generation and start of the demo application step-by-step. From the customization to the running HMI application.
- Settings and specialties of the different support HMI systems.

## Programming Environment

The demo programs are tested for the following software:

- Unity Application Generator 3.3 SP4
- Unity Pro 4.1, 5.0, 6.0, 7.0, 8.0, and 8.1
- Schneider Electric Vijeo Citect 6.10, 7.0, 7.10, 7.20, 7.30, 7.40, and 7.40 SP1
- Schneider Electric Monitor Pro 7.6 (not supported for new features of UAG 3.2 or later)
- Microsoft Internet Explorer

# System Requirements

## Operating System

One of the following operating systems is required:

- Windows XP Professional
- Windows Vista Business
- Windows 7 Professional (32/64 bit)
  **NOTE:** To run UAG on Windows 7 Professional system, you have to have administrator rights.

## Unity Application Generator

- Unity Application Generator 3.3 SP4, Schneider Electric

## HMI Systems

- Microsoft Internet Explorer (for Generic HMI), Microsoft
- Schneider Electric Vijeo Citect 6.10/7.0/7.10/7.20/7.30/7.40/7.40 SP1
- Schneider Electric Monitor Pro 7.6 (not supported for new features of UAG 3.2 or later)

## Open Factory Server

An OPC Factory Server (OFS) 3.1 or higher has to be installed on the system.

## Supported PLC Software

- Schneider Electric Unity Pro 4.1/5.0/6.0/7.0/8.0/8.1

## Supported PLC Hardware

Hardware, which is listed within Unity Application Generator, will be supported.

# Installation Hints

## Introduction

The user has to install the necessary software packages as defined in the specific user manual. The following installation hints will help to avoid the most common difficulties with the software installation.

## Installation Sequence

Unity Application Generator is using different directories of the other applications like Unity Pro or Vijeo Citect. Therefore it is necessary to install the application in the following sequence.

1. Unity Pro
2. Vijeo Citect / Monitor Pro and the Open Factory Server (OFS) (see Monitor Pro below)
3. Unity Application Generator

## Registration

Registration of the software installation is strongly recommended. Follow the instructions in the software to register this product.

## Unity Application Generator

During the installation process, you will be asked to enter the following information:

- Serial Number
- Part Number

This information is provided with the software and can be found inside the package.

After the installation of UAG the software can be registered or used as a demo version for 21 days.

## Monitor Pro

You have to follow the installation steps from the `Installation Guide.pdf`. This document will be delivered within the installation CD-ROM of Monitor Pro. The user can find the documentation on the CD-ROM within the `\Documentation` directory.

# Chapter 2
# Description of the Sample Application

## What Is in This Chapter?

This chapter contains the following topics:

# Example: Softdrink Production

## General Information

The sample project gives an example how to handle automation with UAG and shows the basic features of UAG.

Thus, the sample mainly focuses on the features and handling of UAG and less on a real project.

Therefore only a part of an entire plant for soft drink production is configured in this sample.

## Description of the Production

A company produces soft drinks with 3 different flavours and fills the drinks in bottles, cans and casks.

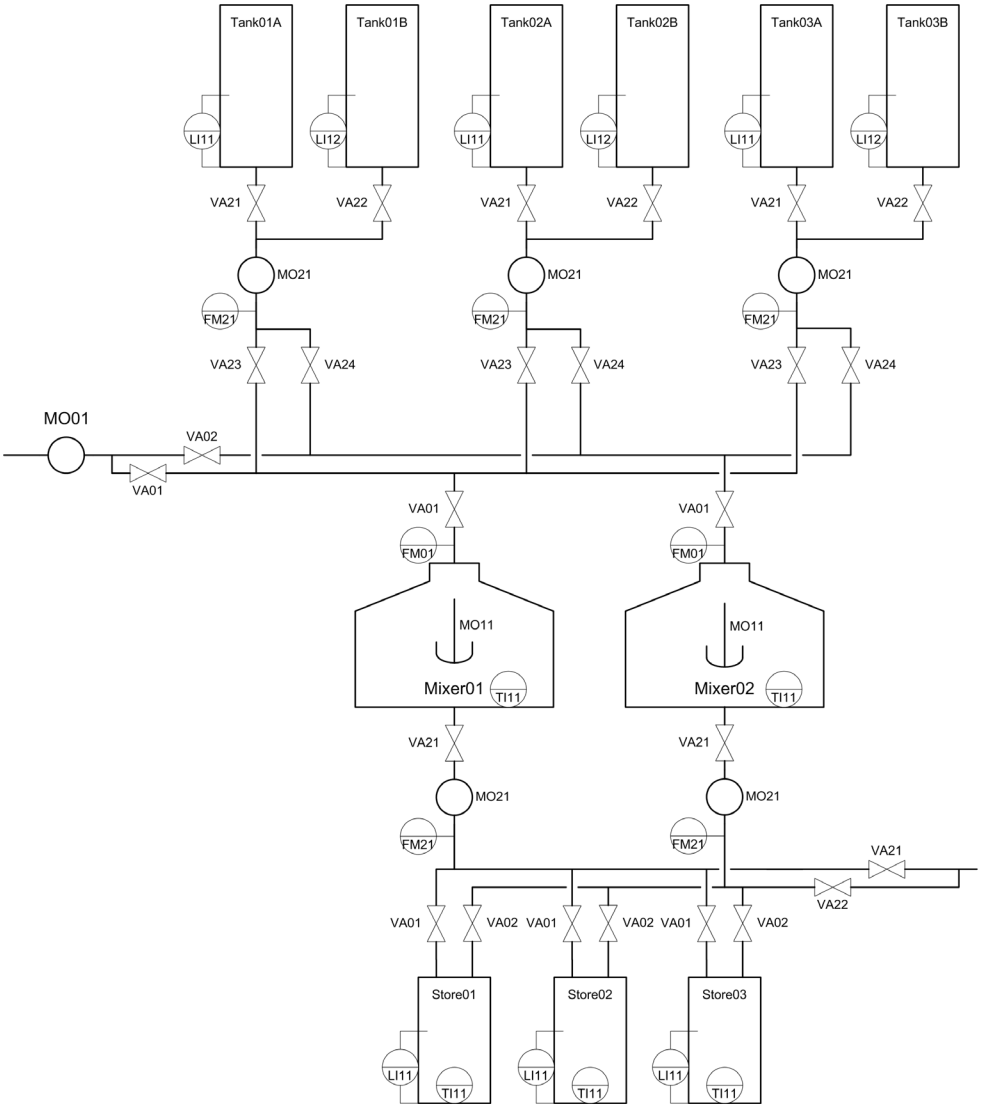
The company has divided the whole process into the soft drinks production and the filling and logistics part, but the sample will mainly deal with the production part.

The fluid ingredients for the drinks are stored in tanks. There are 2 tanks for each ingredient. According to the mixture the ingredients are pumped to a vessel, where the ingredients are mixed and hold on temperature.

Two vessels are available for mixing the drinks. Afterwards the final product is pumped to a storage tank and from the tank to the different filling stations.

Three storage tanks exist, 1 for each drink. A cleaning of the mixers with water is possible.

The figure below shows the production line with the tanks, vessels, valves, pumps and indicators.



## Structuring the Process

### General

The soft drinks process is divided into “Production” and “Filling”. This is reflected as 2 different **Areas** in UAG.

The production consists of “Ingredients”, “Mixers” and “Storage”.

The filling area consists of “Bottles”, “Cans” and “Casks”.

This division is on UAG’s **Process Cell** level.

UAG’s **Unit** level divides the process in a more detailed way.

The ingredients for the drinks are “Sugar”, “Water”, “Concentrate” and “CO2”.

