

# Integration of Spectrum Controls Quantum Modules in Unity

**Unity Version:** UNITY XL, UNITY L V2.1

**Date:** June 1, 2005

**Catalog Numbers:** 140 AUI 040 00sc  
140 ACI 051 00sc  
140 ACI 052 00sc  
140 EHC 208 00sc

## Introduction

When using Concept with Quantum I/O the user is able to import third party modules into the Concept configuration database. Unity Pro V2.1 offers the same capability with third-party modules. This document explains the steps necessary to incorporate Spectrum Controls Quantum modules into Unity.

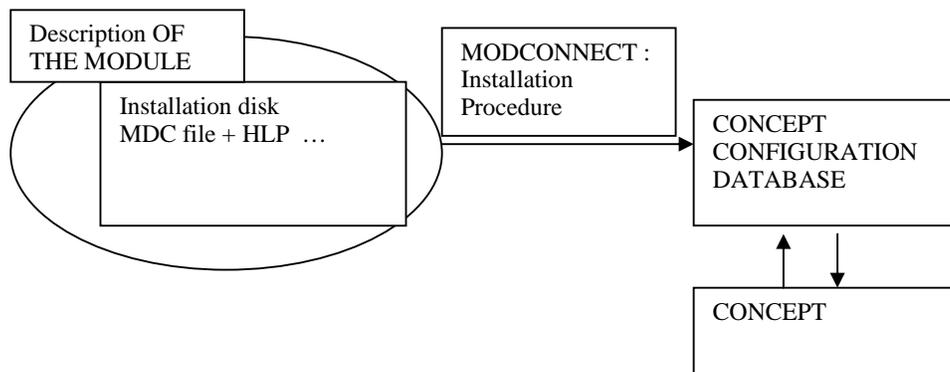
## Background: Using the ModConnect Tool

The method of integrating Quantum I/O into Unity is similar to the installation of a module in Concept. In order to incorporate the module into Concept you must use the ModConnect tool and a Spectrum Controls supplied MDC text file. This file describes the parameters for the module and the default values.

Example of a Spectrum I/O MDC file:

```
[MDC-AVI-050-00SC]
Version           = 1.0
Description       = AN IN 32CH Voltage
InputBytes       = 18
OutputBytes      = 2
ModuleFamily     = 2
ModuleBits       = 12
ModuleId         = 308
DialogId         = 4095
DefaultModuleData = 8192,10898,517
ModuleHepId      = 1
DialogHelpId     = 2
```

After the file has been imported into the Concept configuration database the module may be used.



## Incorporating I/O into Unity

Unity does not support specific module profiles. Instead there are three « generic » profiles that may be used to describe an I/O:

- Family ANALOG GEN ANA IO => 1 generic analog IO (maximum 16 in/out)
- Family DISCRETE GEN IO => 1 generic discrete IO (maximum 64 in/out)
- Family NETWORK GEN NOM => 1 generic NOM (nothing specific except NOM flag set)

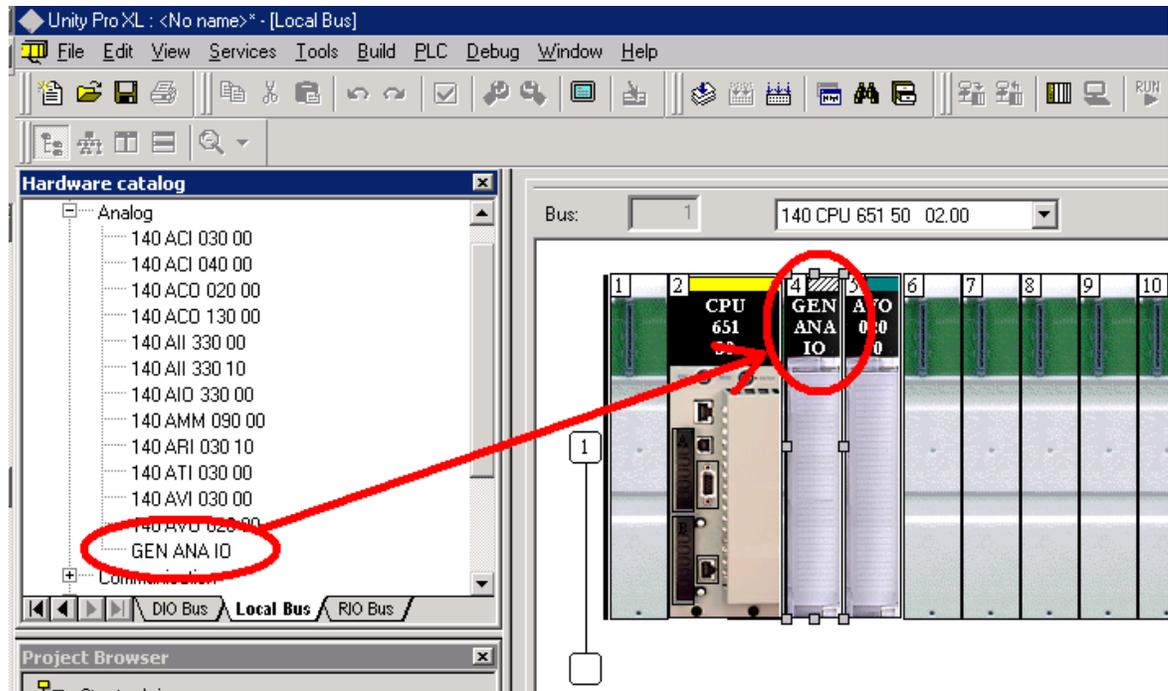
The ANALOG generic profile is typically used for Spectrum Controls I/O.  
 (Note that loadable functions and dialog boxes are not supported with these profiles.)

