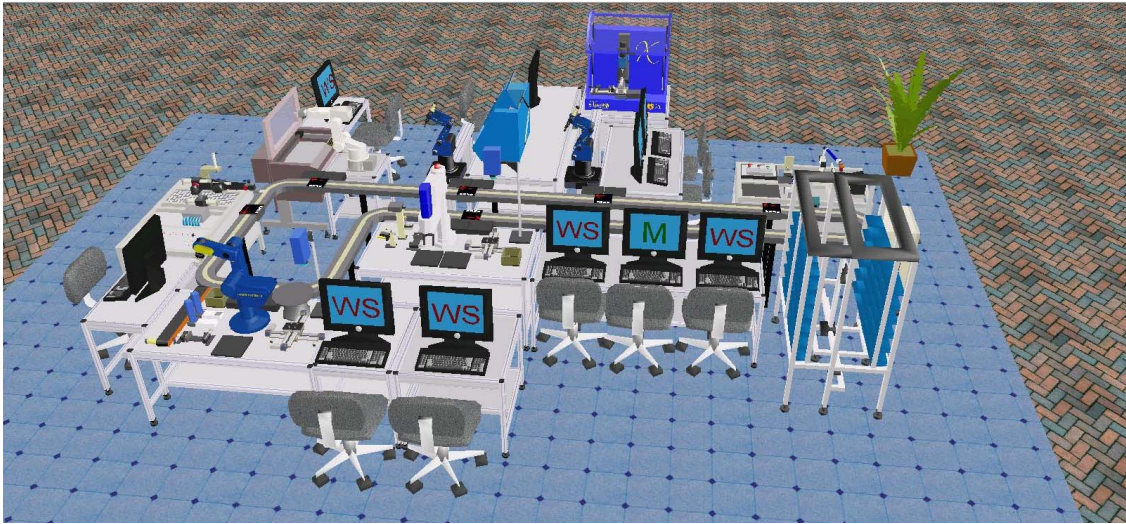# OpenCIM™

OpenCIM Offline and OpenFMS

Computer Integrated Manufacturing
for Industrial Training Applications

Software Version 4.6

User Manual

intelitek▶▶

Copyright © 2012 by Intelitek Inc.

OpenCIM User Manual

Cat.# 100094 Rev. H

April 2012

Table of Contents

1

Introduction

This chapter introduces the OpenCIM software and describes the various chapters of this manual, the information included in each chapter, how to use this manual and so on. It includes the following sections:

- **About CIM**, introduces general CIM concepts and advantages.
- **About OpenCIM**, introduces Intelitek's OpenCIM software, and describes the available OpenCIM products.
- **About This Manual**, provides the list of chapters available in this manual.
- **Who Should Use This Manual**, describes the varied target audience (such as, students, instructors and administrators).
- **How to Use This Manual**, describes which chapters are intended for which target audience.

About CIM

To stay competitive, factories are increasingly automating their production lines with Computer Integrated Manufacturing (CIM) systems. A CIM cell is an automated assembly line that uses a network of computers to control robots, production machines, and quality control devices. The CIM cell can be programmed to produce custom parts and products.

CIM provides many advantages:

- Computer integration of information gives all departments of a factory rapid access to the same production data.

- Accessibility of production data results in faster response to change, which in turn shortens lead times, increases the company's responsiveness to customer demands and competition, and improves due-date reliability.

- Computer aided scheduling optimizes the use of the shop floor. This improves the utilization of machine tools, and reduces work-in-progress and lead times.

- Real-time production data can be used to optimize the production processes to improve quality, using techniques such as statistical process control.

- Computer analysis and prediction of material requirements for production can reduce inventory levels and lead times. Integration with suppliers and customers can provide even greater benefits.

- Downloading machining instructions, including tool changes, from CAM (computer aided manufacturing) systems to CNC machines (computer numerically controlled) reduces machine setup times and increases machine utilization.

The trend among manufacturers today is to produce smaller batches of more varied products. Without CIM automation, this trend would result in higher costs associated with increased setup time and additional labor.

There is a shortage of qualified CIM technicians and engineers. Manufacturers demand graduates who understand the integration of all elements of a CIM. Intelitek's OpenCIM system addresses this need by providing an industrial-level training system for the educational environment.

About OpenCIM

OpenCIM is a system which teaches students the principles of automated production using robotics, computers, and CNC machines. It also allows advanced users to search for optimal production techniques by experimenting with different production techniques.