Two “Lines” and “Cleaning” belong to the mixers and storage exists for “Coke”, “Lemon” and “Orange”.

“Bottles” and “Cans” have 2 lines each, “Casks” has 1 line only.

### Naming Rules for Units

As UAG requests unique names for every level (except for the Control Module level), the company defined a naming rule for units.

This rule allows only starting the name with a 2 digit number followed by the real unit name.

The first digit is related to the **Process Cell**:

0 - “Ingredients”, 1 - “Mixers”, 2 - “Storage”, 3 - “Bottles”, 4 - “Cans”, 5- “Casks”.

The second digit is a consecutive number within the **Unit** level.

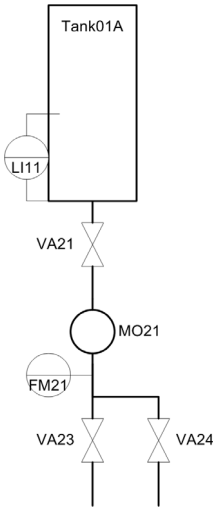
With the definition of these 3 levels (**Area**, **Process Cell** and **Unit**) the basic structure of the soft drink process is defined.

Afterwards the production equipment (tank, vessel etc.) with its functions can be added.

# Definition and Behavior of a Tank and the Mixer

## Elements of the Tank

The figure below shows the elements of a tank for ingredients with a level indicator in the tank, the output valve, motor and flow meter for pumping the ingredient to the vessel and 2 valves for enabling vessel 1 or 2.




In UAG the tank is realized as equipment module and the single elements of the tank are realized as Control Modules.

## Control Modules

Control Modules are based on a SCoD library. For this sample the basic libraries for Vijeo Citect (*UAGBasic_Citect.osl*) and Monitor Pro (*UAGBasic_MPro.osl*) were used.

For Generic HMI the basic library from Vijeo Citect was copied (*UAGBasic_Gen.osl*) and stored in the project folder.

### Used SCoDs

The following SCoDs were used:

| SCoD | Description |
|---|---|
| VALMO | valves<br>(respectively VALMO_P10 for the Process Application Library) |
| MOT1D1S | pump<br>(respectively MOT1D1S_P10 for the Process Application Library) |
| ANA_IN | level, flow meter<br>(respectively ANAIN_P10 for the Process Application Library) |

### Equipment Module

Beside the elements shown in the figure above the equipment module `TA01A` (the tank) contains additionally 1 logical Control Module for controlling the operation modes of the tank, this is called `CTRL` and based on the SCoD OP_MODE (respectively OPMODE_P10 for the Process Application Library).

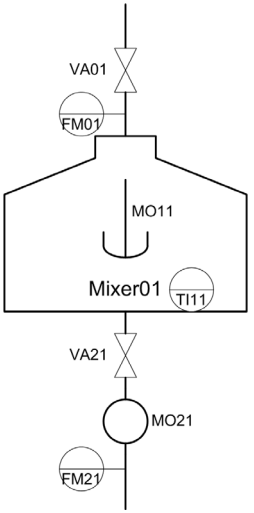
### Behavior of the Tank

The behavior of the tank is basically defined by the behavior of the SCoDs used for the tank's single elements. The interaction of the elements is done by interlocks and links.

## Elements of the Mixing Vessel

The sample project contains also 1 mixing vessel, which is designed in a similar way as the tank described above. Concerning the physical model there is nothing new, but the linking to the topological model is different.

The elements of the vessel are shown in the following figure:



# Chapter 3
# Creating a Demo Project

## Overview

This chapter contains information how to create the demo project with the explanation of the primary settings. The user has to enter these settings to create an accurate project.

This chapter describes only the part of UAG. The steps to create a project are the same regardless of the HMI system used. If there are any variations for the different HMI systems there will be hints about it.

## What Is in This Chapter?

This chapter contains the following topics:

# Demo Project

## Introduction

Demo projects (*demoplant.osp*) are part of the Unity Application Generator DVD. The demo projects are delivered within the *\Examples\Demo Plant* folder and its subfolders.

The folders contain the necessary files to start the demo application for the desired HMI system.

Demonstration projects are used as examples only and are not to be used in production environments.

The following list shows the different subfolders to the HMI systems:

- Vijeo Citect
  *\Examples\Demo Plant\VijeoCitect*
- Monitor Pro
  *\Examples\Demo Plant\Monitor Pro*
- iFIX
  *\Examples\Demo Plant\iFIX*
- Generic HMI
  *\Examples\Demo Plant\Generic*

## Using the Demo Project

**NOTE:** If you want to use the prepared demo projects instead of creating them by yourself, copy the files as shown in the table below, skip the next sections and continue with Generate the Project *(see page 41)*.

The following table shows the steps to copy the sample files from DVD to the working folder.

| Step | Action | Comment |
|---|---|---|
| 1 | Copy the files and folder structure (e.g. for Vijeo Citect). | You have to copy the whole folder structure with the files from DVD (*\Examples\Demo Plant\VijeoCitect\Citect71*) to your harddisk (e.g. *c:\Demo Plant*) |
| 2 | Clear the read-only attribute. | Select the *c:\Demo Plant* folder and clear the read-only attribute (when asked, choose **Apply changes to this folder, subfolders and files**). |

## Final Steps for Vijeo Citect

See Final Steps for Vijeo Citect *(see page 43)*.

## Final Steps for Monitor Pro

See Final Steps for Monitor Pro *(see page 45)*.

## Final Steps for Generic HMI

See Final Steps for Generic HMI *(see page 50)*.

# Prepare the Process Application Library

## Introduction

This section describes the preparation of the Process Application Library.

## Steps for the Preparation

The following table describes the preparation of the Process Application Library.

| Step | Action | Comment |
|---|---|---|
| 1 | Upgrade the Process Application Library. | Open the file *UAGProcessLibVCVxx.osl* (where xx is the version of the Process Application Library) under *C:\Program Files\Schneider Electric\Unity Application Generator\Db* and confirm the upgrade. |
| 2 | Close the SCoD Editor. | - |

# Customization

## Introduction

This section describes the creation of the customization of a new UAG project.

## Requirements

Unity Application Generator and its sub programs have to be installed correctly.

## Steps for the Customization

The following table describes the steps to generate *demoplant.osc*.

| Step | Action | Comment |
|---|---|---|
| 1 | Create a working directory. | You have to create a working directory within the Microsoft Windows Explorer or a similar program. Please create the directory *c:\Demo Plant*. |
| 2 | Start the *CustomEdit.exe* application. | You can start the application by using the program icon within UAG program group. |
| 3 | Create a new customization. | Enter **File** →**New** to create a new Customization. Select the HMI system which you want to customize for.<br>● Vijeo Citect<br>● Monitor Pro<br>● iFIX<br>● Generic |
| 4 | Select a SCoD library. | Select and import the following SCoD library using **General** →**Libraries**<br>● Process Application Library for Vijeo Citect: *<INSTALLDIR>\DB\UAGProcessLibVCVxx.osl* (where *xx* is the the version of the Process Application Library).<br>● Vijeo Citect: *<INSTALLDIR>\DB\UAGBasic_Citect.osl*<br>● Monitor Pro: *<INSTALLDIR>\DB\UAGBasic_MPro.osl*<br>● Generic: *<PROJECTDIR>\UAGBasic_Gen.osl*<br>Save the customization file (e.g. *c:\Demo Plant\ DemoPlant.osc)* to be able to specify the *<PROJECTDIR>* under **General** →**Path**.<br>Press **Import** to get the data. |
| 5 | Additional actions necessary for Process Application Library for Vijeo Citect. | ● Set the Symbol Library Project under **General** →**VijeoCitect** →**Symbol Library Project** to **UAGProcessLibVCVxxp0** (where **xx** is the the version of the Process Application Library).<br>● Set the location of the following archive name under **Data** →**Archive Name**<br>  ● Archive Name: `Daily`<br>  ● Description: `Daily Log`<br>  ● Location: `ArchiveUnitLocation` |