Module	Register parameters	Variables available
Generic Analog I/O	20 Words maximum	%IW1 to %IW16 %QW1 to %QW16
Generic Discrete I/O	2 words maximum	%I1 to %I64 %Q1 to %Q64 or %IW1 to %IW8 %QW1 to %QW8 (depending on chosen mapping register or bit)
Generic NOM	0	Not Applicable

## Module Installation

Three steps are necessary to use a generic module:

**Step 1:** The user chooses a generic module in the Unity Pro configuration editor.



**Step 2: Module Settings:**

The user has to enter specific parameters found in the I/O MDC file in order to define the module attributes.

The following information has to be entered:

- Module Identification
- Module Settings

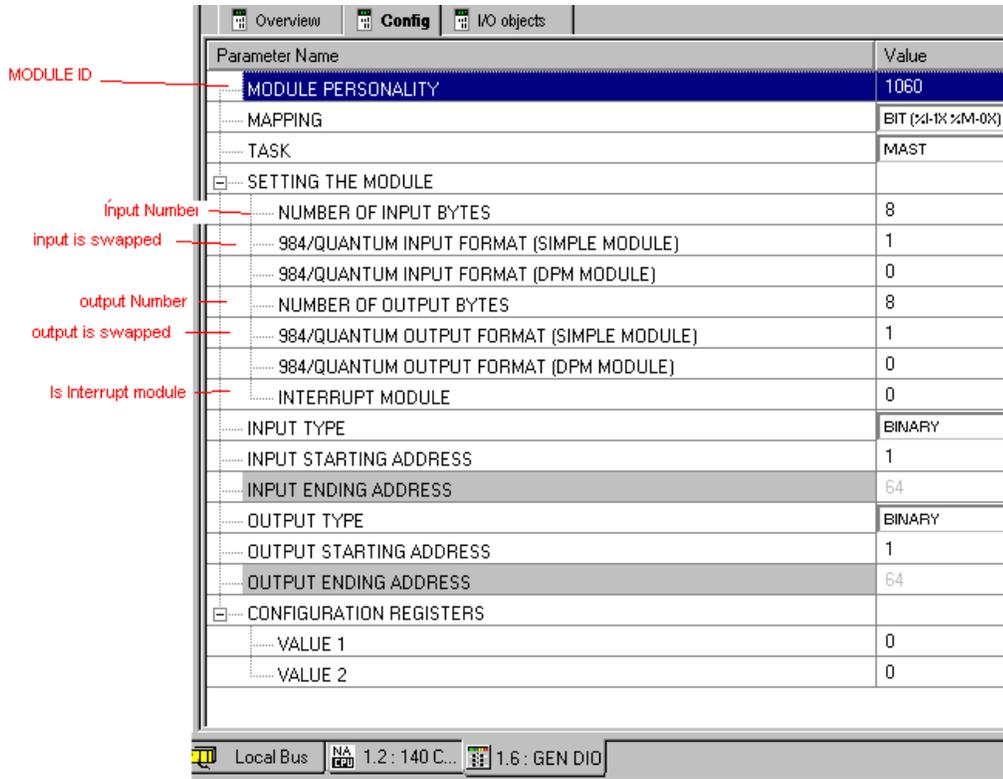
The table below describes the correlation between the MDC file fields and the generic editor.