OpenCIM offers a simulation mode in which different production strategies can be tested without actually operating the CIM equipment.

OpenCIM provides a realistic, expandable environment through interfaces to third party hardware (CNC machines, robots, peripheral equipment, etc.). Students can learn first-hand how other disciplines such as Production Scheduling, Manufacturing Resource Planning (MRP), Order Entry Systems, and Database Management Systems (Xbase) can be used to optimize the production process.

In this version of OpenCIM, two additional OpenCIM products are also available:

- **OpenFMS** – for a small CIM system which may include a single robot and one or two CNC machines.
- **OpenCIM Offline** – a simulation only version of OpenCIM.

About This Manual

This manual is a complete reference guide to the OpenCIM system. It explains how to install, configure and operate the OpenCIM software. Indications as to which information is not relevant to the additional OpenCIM products are provided in the appropriate sections.

This manual includes complete details on how to produce custom parts, add your own computer-controlled equipment, and how to interface with other software.

How This Manual is Organized

Chapter 1   **Introduction**: Provides an introduction to OpenCIM and to this OpenCIM User Manual.

Chapter 2   **System Overview**: Describes the hardware and software components which comprise an OpenCIM cell.

Chapter 3   **Safety**: Provides the general rules, followed by a brief discussion of the safety requirements of each component.

Chapter 4   **Installation**: Describes the hardware assembly process and the software installation and configuration procedures.

Chapter 5   **Project Manager**: Describes the Project Manager application that launches the Virtual CIM Setup and CIM Manager. It enables users to manage their own projects, and administrators to manage the projects in the archive.

Chapter 6   **Operating CIM Manager**: Describes how to operate the CIM Manager which is used for operating the OpenCIM system and controlling production.

Chapter 7   **CIM Manager Utility Programs**: Describes the CIM Utility Programs which are used for preparing the OpenCIM system for production.

Chapter 8   **Virtual CIM Setup**: Describes the Virtual CIM Setup aplication which is an interactive graphic module that enables you to create a simulated CIM cell.

Chapter 9   **OpenCIM Device Drivers**: Describes the OpenCIM devices drivers, which are interface programs that translate and transmit messages between the CIM Manager and the various machines and controllers at CIM stations.

Chapter 10   **Web Viewer**: Describes the Web Viewer application enabling you to remotely access a specific CIM Manager cell and track the production cycle.

Chapter 11   **OpenCIM Programming**: Provides various programming and advanced OpenCIM features.

Chapter 12   **Inside OpenCIM**: Describes various OpenCIM administration procedures for the advanced user, and describes the OpenCIM directory structure.

Chapter 13   **Troubleshooting**: Describes device error handling, OpenCIM error messages and more.

Chapter 14   **Glossary**: Provides the abbreviations and terminology used in OpenCIM

Chapter 15   **Intelitek Software Licensing**: Describes the various procedures involved in registering your OpenCIM software.

Who Should Use This Manual

This manual is intended to be used by the following:

Students — Students can operate the OpenCIM system to gain experience with computer integrated manufacturing (CIM) or Flexible Manufacturing Systems (FMS). By working with a complete CIM system, students are encouraged to think "globally" about the manufacturing process. Students can also concentrate on a particular aspect of a CIM system such as controlling robots, CNC machines, etc.

Industrial Management Students — OpenCIM allows advanced users to implement and experiment with theories concerning optimal computer integrated manufacturing techniques such as:

- The effect of different machines which can perform the same process
- Modifying a process by changing a machine's control program
- Alternate part definitions

OpenCIM can also be used in simulation mode to search for optimal production strategies by experimenting with the following:

- The causes of production bottlenecks
- The effects of alternative production schedules
- What-if analyses

For example, OpenCIM can help answer questions such as: *Is it more efficient to do a quality control check at the end of each operation or just once at the end of the manufacturing process*?

With OpenCIM you can use a simulation mode to easily test both methods and then observe the results.

Instructors — Instructors who want to demonstrate automated production techniques using the OpenCIM system.

System Administrators — System administrators in charge of installing, maintaining, and troubleshooting the OpenCIM system will want to become familiar with all aspects of this manual.

How to Use This Manual

The OpenCIM software can be operated and used fully without OpenCIM hardware. Therefore, the emphasis in this manual is placed on the use of the software.