| Step | Action | Comment |
|---|---|---|
| 6 | Additional actions necessary for Vijeo Citect. | Set the Symbol Library Project under **General** →**VijeoCitect** →**Symbol Library Project** to **UAGBasic**. |
| 7 | Additional actions necessary for Monitor Pro. | • Set the Monitor Pro driver under **General** →**Monitor Pro** →**Monitor Pro driver to OPC** (OFS, supporting both Modbus Plus and TCP/IP).<br>• Add the following archive name under **Data** →**Archive Name**<br>  • Archive Name: `ArchiveUnit`<br>  • Description: `Archive of PIDs`<br>  • Location: `ArchiveUnitLocation` |
| 8 | Additional actions necessary for Generic. | Enter the following export format name under **General** →**Export Format Name**<br>• Name: `Generic HMI Data`<br>• Transformation File: `IOPoints2HTML.xsl`·<br>• Output File: `DemoPlantIOPoints.html`<br>• Transformation File: `GenericHMICSV.xls`<br>• Output File: `DemoPlant.csv` |
| 9 | Analyze the customization. | You have to start the analysis.<br>Enter **Customization** →**Analyze**.<br>The state of every topic within the protocol has to be OK. |
| 10 | Save the customization. | You have to enter a name for your customization. For example *c:\demoproj\demoplant.osc*. |
| 11 | Close the Customization Editor. | - |

# Create a new Project with Unity Application Generator

## Introduction

This section describes the creation of a new UAG project.

The creation of a new project can start with the Topological Model, with the Physical Model or mixed.

**NOTE:** If you want to create the project step-by-step, you do not have to copy the demo files from the DVD to your hard disk as described in Using the Demo Project *(see page 24)*.

## Requirements

The following applications have to be installed correctly:
- Unity Application Generator and all of its sub programs
- Unity Pro
- Vijeo Citect or Monitor Pro

## Options for the Demo Project

The following table shows the general steps to generate the `demoplant.osp`.

| Step | Action | Comment |
|---|---|---|
| 1 | Start Unity Application Generator. | You can start the application by using the program icon within the UAG program group. |
| 2 | Create a new project. | Select the customization just created from the file name list. |
| 3 | Set the working directories. | Set the directories for the different files which will be generated from Unity Application Generator.<br>You can enter the options under: **View** →**Options** within UAG.<br>Please enter the following settings:<br>● `General`<br>  ● `Default Documentation Path`: c:\demoproj<br>● `PLC`<br>  ● `PLC Project Path`: c:\demoproj<br>● `Monitor Pro`<br>  ● `HMI Application Path`: c:\demoproj<br>  ● `Data Server Application Path`: c:\demoproj<br>● `Analyze Project`: no changes necessary<br>● `Memory Mapper`: no changes necessary |
| 4 | Save the project | You can save the project under: **File** →**Save As...** within UAG.<br>Please browse to the path of the demo project (c:\demoproj). |

## Creation of the Topological Model

The following table shows the steps to generate the Topological Model of the `demoplant.osp` project.

| Step | Action | Comment |
|---|---|---|
| 1 | Right-click on `Network Segments` and select `New Network Segment` from the context menu. | Create a new network segment. For example:<br>● `Name`: plant_net<br>● `Network Type`: Ethernet<br>● `Subnet Mask`: 255.255.255.0 |
| 2 | Right-click on `Data Servers` and select `New Data Server...` from the context menu. | Create a new Data Server with the following settings:<br>● `Basic` tab<br>  ● `Name`: DS_DRINK<br>  ● `Network Timeout`: 1s<br>● `Networks` tab<br>  ● `Network Type`: Ethernet<br>  ● `Network Segment`: plant_net<br>  ● `IP Address`: 127.0.0.1 (simulator of Unity Pro) |
| 3 | Right-click on `HMIs` and select `New HMI...` from the context menu. | Create 2 new HMI with the following names:<br>● `HMI`: HMI_P<br>● `HMI`: HMI_F |
| 4 | Right-click on `HMI_P` and select `New Control Domain...` from the context menu. | Create a new Control Domain with the following settings:<br>● `Name`: CD_P<br>● `Description`: Control Domain Production |
| 5 | Right-click on `HMI_P` again and select `Open Pictures...` from the context menu. | - |
| 6 | Right-click in the `Pictures` dialog box and select `New Picture Group` from the context menu. | Create 2 new picture groups with the following names:<br>● `Picture Group`: PG_Ingredients<br>● `Picture Group`: PG_Mixers |
| 7 | Right-click on `PG_Ingredients` and select `New Picture` from the context menu. | Create a new picture with the following name:<br>● `Picture`: Tank01A |
| 8 | Right-click on `PG_Mixers` and select `New Picture` from the context menu. | Create a new picture with the following name:<br>● `Picture`: Mixer01A |
| 9 | Right-click on `HMI_F` and select `New Control Domain...` from the context menu. | Create a new Control Domain with the following settings:<br>● `Name`: CD_F<br>● `Description`: Control Domain Filling |

| Step | Action | Comment |
|---|---|---|
| 10 | Right-click on `PLCs` and select `New PLC...` from the context menu. | Create a new PLC with the following properties:<br>• `Basic` tab<br>  • `Name`: PLC01<br>  • `Description`: PLC for Production<br>  • `PLC Family`: Unity/Quantum<br>  • `Local Rack`: 140 XBP 006 00<br>  • `CPU`: 140-CPU-651-60 |
| 11 | Expand `PLC01`, right-click on `Local` and select `Open Rack` from the context menu. | Create a new rack with the following modules in the different slots (1.. 6):<br>• `Power supply`: 140-CPS-114-x0<br>• `CPU`: 140-CPU-651-60<br>• `Digital In`: 140-DDI-353-00<br>• `Digital Out`: 140-DDO-353-00<br>• `Analog In`: 140-AVI-030-00 |
| 12 | Right-click on the `CPU 140-CPU-651-60` and select `Properties…` from the context menu. | `Ethernet`<br>• `Network Segment`: plant_net<br>• `IP Address`: 127.0.0.1 |
| 13 | Right-click on `PLC01` and select `New Rack...` from the context menu. | Create a new rack with the following properties:<br>• `Link Type`: Ethernet I/O<br>• `Rack Type`: ETHERNET I/O (128)<br>• `Head Slot`: 2<br>• `Description`: Remote I/O |
| 14 | Right-click on `Ethernet I/O_Drop 2` and select `Open Rack` from the context menu. | Create a new rack with the following modules in the different slots (1.. 6):<br>• `Digital In`: 170-ADI-350-00<br>• `Digital In`: 170-ADI-350-00<br>• `Digital Out`: 170-ADO-350-00<br>• `Digital Out`: 170-ADO-350-00<br>• `Analog In`: 170-AAI-030-00<br>• `Analog In`: 170-AAI-520-40 |
| 15 | Right-click on `Ethernet I/O_Drop 2` and select `Properties...` from the context menu. | Set the following IP Addresses for the modules:<br>• 170-ADI-350-00: 192.168.1.100<br>• 170-ADI-350-00: 192.168.1.101<br>• 170-ADO-350-00: 192.168.1.102<br>• 170-ADO-350-00: 192.168.1.103<br>• 170-AAI-030-00: 192.168.1.104<br>• 170-AAI-520-40: 192.168.1.105 |

