



I / G E A R

Introduction to I/Gear

CONNECTING TECHNOLOGIES



© 2009 I/GEAR, Inc. All rights reserved.

The information contained in this document represents the current view of I/GEAR on the issues discussed as of the date of publication. Because I/GEAR must respond to changing market conditions, it should not be interpreted to be a commitment on the part of I/GEAR, and I/GEAR cannot guarantee the accuracy of any information presented after the date of publication. This white paper is for informational purposes only.

I/GEAR MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

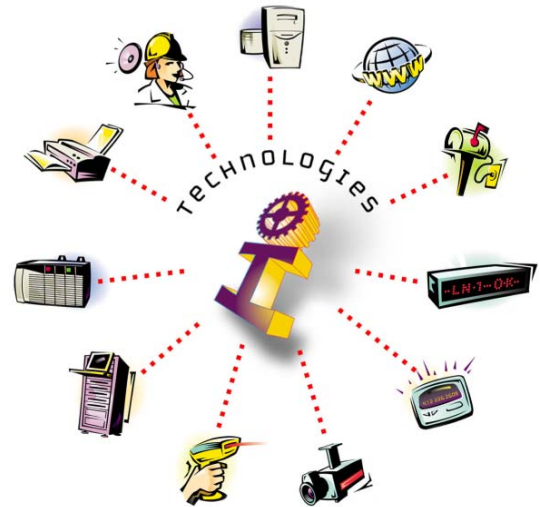
Other product and company names mentioned herein may be the trademark of their respective owner

Overview

There is an immediate need within manufacturing for sharing and distributing information between plant floor devices, Enterprise Resource Planning (ERP) systems, and other third party applications. The speed of consumer-to-business and business-to-business communications is driving a greater need for plantfloor processes to be in step with business systems. Yet a gap exists between these realms.

I/Gear is a revolutionary industrial middleware tool that bridges this gap. By enabling functional “plug and play” interchange between disparate systems, I/Gear becomes the cornerstone of an integration framework that enables a broad range of Manufacturing Execution System level solutions. I/Gear provides omni-directional connectivity between systems that include: databases, Programmable Logic Controllers, pagers, e-mail servers, serial devices and TCP/IP devices. Specific serial and TCP/IP devices may include barcode scanners, Radio-Frequency Identification systems, Andon displays, other hardware devices and third-party software applications.

I/Gear is designed using the tools and technologies of Microsoft’s Windows Distributed interNet Applications (Windows DNA) architecture for Manufacturing. Windows DNA is a strategic framework for creating, deploying and supporting distributed and scalable solutions. DNA is based on core Microsoft technologies that enable multi-vendor software applications to work together in a single distributed, scalable, enterprise-



wide “digital nervous system.” I/Gear makes use of these concepts and technologies to provide a reliable, scalable and open infrastructure for integrating systems within the manufacturing environment.

I/Gear Architecture

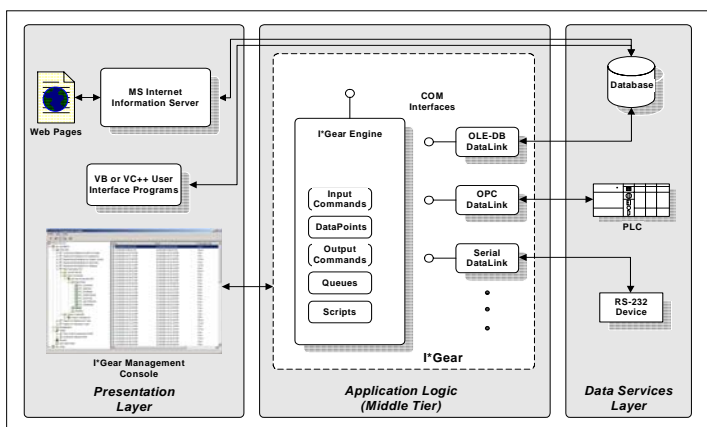
I/Gear is a multi-threaded, transaction processing application designed to run as a Windows service. Major elements of the I/Gear architecture include: the I/Gear Engine, DataLinks, and Scripts. As described in more detail below, the DataLinks provide the means for I/Gear to communicate with external systems and devices, and Scripts provide a mechanism for additional information processing capabilities.

I/Gear Engine

The core of each I/Gear node is the engine. The I/Gear Engine is a Windows service that acts as a middle-tier that contains all of the DataLink and Script components of I/Gear. The engine also provides queuing mechanisms for the DataPoints, Input Commands and Output Commands defined for each instantiated DataLink.

DataLinks

The primary components of the engine are DataLinks. DataLinks are I/Gear “Plug-ins” that understand how to communicate with a particular type of device or system. Each instance of a registered DataLink is configured to talk to a specific device or system.