This manual assumes all users are familiar with the following topics:

- Safety and basic operating procedures associated with robots, CNC machines, and all other equipment in the CIM environment.
- Basic operation of MS-Windows.

System administrators and advanced users should be familiar with the following topics:

- Robotic programming using the ACL language
- Robotic programming using Scorbase language
- Controlling and operating machines (e.g., CNCs)
- RS232 communications
- PC LAN administration, operation, and troubleshooting
- Setting up programmable logic controllers (PLCs)

Even if you will not be using the software in conjunction with an actual OpenCIM system, all users should read the background information provided in Chapter 1, Introduction, and Chapter 2, System Overview and the Safety guidelines in Chapter 3.

The installation instructions provided in Chapter 4, Installation, are intended for instructors and technical personnel who will be handling software and hardware installation.

Chapter 5, Project Manager, which activates the OpenCIM applications (CIM Manager and CIM setup), enables you to manage your own projects and provides an archive containing read-only projects managed by the CIM administrators.

Chapter 6, Operating CIM Manager and Chapter 7, CIM Manager Utility Programs, are organized to help all users begin using the OpenCIM system as quickly as possible. The material is presented in the order required to prepare and operate the OpenCIM system and the procedures guide you through the basic steps of software operation.

Chapter 8, Virtual CIM Setup, presents the Virtual CIM module, and teaches you how to set up the CIM by means of a graphic editor.

Chapter 9, OpenCIM Device Drivers, describes the operation of OpenCIM device drivers that are used in OpenCIM.

Chapter 10, Web Viewer, describes how you can remotely access a specific CIM Manager cell and track the production cycle.

Chapter 11, OpenCIM Programming, enables advanced users to do their own production experiments beyond the scope of the sample applications in order to explore new CIM techniques.

Chapter 12, Inside OpenCIM, contains details about the OpenCIM software, files and directory structure, and provides the information necessary for customizing the OpenCIM environment.

Chapter 13, Errors and Troubleshooting, provides detailed information on error handling and troubleshooting.

Chapter 14, Glossary, presents explanations of abbreviations and terminology used in the OpenCIM system.

Chapter 15, Intelitek Software Licensing, provides full details on the OpenCIM software licensing process.

2

System Overview

This chapter describes the hardware and software components which comprise an OpenCIM cell. It discusses each component individually and also how all components work together. It includes the following sections:

- **OpenCIM Description**, describes the OpenCIM unique features, and the additional OpenCIM software packages that are provided.

- **Production Operations**, describes the operations performed in the CIM cell when producing a product and provides an OpenCIM sample application.

- **Components of the OpenCIM Cell**, describes the basic elements (hardware and software) of the OpenCIM cell.

- **Stations**, describes the various stations (such as ASRS, Assembly, QC stations) including their functionality and provides a schematic example of the OpenCIM cell.

- **Material Flow in the OpenCIM Cell**, describes the basic flow of parts in a CIM Cell and provides descriptions of the various components involved in material flow. These may include templates, pallets, robots and so on.

- **CIM Control & Optimization**, describes how the OpenCIM cell is controlled and provides descriptions of the elements involved in the control of the CIM cell. These include the CIM Manager, the station manager, the graphic display and so on.

- **Device Drivers**, describes the various device drivers in OpenCIM. These include ACL device driver, CNC machine device driver, PLC device driver, Scorbase device drivers and so on.

- **OpenCIM Communication Network**, describes the communication networks that currently exist in OpenCIM. These include LAN, RS232 and I/O (inputs/outputs).

- **Integration**, provides an example scenario containing a step-by-step description of the various systems and devices that are associated with making a product in OpenCIM.

OpenCIM Description

This section describes the OpenCIM features as well as the OpenCIM software packages (such as, OpenFMS, and OpenCIM Offline that are provided in OpenCIM.

Unique Features

This section gives the background for understanding what is special about OpenCIM and basic operations performed in the OpenCIM system.