## Creation of the Physical Model

The following table shows the steps to generate the Physical Model of the `demoplant.osp` project.

| Step | Action | Comment |
|---|---|---|
| 1 | Right-click on `Site` and select `New Area` to create the areas `Production` and `Filling` | Enter the following properties to the created areas:<br>• `Production`<br>  • for `Name` and `Description`: Production<br>• `Filling`<br>  • for `Name` and `Description`: Filling |
| 2 | Right-click on the area `Production` and select `New Process Cell` to create the process cells `Ingredients`, `Mixer` and `Storage`. | Enter the following properties to the created process cells:<br>• `Ingredients`<br>  • for `Name` and `Description`: Ingredients<br>• `Mixers`<br>  • for `Name` and `Description`: Mixers<br>• `Storage`<br>  • for `Name` and `Description`: Storage |
| 3 | Right-click on the process cell `Ingredients` and select `New Unit` to create the units `01Sugar`, `02Water`, `03Concentrate` and `04CO2`. | Enter the following properties to the created units:<br>• `01Sugar`<br>  • for `Name` and `Description`: 01Sugar<br>• `02Water`<br>  • for `Name` and `Description`: 02Water<br>• `03Concentrate`<br>  • for `Name` and `Description`: 03Concentrate<br>• `04CO2`<br>  • for `Name` and `Description`: 04CO2 |
| 4 | Right-click on the unit `Sugar` and select `New Equipment Module` to create the equipment module `TA01A`. | Enter the following properties to the created equipment module:<br>• `TA01A`<br>  • `Name`: TA01A<br>  • `Description`: Tank 01 for Sugar<br>  • `PLC Name`: PLC01<br>  • `PLC Section Name`: Tank01A<br>  • `Control Domain`: HMI_P / CD_P<br>  • `Picture`: \PG_Ingredients\Tank01A |

| Step | Action | Comment |
|---|---|---|
| 5 | Right-click on the equipment module `TA01A` and select `New Control Module` to create the control modules `LI11`, `VA21`, `MO21`, `FM21`, `VA23`, `VA24`, `CTRL`. | Enter the following properties to the created control modules:<br>• `LI11`<br>  • `Control Modules Type`: ANA_IN / ANAIN_P10<br>  • `Control Modules Name`: LI11<br>  • `Control Modules Description`: Level Indica. Tank<br>  • `PLC`: PLC01<br>• `VA21`<br>  • `Control Modules Type`: VALMO / VALMO_P10<br>  • `Control Modules Name`: VA21<br>  • `Control Modules Description`: Tank Valve Out<br>  • `PLC`: PLC01<br>• `MO21`<br>  • `Control Modules Type`: MOT1D1S / MOT1D1S_P10<br>  • `Control Modules Name`: MO21<br>  • `Control Modules Description`: Motor Flow Pump<br>  • `PLC`: PLC01<br>• `FM21`<br>  • `Control Modules Type`: ANA_IN / ANAIN_P10<br>  • `Control Modules Name`: FM21<br>  • `Control Modules Description`: Flow Meter Tank<br>  • `PLC`: PLC01<br>• `VA23`<br>  • `Control Modules Type`: VALMO / VALMO_P10<br>  • `Control Modules Name`: VA23<br>  • `Control Modules Description`: Valve to Mixer 01<br>  • `PLC`: PLC01<br>• `VA24`<br>  • `Control Modules Type`: VALMO / VALMO_P10<br>  • `Control Modules Name`: VA24<br>  • `Control Modules Description`: Valve to Mixer 02<br>  • `PLC`: PLC01<br>• `CTRL`<br>  • `Control Modules Type`: OP_MODE / OP_MODE_P10<br>  • `Control Modules Name`: CTRL<br>  • `Control Modules Description`: Operation Control<br>  • `PLC`: PLC01 |
| 6 | Add the process cells `Bottles`, `Cans` and `Casks` to the area `Filling`. | Enter the following properties to the created process cells:<br>• `Bottles`<br>  • for `Name` and `Description`: Bottles<br>• `Cans`<br>  • for `Name` and `Description`: Cans<br>• `Casks`<br>  • for `Name` and `Description`: Casks |

| Step | Action | Comment |
|---|---|---|
| 7 | Add the following units:<br>• to the process cell `Mixer`: `11Line`, `12Line` and `13Cleaning`<br>• to the process cell `Storage`: `21Coke`, `22Orange`, `23Lemon`<br>• to the process cell `Bottles`: `31Line`, `32Line`<br>• to the process cell `Cans`: `41Line`, `42Line`<br>• to the process cell `Casks`: `51Line` | Enter the following properties to the created units:<br>• `11Line`<br>  • for `Name` and `Description`: 11Line<br>• `12Line`<br>  • for `Name` and `Description`: 12Line<br>• `13Cleaning`<br>  • for `Name` and `Description`: 13Cleaning<br>• `21Coke`<br>  • for `Name` and `Description`: 21Coke<br>• `22Orange`<br>  • for `Name` and `Description`: 22Orange<br>• `23Lemon`<br>  • for `Name` and `Description`: 23Lemon<br>• `31Line`<br>  • for `Name` and `Description`: 31Line<br>• `32Line`<br>  • for `Name` and `Description`: 32Line<br>• `41Line`<br>  • for `Name` and `Description`: 41Line<br>• `42Line`<br>  • for `Name` and `Description`: 42Line<br>• `51Line`<br>  • for `Name` and `Description`: 51Line |

| Step | Action | Comment |
|---|---|---|
| 8 | Add the equipment modules `SO01A`, `MI01A` and `SI01A` to the unit `Mixer -> 11Line` | Enter the following properties to the created equipment module:<br>• `SO01A`<br>  • `Name`: SO01A<br>  • `Description`: Input for Vessel<br>  • `PLC Name`: PLC01<br>  • `PLC Section Name`: Source<br>  • `Control Domain`: HMI_P / CD_P<br>  • `Picture`: \PG_Mixers\Mixer01A<br>• `MI01A`<br>  • `Name`: MI01A<br>  • `Description`: Vessel<br>  • `PLC Name`: PLC01<br>  • `PLC Section Name`: Source<br>  • `Control Domain`: HMI_P / CD_P<br>  • `Picture`: \PG_Mixers\Mixer01A<br>• `SI01A`<br>  • `Name`: SI01A<br>  • `Description`: Sink<br>  • `PLC Name`: PLC01<br>  • `PLC Section Name`: Source<br>  • `Control Domain`: HMI_P / CD_P<br>  • `Picture`: \PG_Mixers\Mixer01A |

| Step | Action | Comment |
|---|---|---|
| 9 | Add the following control modules:<br>• to the equipment module SO01A: VA01, FM01 , CTRL<br>• to the equipment module MI01A: LI11, TI11, MO11<br>• to the equipment module SI01A: VA21, MO21, FM21 | Enter the following properties to the created control modules:<br>• VA01<br>  • Control Modules Type: VALMO / VALMO_P10<br>  • Control Modules Name: VA01<br>  • Control Modules Description: Input Valve Vessel<br>  • PLC: PLC01<br>• FM01<br>  • Control Modules Type: ANA_IN / ANAIN_P10<br>  • Control Modules Name: FM01<br>  • Control Modules Description: Input Flow Vessel<br>  • PLC: PLC01<br>• CTRL<br>  • Control Modules Type: OP_MODE / OP_MODE_P10<br>  • Control Modules Name: CTRL<br>  • Control Modules Description: Operation Control<br>  • PLC: PLC01<br>• LI11<br>  • Control Modules Type: ANA_IN / ANAIN_P10<br>  • Control Modules Name: LI11<br>  • Control Modules Description: Level Vessel<br>  • PLC: PLC01<br>• TI11<br>  • Control Modules Type: ANA_IN / ANAIN_P10<br>  • Control Modules Name: TI11<br>  • Control Modules Description: Temperature Vessel<br>  • PLC: PLC01<br>• MO11<br>  • Control Modules Type: MOT1D1S / MOT1D1S_P10<br>  • Control Modules Name: MO11<br>  • Control Modules Description: Motor for Mixer<br>  • PLC: PLC01 |

| Step | Action | Comment |
|---|---|---|
| 9 | Step 9, continued. | • VA21<br>  • Control Modules Type: VALMO / VALMO_P10<br>  • Control Modules Name: VA21<br>  • Control Modules Description: Valve Output<br>  • PLC: PLC01<br>• MO21<br>  • Control Modules Type: MOT1D1S / MOT1D1S_P10<br>  • Control Modules Name: MO21<br>  • Control Modules Description: Pump for Output<br>  • PLC: PLC01<br>• FM21<br>  • Control Modules Type: ANA_IN / ANAIN_P10<br>  • Control Modules Name: FM21<br>  • Control Modules Description: Output Flow<br>  • PLC: PLC01 |

