Instrument Communication Module (ICM)

The Instrument Communication Module is an open source LabVIEW tool for reading and writing data from a wide range of devices and instruments. It is configuration based: once it has been integrated into a LabVIEW program or system, one can easily change or add the devices or instruments being communicated with just by editing the .icm configuration files. ICM includes options for dealing with a wide range of buses, a large variety of parsing options, and numerous scaling options.

The ICM Configuration Editor allows one to create and edit .icm configuration files, which define how ICM will communicate with an instrument. This editor has sections for General configuration parameters, Query definition, Bus configuration, Parsing definition, Scaling options, and Output configuration.

The General Parameters of the .icm file are the same for every ICM configuration. They include the following parameters:

|  |  |
| --- | --- |
| Data Stream Name | A descriptive name given to the overall configuration. This will often refer to the instrument that data is read or written to. |
| Version | This version number can be assigned as needed to track changes to the configuration file. |
| Author | This is a documentation field in which one can place the author or company which created the configuration. |
| Description | This is a documentation field that is used to give a description of what the configuration file is intended to do. |
| Comments | This documentation field is available for supplying additional information about how to use the configuration file, how to configure the instrument to work with the configuration file, or other information. |
| Location | This is a documentation field that can be used to note the location or project for which the configuration was created. |
| Date | This field is filled in by the ICM system. It identifies the date on which the configuration file was last edited. When a streaming .icm file is created, this field identifies the date on which the streaming file began recording. |
| Time | This field is filled in by the ICM system. It identifies the time at which the configuration file was last edited. When a streaming .icm file is created, this field identifies the time at which the streaming file began recording. |
| # or Records | This field is only used for streaming files, and contains the number of records of data that the streaming file contains. |

# LabVIEW Source Code vs Executable and Directory Structure

ICM is intended to be compatible running either inside the LabVIEW environment as source code, or as part of a built executable. Since ICM uses a plug-in architecture, the various modules must have source distributions for them to be used in a .exe. Without building source distributions, dependencies that are normally part of the LabVIEW environment will not be available to the .exe.

The main ICM directory is located at C:\OCC\ICM. (This location is coded within the ICM Main Path.vi. If you need to install ICM in a different directory, this VI will need to be edited.) Within this directory are many sub-directories containing both source code and source distributions. The main modules of ICM are Buses, Parsers, Queries, and Scalings. Each of these has two directories. The parent class for each module is located in a similarly-named directory that is singular instead of plural: e.g. Bus, Parser, Query, and Scaling. Within the directories with plural names are the child classes, or the specific implementations of each parent class. For example, the Buses directory may contain Clock Time, Modbus, NI Analog, Serial, and UDP directories, each containing a child class of Bus.

Also within the main ICM directory is a directory name Source Distributions. This is where the built source distributions of all the classes are contained, for use with executable programs. The ICM system includes internal logic to detect if it is run as source code or as an executable, which it uses to determine where to find the classes. Due to the limitations of the Source Distribution mechanism, the file hierarchy inside the Source Distributions is slightly different than it is inside the actual source folders. For example, the folder C:\OCC\ICM\Buses\Clock Time contains the file Clock Time.lvclass as well as all the VI’s and controls for that child class. However, the folder C:\OCC\ICM\Source Distributions\Buses contains the Clock Time.lvclass file, though the sub-VIs and controls are in the directory C:\OCC\ICM\Source Distributions\Buses\Clock Time. There is special code inside the ICM system to deal with this directory difference as well, depending on whether the code is run as source or in a .exe.

Within the ICM project, there is a Source Distribution build specification that builds all the modules’ parent and child classes at once. The project can be edited as needed before building these source distribution. In particular, if the computer being used for building the source distributions does not contain some toolkits that some classes require, those classes could be removed from the project before the build is executed. However, it is important to keep the overall logic and file hierarchy in place in order for ICM to work in both source code and in an .exe.