OpenCIM software provides unique industrial capabilities not found in other educational CIMs:

- OpenCIM "feels" familiar to first-time users because it is based on the standard Windows Graphic user interface.
- OpenCIM allows for targeted training at a given station or device.
- OpenCIM is realistic because it uses equipment found in actual industrial CIMs.
- OpenCIM resembles industrial CIMs in its ability to grow by using distributed processing at each production station. Distributed processing also makes for a more robust system. Even if the PC performing the central manager function goes down, each machine can still be operated in a stand-alone mode.
- OpenCIM uses a sophisticated network of PCs which allows various devices to perform multiple operations simultaneously. This network also allows CIM devices to communicate with each other.
- OpenCIM provides you with a powerful, yet flexible report generator. This utility program allows you to access nine types of predefined reports or gives you the option of creating your own user-defined reports.
- OpenCIM uses the latest object oriented techniques in:
- Defining the CIM Layout: Click on a Graphic object and drag it to the appropriate location on the CIM layout screen (e.g. Drag a robot in order to place it beside a CNC machine).
- Defining an Object's Properties: Click on an object to set its properties, e.g. the type of parts a machine can handle.
- Graphic Production Tracking: Uses Graphic objects to simulate CIM operation on screen.
- OpenCIM allows you to run a production simulator on a PC to observe results without actually operating the CIM production line.
- OpenCIM provides the opportunity to observe how a set of diverse hardware components work together in a real-world environment.
- OpenCIM is more comprehensive than other limited function CIMs. It can use a variety of equipment including:
- A variety of robots
- Processing machines
- Quality control devices (machine vision, laser scan meter, height gauge, CMM, caliper)
- Automated storage and retrieval systems (ASRS)
- Peripheral devices (barcode scanner, X-Y table, electric screwdriver, laser engraver, etc.)

- Custom devices by allowing you to easily set up your own device interfaces
- OpenCIM offers Graphic production tracking allowing you to observe each production operation on a central display.
- OpenCIM provides an open environment for advanced users who want to:
- Add their own devices
- Design their own products
- Interface their own software (e.g. MRP and cost analysis)
- Analyze CIM production data
- OpenCIM is a robust system that enables recovery from errors without the need to reset the entire CIM cell.

OpenCIM Additional Software Packages

The additional OpenCIM software packages, OpenFMS, OpenCIM Offline that are provided, are each described in detail in the following sections.

OpenFMS

OpenFMS is designed for use with flexible manufacturing systems. OpenFMS includes all the software modules and features of OpenCIM, and is intended to support systems with one robot tending one or two machines and quality control devices.

The following items are not included in the Virtual FMS Setup module, nor can they be configured for online operation.

Included in OpenFMS	Not included in OpenFMS
All types of robots	
Slidebases, linear conveyors, XY and linear positioning tables	Closed loop conveyor
ASRS-36 and all smaller storage device and part feeders	$ASRS^2$ and ASRS carousel
All CNC machines	Laser engraver
ViewFlex machine vision system, electronic calipers, laser scan meter	Coordinate measuring machine, electronic height gauge, barcode reader
Automatic gluing application; Automatic screw driving application	Hydraulic robot and pressing station; Pneumatic part feeding/sorting station; Process control station

OpenCIM Offline

OpenCIM Offline is the simulation version of OpenCIM. The user can design and run an unlimited variety of CIM or FMS cells in simulation mode.

It does not support hardware or online operation.

Device drivers are not included in this package.

Production Operations

The following operations are performed in the CIM cell when producing a product:

- Supplied parts (raw materials) are loaded into storage locations.
- Manufacturing orders are generated by the CIM Manager or by an external production scheduling package such as Fourth Shift or MAPICS.
- Parts are removed from the ASRS and transported on the conveyor to production stations.
- Robots take parts from the conveyor and move them to various production machines (e.g. CNC machines) at a station (machine tending).
- Typical production tasks include:
- Processing in a CNC machine
- Assembling two or more parts
- Quality control tests
- Robots return processed parts to the conveyor for transportation to the next station.
- Finished products are removed (unloaded) from the cell.

OpenCIM Sample Application - The Covered Box

The following Covered Box sample application is used in this manual to demonstrate the concepts of the OpenCIM system and can be found in the TUTORIAL_SAMPLE provided in the archive project list. The steps shown below are explained in more detail as each topic is introduced later in this manual.