# Other Settings

## Introduction

The following table contains the settings which have to be entered for a project.

| Step | Action |
|---|---|
| 1 | Create a channel. |
| 2 | Create the `HMI Communication` ranges. |
| 3 | Do the I/O Mapping for process cell `Ingredients`. |
| 4 | Do the I/O Mapping for process cell `Mixers`. |
| 5 | Create the interlocks for the process cell `Ingredients`. |
| 6 | Create the interlocks for the process cell `Mixers`. |
| 7 | Create a variable. |
| 8 | Create links for process cell `Ingredients`. |

Detailed descriptions of the single steps can be found in the tables below.

## Create a Channel

| Step | Action | Comment |
|---|---|---|
| 1 | Right-click on `PLC01` in the Topological Model and select `Open Channels` from the context menu. | There has to be a channel between the PLC and HMI for the communication of the data.<br>Right-click in the `Channels` dialog box and select `New` from the context menu.<br>Enter the following settings:<br>• `Name`: CH_PLC01_DS<br>• `Communication Type`: Data Server<br>• `Communication Partner`: DS_DRINK<br>• `Network Type`: Ethernet<br>• `Communication Path - Path 1`: plant_net: Slot 2 (127.0.0.1 <=> (127.0.0.1)<br>• `Communication Path - Timeout`: 1s |

## Create the `HMI Communication` Ranges

| Step | Action | Comment |
|---|---|---|
| 1 | Create the `HMI Communication` ranges. Right-click the channel and select `Open HMI Communication` from the context menu. | Press the `Default` button to create the address range. |

## Do the I/O Mapping for Process Cell `Ingredients`

| Step | Action |
|---|---|
| 1 | Open the variables of the Control Module `VA21`. |
| 2 | Open the digital input module `140-DDI-353-00` of your PLC. |
| 3 | Arrange the both windows horizontally. |
| 4 | Select the `IO_PLC` input variables |
| 5 | Drag the variable and drop them within the PLC module. |
| 6 | Do the same with the output variables into the output module `140-DDO-353-00`. |
| 7 | Map also the input and output variables of the Control Modules `MO21, VA23` and `VA24`. |
| 8 | Map the `IO_PLC` input variables of the Control Modules `FM21` and `LI11` to the analog input module 140-ACI-030-00. |

## Do the I/O Mapping for Process Cell `Mixers`

| Step | Action |
|---|---|
| 1 | Map the input variables of the Control Modules `VA01, MO11, VA21` and `VA24` to the first digital input module (`170-ADI-350-00`) of the `Ethernet I/O_Drop 2`. |
| 2 | Map the output variables of the Control Modules `VA01, MO11, VA21` and `VA24` to the first digital output module (`170-ADO-350-00`) of the `Ethernet I/O_Drop 2`. |
| 3 | Map the IO_PLC input variables of the Control Modules `FM01, LI11, TI11` and `FM21` to the analog input module `170-AAI-030-00`. |

## Create the Interlocks for the Process cell Ingredients

The valve VA21 has to be closed / cannot be opened, if the level of the tank is less than 10l.

The pump MO21 can only start, when VA21 and VA23 or VA24 are opened. The level of the tank has to be greater than 100l

Valve VA23 can only be opened if valve VA24 is closed and vice versa.

| Step | Action | Comment |
|---|---|---|
| 1 | Right-click on the control module VA21 in the Physical Model and select Open Interlock from the context menu. | Set the interlock (TA01A_LI11_PV < 10.0). |
| 2 | Create also the interlocks for the control modules MO21, VA23 and VA24. | Interlocks to be created:<br>• MO21:<br>NOT (TA01A_VA21_QOpen AND (TA01A_VA23_QOpen OR TA01A_VA24_QOpen) AND (TA01A_LI11_PV > 100.0))<br>• VA23:<br>(TA01A_VA24_QOpen = 1)<br>• VA24:<br>(TA01A_VA23_QOpen = 1) |

## Create the Interlocks for the Process Cell Mixers

VA01 has to close, if the level is above 1000l. MO11 can only run, if every valve is closed, the level is higher than 200 l and the temperature is greater than 30° C.

MO21 can only run, if VA21 is opened.

VA21 has to close, if the level is below 40l.

| Step | Action | Comment |
|---|---|---|
| 1 | Right-click on the Control Module VA01 in the Physical Model and select Open Interlock from the context menu. | Set the interlock (MI01A_LI11_PV > 1000.0). |
| 2 | Create also the following interlocks for the Control Modules MO11, VA21 and MO21. | Interlocks to be created:<br>• MO11:<br>NOT (SO01A_VA01_QOpen AND SI01A_VA21_QOpen AND (MI01A_LI11_PVR > 200.0) AND (MI01A_TI11_PVR > 30.0)))<br>• MO21:<br>(SI01A_VA21_QOpen = 0)<br>• VA21:<br>(MI01A_LI11_PV < 40.0) |

## Create a Variable

| Step | Action | Comment |
|---|---|---|
| 1 | Right-click on the equipment module `TA01A` in the Topological Model and select `Open Variables` from the context menu. | Right-click in the `Variables dialog box` and select `New` from the context menu to create a variable with the following settings.<br>• `Basic` tab<br>  • `Name`: Disable<br>  • `Description`: Disable Operation<br>  • `Connection Type`: PLC_HMI<br>  • `Data Type`: BOOL<br>  • `Initial Value`: 0<br>  • `Update Initial Value`: Not_Assigned<br>Click `Apply` to get further property tabs.<br>• `Communication` tab<br>  • `Communication Frame`: RW_ANY (RD&EW)·<br>• `Alarm` tab<br>  • `Alarm`: Yes<br>  • `Alarm Group`: AllAlarms-General Alarm Group<br>  • `Alarm Priority`: 2 - Medium Priority<br>  • `Alarm Limit`: 1 |

## Create Links for Process Cell `Ingredients`

| Step | Action |
|---|---|
| 1 | Right-click on the Control Module `CTRL` in the Physical Model and select `Open Link` from the context menu. |
| 2 | Drag the variable `Disable` from the variable window to the pin `Err` in the link window of the Control Module `CTRL`. |
| 3 | Double-click on the Control Module `CTRL` to open the variable dialog box. |
| 4 | Right-click on the Control Module `VA21` in the Physical Model and select `Open Link` from the context menu. |
| 5 | Drag the variables `Auto` and `Man` from the variable window of the module `CTRL` to the pin `Aut` and `Man` in the link window of the module `LI11`. |
| 6 | Create also the links for the Control Modules `VA21`, `MO21`, `FM21`, `VA23`, `VA24`. |

# Generate the Project

## Introduction

This section describes what the user has to do to generate the project `demoplant.osp` correctly.