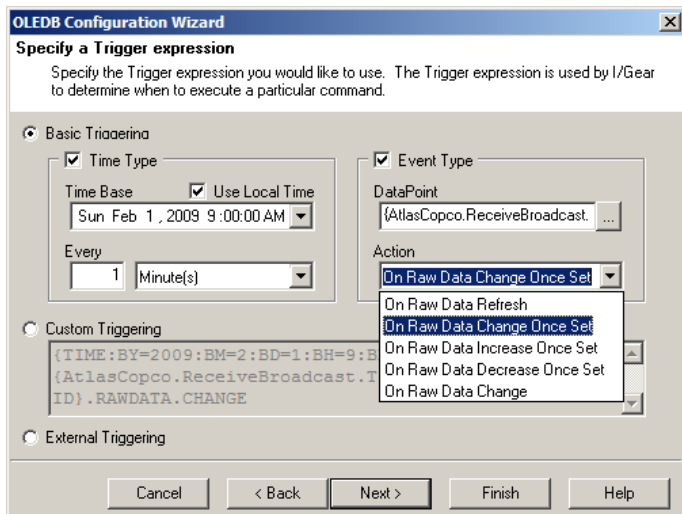
The I/Gear architecture provides a solution framework for solving real world integration challenges.

Command Transactions

Each DataLink instance is comprised of Input and Output Commands. Commands are configured to move information into or out of the engine.

Command Triggering

Command Triggers are defined to activate the execution of a particular Input or Output Command. These triggers can be defined to occur on a recurring time basis or on an event such as the change, increase, decrease, or refresh of the value of a specific DataPoint. For example, a Command Trigger could be defined to execute an Input Command every 30 seconds to read data from a PLC. Additionally, another Command Trigger could be defined to execute an Output Command to log this information to a database system on an event such as a change in the value of a DataPoint. This effectively logs data from the PLC device every 30 seconds to a remote database system.



Extensive triggering methods are available including: time based, event based, custom expressions, and external (using the I/Gear object model).

More complex triggering requirements can be met through I/Gear's Custom Triggering and External Triggering capabilities. Custom Triggering allows the user to extend the basic triggering (time and event based) with a simple VB Script expression. The External Triggering method disables the basic and

custom triggering methods and enables the execution of the command via the I/Gear object model.

DataPoints

DataPoints are used to hold information inside the I/Gear engine. DataPoints work in conjunction with Input Commands to define a single piece of data from a specific system or device. For example, if an Input Command were defined to read a block of data from a PLC, DataPoints would define the actual data elements within the block of data.

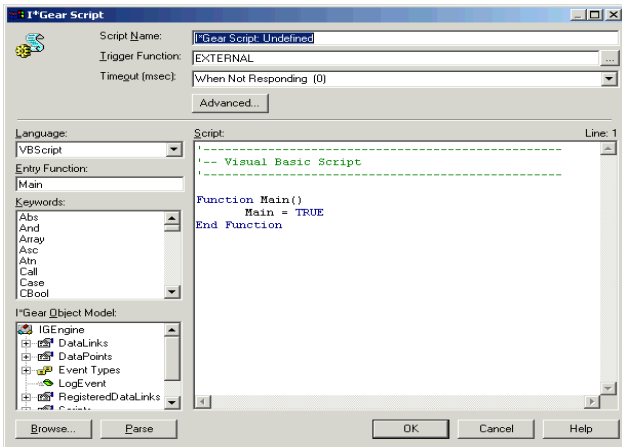
DataPoints automatically support any required data type, and also may automatically reference single values or arrays.

Another type of DataPoint is the User DataPoint. These DataPoints work in the same manner as other DataPoints except they are not bound to a specific Input Command. User DataPoints are simply used for storing and retrieving system information such as time tags or constant expressions.

Additionally, I/Gear provides a custom expression builder for developing Formulas for processing DataPoints. Each time a DataPoint is referenced in an Output Command or Script, the formula (if defined) is evaluated. The Output Commands and Scripts may reference either the DataPoint's raw data, or the Formula value.

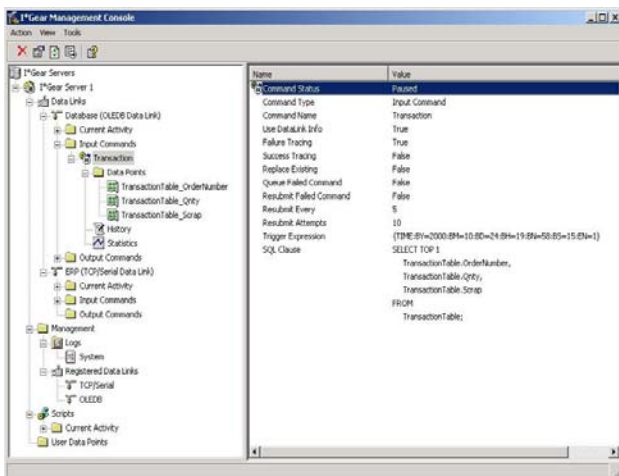
Scripting

Scripts are user-defined VBScript or JScript (Java) macros that can be used to implement the business logic that operates on the information flowing between the devices in the system. The execution of the Scripts is controlled by the same advanced triggering methods available for Input and Output Commands. Scripts have access to the full I/Gear object model.