The sample application produces a simple, covered box from a small, solid cube and a matching cover. Each component part is assumed to be in place on a separate template in the ASRS.
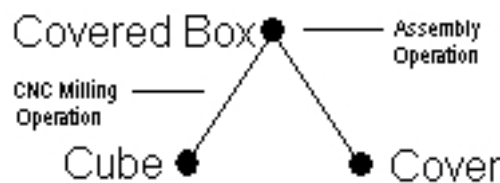
Figure 2-1: Part Definition Tree for Sample Application

The following steps detail the process of making a covered box:

1. The ASRS robot takes a solid cube and a cover from a storage cell and places them on separate pallets on the conveyor.
2. When the cover arrives at the assembly station, the assembly robot places it in a rack until the matching box arrives.
3. When the cube arrives at a CNC station, the CNC robot places the cube into a milling machine. The CNC machine reams out the center of the cube to form a box.
4. The CNC robot places the box on the conveyor.
5. When the box arrives at the assembly station, the robot places it on a rack.. When all the parts required for the assembly are on their rack, the robot places the base part (box) on the jig. The robot then retrieves the matching cover from the rack and places it on the box. The robot places the covered box on the conveyor.

6. When the covered box arrives at the ASRS, the robot places the finished product in a storage cell.

Components of the OpenCIM Cell

This section describes the elements of the OpenCIM cell. The topics covered include the physical configuration of the cell, material flow, control and production devices and communication networks. The emphasis is on the role each component plays in the integrated system, rather than on providing a detailed description of the component. Later chapters cover OpenCIM software in greater detail. Consult the appropriate user's manuals for details about each hardware component.

CIM cells are composed of the following basic elements:

Component	Description
Conveyor	Device that transports parts from station to station.
Production (Work) Stations	Locations around the cell where parts are processed and stored by machines and robots. Robots move parts between the conveyor and station machines.
CIM Manager	The PC that contains the CIM Manager software which coordinates the functioning of all devices in the cell using a LAN.
Station Manager	A PC that controls the different devices at a station and has a communication link with the CIM Manager. Device control is performed by OpenCIM device drivers that run on this PC. A device driver controls the operation of a device at the station in response to commands from the CIM Manager and other CIM elements.
Other Software Tools	OpenCIM Modules: Virtual CIM Setup, CIM Manager (with integrated Part Definition, Machine Definition, Storage Definition, MRP, Scheduler-Gantt, Reporter, Graphic Tracking modules), Project Manager, Performance, Optimization and Web Viewer.
	Third Party Software: Other production related software that interfaces to OpenCIM such as Production Scheduling, Manufacturing Resource Planning (MRP and MRP-II), Order Entry Systems, Data Base Management Systems (Xbase), etc.

Typically, a separate PC is dedicated to running or controlling each of the above elements. While two or more functions can be combined on a PC, the following discussion assumes the use of dedicated PCs.

Stations

The OpenCIM cell is composed of a set of stations located around a conveyor as shown schematically in the figure below:
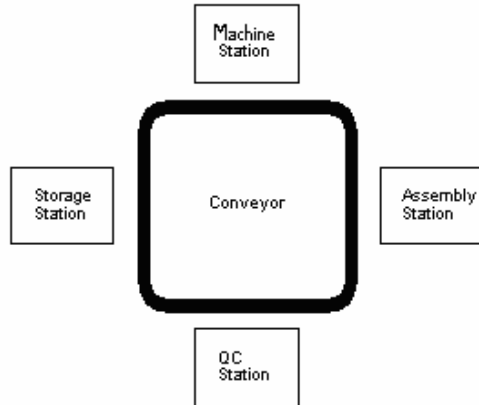
Figure 2-2: Schematic Example of an OpenCIM Cell

Each station is controlled by a Station Manager PC. A CIM Manager PC coordinates the activities of all stations. The number of stations may vary from cell to cell. A typical educational OpenCIM cell ranges from up to eight stations arranged around one conveyor to as few as a single station consisting of a robot tending a machine. The software can be adapted to more stations and conveyors.

Production commands are sent from the CIM Manager computer to the device drivers via the Station Manager PC. Status messages generated by devices are interpreted by the device driver and sent back to the CIM Manager.

Just as each industrial cell is an individual application of CIM technology, every OpenCIM cell has its own configuration. Generally, the stations that are usually available are described in the following table:

Station	Description
ASRS Station	Automated Storage and Retrieval System. Automatic warehouse which supplies raw materials to the OpenCIM cell, stores parts in intermediate stages of production, and holds finished products.
Machine Station	Station where materials are shaped, formed, or otherwise processed (e.g. using a CNC machine or laser engraver).
Assembly Station	A station where parts are put together. The resulting new part is called an assembly. Peripheral equipment and devices at an Assembly Station include an automatic screwdriver, Welder, X-Y table, part feeders, various robot grippers, etc.
QC Station	Quality Control. Inspection of parts using machine vision, laser scan meter, height gauge, continuity tester, CMM, caliper or other QC machines.