## Requirements

The following applications have to be installed correctly:

- Unity Application Generator and its sub programs
- Unity Pro
- Vijeo Citect, Monitor Pro or iFIX

## Context Menus

Some dialogs have context menus. The user has to click with the right mouse button into the dialog to open the context menu.

## Detected Errors / Alerts

The user can find out if his application will generate correctly by using the analyze function of UAG. There are three different states of information from the analyzer or generator.

1. Detected Error (E)
   A detected error occurs. The user has to correct the program. The generation of the program stops or cannot be started.
2. Alert (W)
   Some settings are missing. The generation of the program continues.
3. Information (I)
   These are information about the program or generation state. The generation of the program continues.

## Project Generation

After the creation of the parts of the UAG project and its settings, the user has to do the following steps to generate the different programs for the PLC and HMI.

| Step | Action | Comment |
|---|---|---|
| 1 | Open the project exclusively. | Open the project exclusively before starting the generation. |
| 2 | Generate the PLC program. | Start the program generation for the PLC with **Generate →PLC**<br>**NOTE:** The generation of the PLC program can take a few minutes. |
| 3 | Generate the HMI program. | Start the program generation for the HMI with **Generate →HMI**<br>**NOTE:** The generation of the HMI program can take a few minutes. |

# Chapter 4
# Starting the Vijeo Citect Demo Application

## Final Steps for Vijeo Citect

### Introduction

The following information contains the steps for starting the demo program for Vijeo Citect and its sub programs.

### Starting Sequence

The starting sequence for the Vijeo Citect demo application is:

1. Unity Pro and its simulation
2. Citect Project Editor
   To pack, compile and run the project.

**NOTE:** To avoid incorrect startup, it is necessary to start the application in this sequence.

### Unity Pro

The following table shows the steps for Unity Pro.

| Step | Action | Comment |
|---|---|---|
| 1 | Open the Unity Pro application. | See under: **Start** →**All Programs** →**Schneider Electric** → **SoCollaborative** →**Unity Pro** →**Unity Pro XL** (for example).<br>You can also run Unity Pro from the UAG graphical interface. |
| 2 | Open the `plc01.stu` file. | Open the file within Unity Pro. You will find the file under `c:\demoproj\plc01`. |
| 3 | Open the `Set Address` window. | Open: **PLC** →**Set Address**. |
| 4 | Set the address parameters. | Set:<br>• `Simulator`<br>  • `Address`<br>  127.0.0.1<br>  • `Media`<br>  TCPIP |
| 5 | Enter the simulation mode. | Set: **PLC** →**Simulation Mode**. |
| 6 | Connect to PLC (Simulator). | Set: **PLC** →**Connect**. |

| Step | Action | Comment |
|---|---|---|
| 7 | Load the application to the PLC and start. | Set: **PLC →Transfer Project to PLC**.<br>Select **PLC Run after Transfer** and click **Transfer**. In the following dialog box click **Rebuild All and Transfer**. |
| 8 | Confirm Run on this Project. | A window is going to open to confirm the run of the project. Enter `OK`.<br>The PLC simulator is running now. You will get the information from the status bar at the bottom of the Unity Pro application. |

## Vijeo Citect Project Editor

The following table shows the steps for the Vijeo Citect Project Editor.

| Step | Action | Comment |
|---|---|---|
| 1 | Verify that the latest PLC project is loaded to the simulator. | - |
| 2 | Open the **Vijeo Citect Project Editor**. | See for example under: **Start →All Programs →Schneider Electric →SoCollaborative →Vijeo Citect 7.20 →Vijeo Citect Explorer**. |
| 3 | Set the project ID of each project in Citect unique before compiling. | See under: **Project Properties** of each project.<br>Then switch to the **Vijeo Citect Project Editor** window. |
| 4 | Pack each HMI project, generated in UAG, in the **Vijeo Citect Project Editor**. | See under: **File →Pack**.<br>**NOTE:** This step may already be done by the generator according to the settings of the **Post Build Activities** parameter. |
| 5 | Add a test user to the HMI project. | Configure **Privilege 8** for the test user. |
| 6 | Compile the project(s). | See under: **File →Compile**. |

## Start the Vijeo Citect HMI System

| Step | Action |
|---|---|
| 1 | Run **Citect Computer Setup Wizard**. |
| 2 | Use **Custom Setup**. |
| 3 | Select **Stand-Alone Computer**. |
| 4 | Select the HMI project, generated in UAG, to be executed. |
| 5 | On the last page of the Startup Functions Setup enter *LoadPreSets* as **Startup Cicode Function**. |
| 6 | Run project (**F5**). |
| 7 | Watch the animation and try to interact. |

# Chapter 5
# Starting the Monitor Pro Demo Application

## Final Steps for Monitor Pro

### Introduction

The following information contains the steps for starting the demo program for Monitor Pro and its sub programs.

### Starting Sequence

The starting sequence for the Monitor Pro demo application is:

1. Unity Pro and its simulation.
2. Monitor Pro Configuration Explorer and start of the Data Server.
3. Monitor Pro Client Builder.

**NOTE:** To avoid incorrect startup it is necessary to start the application in this sequence.

### Unity Pro

The following table shows the steps for Unity Pro.

| Step | Action | Comment |
|---|---|---|
| 1 | Open the Unity Pro application. | See under: **Start** →**All Programs** →**Schneider Electric** → **SoCollaborative** →**Unity Pro** →**Unity Pro XL** (for example). You can also run Unity Pro from the UAG graphical interface. |
| 2 | Open the `plc01.stu` file. | Open the file within Unity Pro. You will find the file under `c:\demoproj\plc01`. |
| 3 | Open the `Set Address` window. | Open: **PLC** →**Set Address**. |
| 4 | Set the address parameters. | Set:<br>• `Simulator`<br>  • `Address`<br>  127.0.0.1<br>  • `Media`<br>  TCPIP |
| 5 | Enter the simulation mode. | Set: **PLC** →**Simulation Mode**. |
| 6 | Connect to PLC (Simulator). | Set: **PLC** →**Connect**. |

| Step | Action | Comment |
|---|---|---|
| 7 | Load the application to the PLC and start. | Set: **PLC →Transfer Project to PLC**.<br>Select **PLC Run after Transfer** and click **Transfer**. In the following dialog box click **Rebuild All and Transfer**. |
| 8 | Confirm Run on this Project. | A window is going to open to confirm the run of the project. Enter `OK`.<br>The PLC simulator is running now. You will get the information from the status bar at the bottom of the Unity Pro application. |

## Monitor Pro Configuration Explorer

The following table shows the steps for the Monitor Pro Configuration Explorer.

| Step | Action | Comment |
|---|---|---|
| 1 | Open the Monitor Pro Configuration Explorer. | See under: **Start →Programs →Monitor Pro →Configuration Explorer**. |
| 2 | Expand the tree view to your computer. | Expand the tree view to: **Enterprise View →FactoryLink Servers → Computer Name**.<br>**NOTE:** Computer Name is the name of the current computer. In that case the Data Server is running on the local computer. Other cases are possible. |
| 3 | `Add Existing Application...` | Select the computer name and click the right mouse button to open the dialog. Select `Add Existing Application...`. |
| 4 | Enter the `Required Parameters`. | Set:<br>• `Required Parameters`<br>  • `Name`<br>  ds01<br>  • `Application Directory`<br>  c:\demoproj\ds01 |
| 5 | `Confirm Action`. | You will get an information about the path of your application directory. You have not a UNC path specified.<br>Please confirm the action with `Yes`. |
| 6 | Start the Data Server. | Expand the tree view to: **Enterprise View →FactoryLink Servers → Computer Name →ds01**<br>**NOTE:** Computer Name is the name of the current computer. In that case the Data Server is running on the local computer. Other cases are possible. |
| 7 | `Start / Stop` | Select `ds01` and click the right mouse button to open the dialog. Select `Start / Stop` and start the Data Server.<br>**NOTE:** You will get an information about the run time, if you have a demo licence, please confirm. |
| 8 | Status information and `FactoryLink Run-Time Monitor`. | You will get many status information about the start of the Data Server within the `Configuration Explorer` window.<br>Additionally to that, the `FactoryLink Run-Time Monitor` was started. That indicates to you that the Data Server is running. |