MDC Fields	Correspondence in unity I/O module editor
Version = 1.0	Not Applicable
Description = AN IN 32CH Voltage	Not Applicable
InputBytes = 18	Number of Input Bytes
OutputBytes = 2	Number of Output Bytes
ModuleFamily = 2	Not Applicable
ModuleBits = 12 ModuleBits is a bit field mask coded as a decimal number. The mask definition is as follows: Bit 0 Input is BCD Bit 1 Output is BCD Bit 2 Input is swapped (984 /quantum input Format simple module ) Bit 3 Output is swapped (984/ quantum input Format simple module ) Bit 4 Reserved for future use Bit 5 Hold last value (same name in editor) Bit 6 Is Interrupt Module (same name in editor) Bit 7 Input Format (for use with Dual Port Modules only) Example 12 = 1100 I/o are swapped ( input / output format =1 in unity editor )	Bit 0 Input is BCD => (same name in editor) Bit 1 Output is BCD=>(same name in editor) Bit 2 Input is swapped => (984 /quantum input Format simple module ) Bit 3 Output is swapped =>(984/ quantum input Format simple module ) Bit 4 Reserved for future use Bit 5 Hold last value =>(same name in editor) Bit 6 Is Interrupt Module => (same name in editor) Bit 7 Input Format (for use with Dual Port Modules only) =>(984/ quantum input Format DPM module <b>The result must be tested see the note)</b>
ModuleId = 308	Module Identification
DialogId = 4095	Not Applicable
DefaultModuleData = 8192,10898,517	To be entered Manually in configuration register SEE STEP 3
ModuleHepId = 1	Not Applicable
DialogHelpId = 2	Not Applicable

**NOTE:**

DPM Input format = Swap the order of bits in the words

984 /input format = Swap the order of BYTES in the word (LSB/MSB)

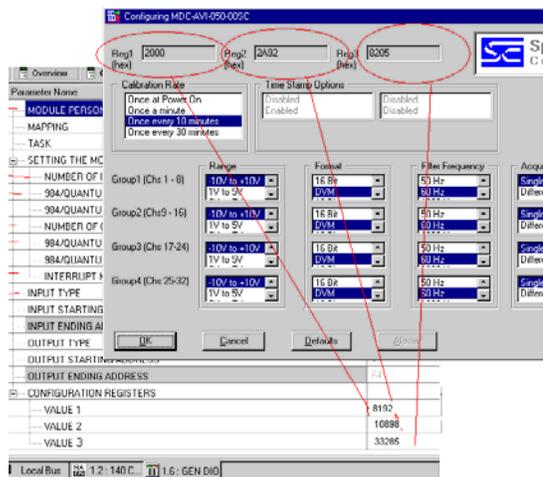


**Step 3:** Configuration of the module:

A maximum of 20 configuration registers are necessary to fill in the I/O configuration editor. The user must fill the configuration parameters in word decimal notation. Refer to the I/O user manual for specific configuration settings.

Spectrum Controls provides a configuration utility, SIOCONFIG, that uses a simple graphical user interface to convert module settings into the proper configuration format. This tool along with I/O MDC files are available on our website: <http://www.spectrumcontrols.com/schneider.htm>

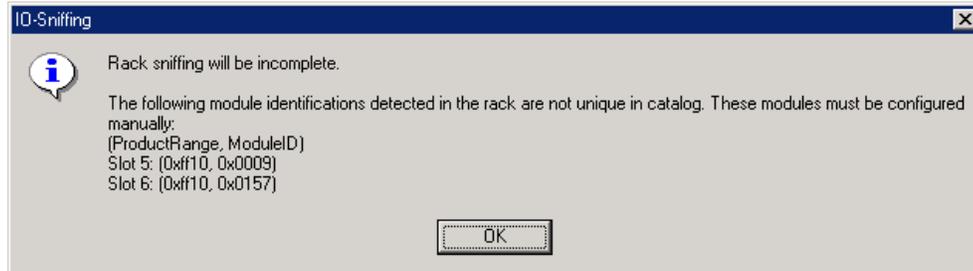
All data that is assigned in the generic configuration screen must be manually copied from the MDC file.



## Limitations and Precautions

- 1) You may not add a module when in the online mode
- 2) The loadable functions are no longer supported. They must be replaced by EF /EFBs generated using the Unity EFB toolkit.
- 3) IO -SNIFFING functionality is no currently functional for generic modules:

You may receive the following message:



### In the case of incorrect declarations:

The number of input /output bytes defined must match those with the physical module.

If the total number of I/O registers declared exceeds the number of physical module I/O registers the configuration will be rejected.

For additional information, sample ladder logic, and MDC files visit [www.spectrumcontrols.com](http://www.spectrumcontrols.com).

Spectrum Controls, Inc.  
PO Box 5533  
Bellevue, WA 98005  
Ph: (425) 746-9481  
Email: [support@spectrumcontrols.com](mailto:support@spectrumcontrols.com)