Various functions may be combined at one station, such as quality control and assembly.

Stations contain devices that perform production activities such as material processing or inspection. The following elements are generally present at a station:

Element	Description
Robot	A device which moves parts around a station (e.g. inserts parts into a CNC machine) and/or performs assembly operations.
Robot Controller	For example, an ACL controller which controls the robot and certain optional peripheral devices (e.g. X-Y table, barcode scanner).
Station Manager PC	A Station Manager PC where the device drivers are located that: • Translate OpenCIM production messages and commands to/from each station device (e.g. the ACL controller). • Provide a user interface for controlling station devices by manually sending OpenCIM commands (e.g. to CNC machines or an ACL controller). • Function as a terminal for devices that use an RS232 interface for setup and programming (such as the ACL controller).
Machine	A device that processes parts at a station. CNC machines such as lathes and mills process parts according to user-supplied G-code programs.
Robot Peripheral	A peripheral device which aids the robot in material handling tasks (e.g. a linear slidebase that supports a robot, an X-Y table, a tool adapter, various grippers such as pneumatic or suction models, etc.).

An example of an OpenCIM cell is shown schematically in the following figure:
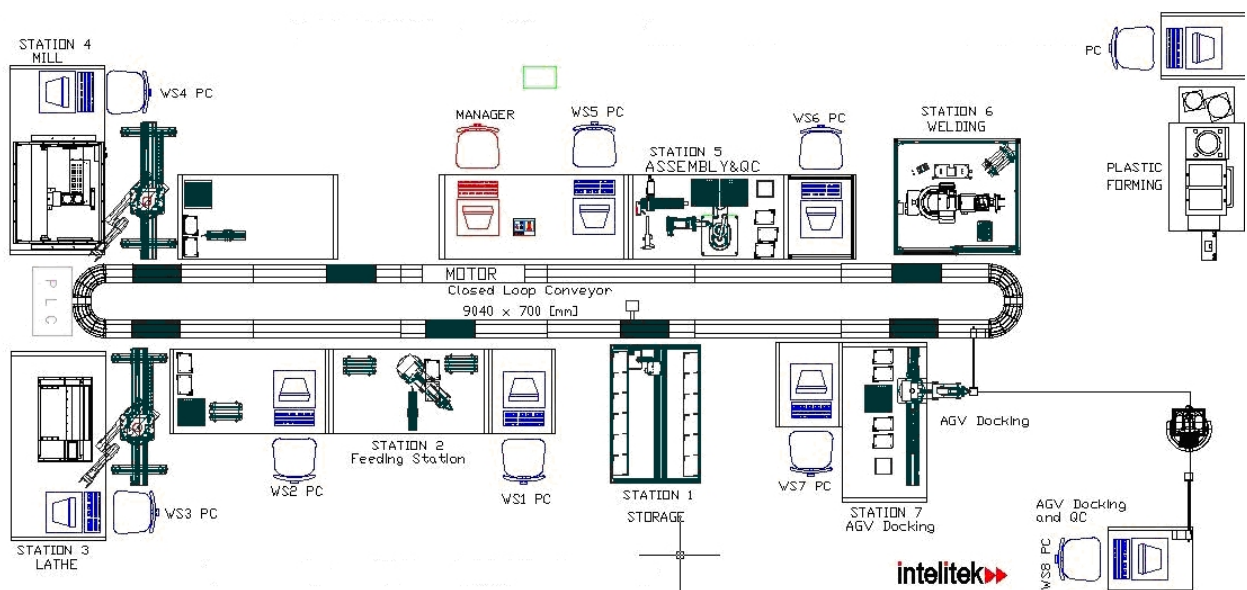
Figure 2-3: Sample OpenCIM Cell

Material Flow in the OpenCIM Cell

Material handling tasks can be divided into two groups:

- **Primary Material Handling**: These tasks perform the transportation of parts between stations.
- **Secondary Material Handling**: These tasks perform the handling of parts within a station, such as placing a template on the conveyor, removing a part from a feeder, inserting a part in a CNC machine, assembling parts and so on.

In an OpenCIM cell, the primary material handling tasks are usually performed by the conveyor. A robot (in combination with its peripherals) performs the secondary material handling tasks at each station.