## Monitor Pro Client Builder

The following table shows the steps for the Monitor Pro Client Builder.

| Step | Action | Comment |
|---|---|---|
| 1 | Open the Monitor Pro Client Builder. | See under: **Start →All Programs →Monitor Pro →Client Builder**. |
| 2 | Open the project. | During the start, the `Monitor Pro Client Builder` wants to open a `fvp` project file.<br>Please select the `c:\Demo Plant\MonitorPro\HMI_P\HMI_P.fvp` project file.<br>**NOTE:** The file is generated automatically from UAG and has the name of your `HMI` from UAG. |
| 3 | Set the Data Server. | See under: **Tools →Servers →Clusters**.<br>Select the `OPCCluster` and change the name of it to the Data Server name.<br>● `Name`<br>ds01<br>Finish the action with `Set` and close the window. |
| 4 | Open the mimic. | Select: **File →Open**. The `Choose Mimic` window is going to open. |
| 5 | Open `picture01` | Select the mimic `picture01` and open it with the `OK` button. |
| 6 | Try the connection. | **1.** Resize the mimic to the half size of the Client Builder window.<br>**2.** Click on the valve symbol.<br>**3.** The `Command-Mode` window is going to open and select `Command-Mode`.<br>**4.** Select `Manual` mode.<br>**Result:** The color of the valve is going to blue. That means the Data Server connection is running correctly. |

# Chapter 6
# Data for the Generic HMI Demo Application

## Overview

This chapter provides information about the data generated for the Generic HMI demo application and on how to work with Extensible Style Sheets (XSL).

It requires that the steps of Creating a Demo Project *(see page 23)* have been done for the Generic HMI generation.

## What Is in This Chapter?

This chapter contains the following topics:

# Generated Files for Generic HMI

## Introduction

The following section contains information on data generated for Generic HMI.

## Generated Files

The following files are generated for a Generic HMI:

- *DemoPlantIOPoints.html*
- *DemoPlant.csv*

## Directory

The files mentioned above are stored in the directory *c:\Demo Plant\Generic\DS_DRINK*.

# Working with Extensible Style Sheets (XSL)

## Introduction

Unity Application Generator provides different style sheets for the transformation of the generated XML code, e.g. to HTML.

The user will find the style sheets under `<installation directory>\Db`.

## User-defined Style Sheets

If the user needs his own style sheets, the Customization Editor provides the possibility to integrate user-defined ones. They have to be stored within the UAG directory `<installation directory>\Db`.

The user can select the style sheets under **General →Data Format Name →Add or Modify**.

**NOTE:** The user-defined style sheets have to accord to the `w3c` standard, see also `www.w3c.org`.

## Change the Style Sheet Path

The user can change the path where the transformation process will find the style sheets. The path will be selected within the Customization Editor under **General →Path →XML Style Sheets**.

**NOTE:** If the path has been changed, the necessary XSL files have to be copied into the new directory.

# Appendices

# Appendix A
# iFIX Demo Application

## Overview

This chapter provides information about the iFIX demo application.

## What Is in This Chapter?

This chapter contains the following topics:

# General

## Testing and Verification

**⚠ WARNING**

**UNINTENDED EQUIPMENT OPERATION**

Test and verify the operation of applications before release.

Perform safety analysis and tests appropriate to the application. This software package provides a means to quickly design an application. System programming and interlocks are required to ensure the safety of the resulting applications.

Any applicable regulatory requirements must be met before releasing an application for general use.

**Failure to follow these instructions can result in death, serious injury, or equipment damage.**

## Programming Environment

Intellution iFIX 3.5 or 4.0, incl. MBT driver V7.156k for iFIX from Dimension Software Inc. A demo version of the MBT driver can be found under \AddOns\MBT IFix TCPIP Driver on the UAG DVD.

UAG is able to generate for iFIX 4.0 but the new features introduced with iFIX 4.0 are not supported. Newer iFIX versions are not tested. UAG iFIX connector does not support most of the new features introduced with UAG 3.2 and later versions.

## Using the Demo Project

Copy the files of the prepared demo projects as shown in the table below.

The following table shows the steps to copy the sample files from DVD to the working folder.

| Step | Action | Comment |
|---|---|---|
| 1 | Copy the files and folder structure | You have to copy the whole folder structure with the files from DVD (*\Examples\Demo Plant\iFIX*) to your harddisk (*c:\Demo Plant)* |
| 2 | Clear the read-only attribute. | Select the **c:\Demo Plant** folder and clear the read-only attribute (when asked, choose **Apply changes to this folder, subfolders and files**). |

# How to Configure iFIX for Use with Unity Application Generator

## Introduction

Before you use iFIX with Unity Application Generator, you have to configure iFIX in a certain way. The steps for this configuration are described below.

**NOTE:** Intellution recommends that the iFIX node name and the computer name are the same. The iFIX node name is specified as part of the installation of iFIX.

After the installation the Modbus Plus/Ethernet Driver MBT V2.0 has to be installed.

## How to get the MBT Driver?

The MBT driver is not a product of the Schneider Electric company. You have to contact Dimension Software Inc. to order the driver.

- Phone
  +1 (828) 635-7189
- Fax
  +1 (828) 625-5319
- Web
  http://www.caro.net/dsi/dsidrvrs.htm
- e-Mail
  dsi@caro.net

The part number of the driver is `MBT`, the description is `MODBUS TCP/Plus OPC Server`.

## Procedure for iFIX Configuration

For iFIX configuration for Unity Application Generator follow the steps:

| Step | Action |
|---|---|
| 1 | Run the system configuration from **Start** →**All Programs** →**iFIX** in the task bar. |
| 2 | Select **Configure** →**SCADA.** |
| 3 | Switch SCADA support to **enable**. |
| 4 | In the I/O driver name click the **?** button. |
| 5 | From the list of available drivers choose **Modbus Plus/Ethernet Driver MBT V2.0**. |
| 6 | Confirm with **OK**. |
| 7 | Click the **Add** button.<br>**Result**: The driver appears in the configured I/O drivers list. |
| 8 | Confirm with **OK**.<br>**Result**: You will be asked to answer the question: **Database DATABASE does not currently exist, or is not in your database directory, use anyway?** |
| 9 | Answer with **Yes**. |
| 10 | Save with **File** →**Save**. |
| 11 | End the configuration with **Exit**. |

# The iFIX System Configuration Utility (SCU)

## Introduction

The user has to configure different settings for iFIX system.