Users may extend I/Gear functionality with VB or Java scripts that interact with a comprehensive I/Gear object model.

Scripts could be used to implement the logic necessary for tailoring I/Gear to become the core element of a specific solution for a given customer. For example, an error-proofing application could make use of a script that compares actual data collected from a PLC to expected values retrieved from a database. The success or failure in matching these values could be used to trigger an Output Command that passes information back to the PLC to enable or inhibit production.



The I/Gear Management Console provides an easy to use point and click interface for configuring and monitoring I/Gear.

Configuration and Troubleshooting

I/Gear Management Console

An I/Gear node is configured and monitored via the Management Console. The Management Console can run on any computer in the organization with network connectivity to the I/Gear Engine. The point and click Management Console interface provides the user with access to the DataLink configuration wizards and the Script editor.

Through the Management Console, users have access to command statistics and the I/Gear event log. DataPoint values and transaction history information may also be examined.

Through the Management Console, users may make on-line configuration changes that eliminate the need to pause or stop the I/Gear engine once it is running. Of course the user may pause or stop any command, any DataLink or the entire I/Gear engine as needed.

Other Features and Benefits:

OPC Compliance

OLE for Process Control (OPC) compatibility ensures operability with many available device drivers. I/Gear is fully compliant with OPC Data Access standards 1.0 and 2.0. This means that I/Gear can send and receive data to any OPC compliant server.

Openness with respect to Database Design

I/Gear may be configured to interface with any database design that is ODBC or OLEDB compliant. I/Gear is not tied to a specific database schema. I/Gear guides the user through the creation of Input and Output commands that interact with the target database.

Data Integrity

I/Gear provides advanced data queuing algorithms to guarantee that the data used by a command is valid at the instant in time that the triggering event occurs. Each DataPoint is buffered in a queue. Each reference to a DataPoint in an Output Command or Script will

receive the correct snapshot value of the DataPoint as defined by the time of validity of the trigger.

System Fault Tolerance

I/Gear is designed to provide the utmost in system fault tolerance. For example, if a command fails to execute successfully (perhaps because the external device or system is not available) the command is placed in a Failed Command Queue. The system provides for automatic retries of any Failed Command Queue items.

I/Gear may also be configured to provide a fail-over capability that allows redundant I/Gear nodes to serve in a hot backup capacity.

Extensibility

I/Gear is much more than a simple data retrieval and transaction system. Through its open COM-based extensibility interfaces, developers can write custom device modules that “plug-in” to I/Gear to allow it to work with virtually any type of system or device. Essentially any programming language capable of creating COM components (Visual Basic or Visual C++ for example) is able to access the interfaces exposed by I/Gear or create custom “plug-in” DataLinks. I/Gear essentially lets developers focus on creativity and innovation in their products, rather than on building infrastructure and services. This results in manufacturing solutions with greater functionality that are simpler to develop, faster to market, less expensive, easier to deploy and more supportable.

DataLinks

OLE-DB DataLink

The OLEDB DataLink allows I/Gear to read and write information to any SQL Compliant OLEDB provider. This also includes any database for which an ODBC driver would exist. This list includes Back-End Database Servers such as Microsoft SQL Server, Oracle, Sybase and DB2, and file based databases including Microsoft Access, Excel, FoxPro, Dbase, Paradox, and Text files.

OPC DataLink

The OPC (OLE for Process Control) DataLink allows I/Gear to read and write information to any OPC Server. OPC is the standard for communicating with plant floor control equipment including PLC’s (Programmable Logic Controllers), PC-based controllers and HMI (Human Machine Interface) Packages.

Pager DataLink

The Pager DataLink allows I/Gear to send messages to digital or alphanumeric pagers. The DataLink works with TAP (Telocator Alphanumeric Protocol) compliant pager service providers. In addition, the DataLink can emulate telephone key presses for proprietary pager services.

E-Mail DataLink

The E-Mail DataLink allows I/Gear to send and receive E-Mail and Text Messages.

TCP/Serial DataLink

The TCP/Serial DataLink allows I/Gear to read and write information to generic Serial or Ethernet (TCP/IP) devices. The DataLink allows for interactive sessions that exchange information with host systems such as an ERP system or a serial based industrial control device. The DataLink also allows for interfacing with devices such as Barcode Scanners, RFID systems and Marquees.

System Requirements

Hardware Requirements

Pentium IV class processor or greater. 1 Gb RAM or greater (2 Gb Recommended).

Operating System Requirements

Windows 2000, Windows XP Professional or Windows Server 2003.