## iFIX System Configuration Utility (SCU)

The following table shows the steps to configure the iFIX system.

| Step | Action | Comment |
|---|---|---|
| 1 | Open the `System Configuration Utility` | Open **System Configuration** from within the iFIX Workspace, so you will later be able to configure Security Areas and alarm areas. |
| 2 | Change the configuration's description. | Open **File** →**Description** and replace **...FIX** by **...DS01**. |
| 3 | Configure the SCADA. | Open the SCADA configuration dialog under: **Configure** →**SCADA ...**.<br>● `SCADA Support`<br>`enable`<br>● `Database Name`<br>`FIX`<br>● `I/O Driver Definition`<br>Choose the MBT driver by clicking the `?` button and add the selection.<br>Click **OK**. When asked whether to use the non-existent database `FIX` choose **YES**. |
| 4 | Enter the security configuration. | Open the Security Configuration window under: **Configure** →**Security ...**. |
| 5 | Open the autologin configuration. | Open the autologin configuration dialog under: **Edit** →**Autologin ...**. |
| 6 | Enter the node name and users. | Add a new configuration.<br>● `Node`<br>ds01<br>● `Application User`<br>`SYSTEM ADMINISTRATOR`<br>● `System User`<br>`SYSTEM ADMINISTRATOR` |
| 7 | Change the security areas. | Open the security area dialog under: **Edit** →**Security Areas...**. |

| Step | Action | Comment |
|---|---|---|
| 8 | Modify the default names. | Modify the default names into:<br>● `Security Areas`<br>  ● `1`<br>  CD01_1<br>  ● `2`<br>  CD01_2<br>  ● `3`<br>  CD01_3<br>  ● `4`<br>  CD01_4<br>  ● `5`<br>  CD01_5<br>Close the **Security Configuration** dialog. |
| 9 | Open the task configuration. | Open the tasks configuration dialog under: **Configure** →**Tasks ...**. |
| 10 | Enter the tasks. | Enter the following tasks and sequence:<br>● `Configured tasks`<br>  % C:\DYNAMICS\IOCNTRL.EXE /a<br>  % C:\DYNAMICS\WSACTASK.EXE<br>  C:\DYNAMICS\WORKSPACE.EXE |
| 11 | Open the alarm area database. | Open the alarm area database configuration dialog under: **Configure** → **Alarm Area Database**. |
| 12 | Add the alarm area. | Enter the name `cd01` and add it. The alarm area name is equivalent to the Control Domain name in the UAG project. |
| 13 | Configure the Local Startup. | Open the Local Startup dialog under: **Configure** →**Local Startup ...**. |
| 14 | Enter the local names. | Enter the following names:<br>● `Local Node Name`<br>  ds01<br>● `Local Logical Name`<br>  ds01 |
| 15 | Restart iFIX. | Note that now the `Node Name` field reads `ds01`. |

# Project Generation

## Introduction

After the creation of the parts of the UAG project and its settings, the user has to do the following steps to generate the different programs for the PLC and HMI.

| Step | Action | Comment |
|---|---|---|
| 1 | Start the iFIX environment. | For the generation of the iFIX HMI files you have to start the iFIX environment first (**Start** →**Programs** →**iFIX** →**Intellution iFIX**). Follow the steps described in The iFIX System Configuration Utility (SCU) *(see page 59)*. |
| 2 | Open the project exclusively. | Open the project exclusively before starting the generation. |
| 3 | Generate the PLC program. | Start the program generation for the PLC with **Generate** →**PLC**<br>**NOTE:** The generation of the PLC program can take a few minutes. |
| 4 | Generate the HMI program. | Start the program generation for the HMI with **Generate** →**HMI**<br>**NOTE:** The generation of the HMI program can take a few minutes. |
| 5 | Reload the database. | Double-click the **Database Manager** node in iFIX workspace. Choose **Open Local Node** and then **Database** →**Reload**. Double-click the node **ds01** and close **Database Manager**. |
| 6 | Start the MBT driver. | Follow the steps described in Settings for the MBT Driver and PowerTool *(see page 63)*. |

# Final Steps for iFIX

## Introduction

The following information contains the steps for starting the demo program for iFIX and its sub programs.

## Starting Sequence

The starting sequence is:

1. Unity Pro and its simulation
2. iFIX Workspace
3. MBT PowerTool (start before switching iFIX Workspace to run mode!)

**NOTE:** To avoid incorrect startup, it is necessary to start the applications in this sequence. Also, the iFIX Workspace has to run for UAG to be able to generate the HMI.

## Unity Pro

The following table shows the steps for Unity Pro.

| Step | Action | Comment |
|---|---|---|
| 1 | Open the Unity Pro application. | See under: **Start** →**All Programs** →**Schneider Electric** → **SoCollaborative** →**Unity Pro** →**Unity Pro XL** (for example).<br>You can also run Unity Pro from the UAG graphical interface. |
| 2 | Open the `plc01.stu` file. | Open the file within Unity Pro. You will find the file under `c:\demoproj\plc01`. |
| 3 | Open the `Set Address` window. | Open: **PLC** →**Set Address**. |
| 4 | Set the address parameters. | Set:<br>• `Simulator`<br>  • `Address`<br>  127.0.0.1<br>  • `Media`<br>  TCPIP |
| 5 | Enter the simulation mode. | Set: **PLC** →**Simulation Mode**. |
| 6 | Connect to PLC (Simulator). | Set: **PLC** →**Connect**. |
| 7 | Load the application to the PLC and start. | Set: **PLC** →**Transfer Project to PLC**.<br>Select **PLC Run after Transfer** and click **Transfer**. In the following dialog box click **Rebuild All and Transfer**. |
| 8 | Confirm Run on this Project. | A window is going to open to confirm the run of the project. Enter `OK`.<br>The PLC simulator is running now. You will get the information from the status bar at the bottom of the Unity Pro application. |

# Settings for the MBT Driver and PowerTool

## Introduction

The iFIX SCADA systems is using the MBT driver for the communication. The user has to take care that iFIX and the MBT driver is installed correctly.

## Settings within the PowerTool

The following table shows the steps of the PowerTool settings for the communication.

| Step | Action | Comment |
|---|---|---|
| 1 | Start the PowerTool | You can start the PowerTool under: **Start →All Programs →iFIX → MBT PowerTool**. |
| 2 | Select `Use Local Server` | Select `Use Local Server` and `Connect...` to the server.<br>The computer will be connecting to the server. |
| 3 | Enable `Channel_1`. | Select `Channel_1` in the tree view.<br>Select the check box and enable `Channel_1` |
| 4 | Change the `Default configuration file name`. | Open **Options →Setup** and select the register `Default Path`.<br>Change the name of the `Default.MBT` to `ds01.MBT`. |
| 5 | Change the modes. | Select the register `Advanced` and set:<br>● `Simulation Mode`<br>`Off`<br>● `Auto Create`<br>`On`<br>● `Auto Start`<br>`Off` |
| 6 | Start the driver. | **Display Mode →Start** |
| 7 | Change to `Stats Mode` | Change from `Config Mode` to `Stats Mode` to control the server connection under **Display Mode →Stats Mode**. |
| 8 | Control the connection state. | You can control the connection to the Data Server by the `Data Quality` within the `Stats Mode`. The `Data Quality` of the subnodes of `CH1` has to be `Good`. |

# Run the iFIX Workspace

## Introduction

To start the generated application, the user has to open the iFIX Workspace.

## Start the Application

The following table shows the steps to start the generated application.

| Step | Action | Comment |
|---|---|---|
| 1 | Start the startup window. | You can start the startup window of iFIX under: **Start →All Programs →iFIX →Intellution iFIX**. |
| 2 | Start the application. | Use the iFIX button to start the administrated application.<br>● `iFIX`<br>`Start iFIX with these settings:` |
| 3 | Reload the database. | Double-click the **Database Manager** node in iFIX Workspace.<br>Choose **Open Local Node** and then **Database →Reload**.<br>Double-click the node **ds01** and close **Database Manager**. |
| 4 | Open the `equip01` picture and `Switch to Run`. | See under: **Workspace →Switch to Run**. |

# Index

## T



## U



## V



## W



## X