When a robot removes a template from the conveyor, it typically places it on a buffer. (A buffer is a tray designed to hold a template when it is removed from the conveyor. The standard buffer is attached to the outer rim of the conveyor.) Once the template is on the buffer, the robot can remove a part from the template and take it to a station device.

The following scenario describes the basic flow of parts within the CIM cell:

- In response to production orders, the CIM Manager issues instructions to release parts from the ASRS and move them from station to station for processing.
- A robot at each station takes parts from the conveyor and places them in station machines.
- After a part has been processed at the station, the robot places the part back on the conveyor where it moves to the next station according to its production plan.

Templates

Templates are plastic trays which can hold various types of parts. They allow parts to be transported on the conveyor.
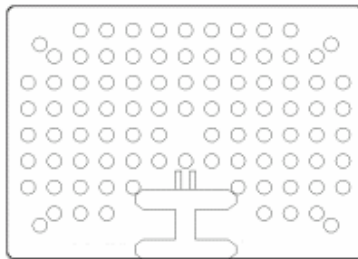
Figure 2-4: An Empty Template

A template contains a matrix of holes in which pins are placed to fit the dimensions of a part. Each arrangement of pins defines a unique template type. Each part may only be held by its assigned template. The handle, located on top of or in front of the template, facilitates grasping by a robot's gripper.

An optional barcode sticker on the side of the template shows the template's ID code. When barcodes are used, a barcode reader can verify the identity of each template inserted or removed from the ASRS.

Storage

An ASRS station is typically used as the main source of raw material for the cell. The ASRS can also serve as a warehouse for parts in various stages of production. Storage cells in the ASRS contain templates, either empty or loaded with parts. A CIM cell may contain any number of ASRS stations.

Part feeders can also be used to supply raw materials at various stations around the cell.

The following ASRS models exist in OpenCIM:

- **ASRS^2**: The ASRS^2 model is specifically designed to work in the OpenCIM environment. This unit contains a dedicated cartesian robot with an additional rotary axis that moves between two sets of storage racks. Each rack has a set of shelves divided into storage cells that are designed to hold part templates. The robot, which is controlled by a standard ACL Controller-B, moves templates between the conveyor and storage cells.
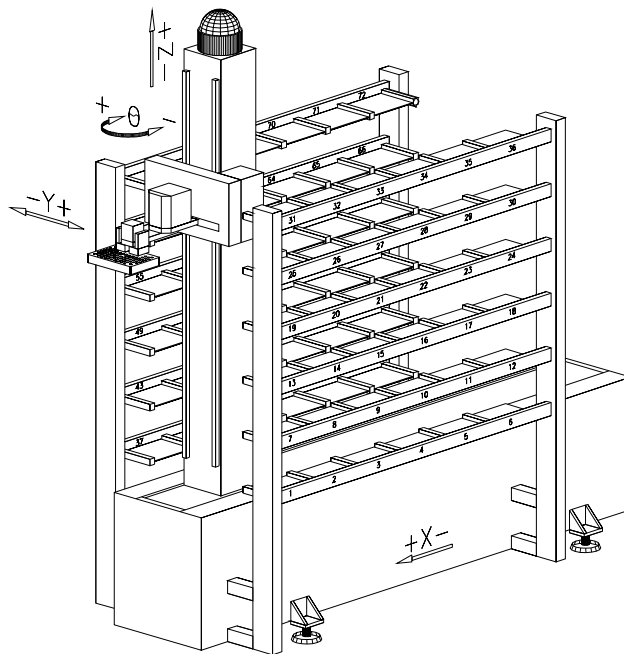
Figure 2-5: The ASRS^2 Robotic Storage Station

- **ASRS Carousel**: The ASRS carousel is a three-tier rotating warehouse which is tended by a robot, and controlled by an ACL controller.
- **ASRS-36**: The ASRS-36 is a cartesian robot with an additional rotary axis. It has a set of storage racks (divided into six levels with six cells each). The robot, which is controlled by a standard ACL Controller-A, moves the parts between the shelves and the conveyor.
- **ASRS-36u,ASRS-36uX2**: The ASRS36u and ASRS-36uX2 are cartesian robots with additional rotary axes. They each have a set of storage racks (divided into six levels with six cells each). The robots, which are controlled by USB controllers, move the parts between the shelves and the conveyor.
- **ASRS Rack**: The ASRS rack has a small number of cells, and is designed for use in a Micro-CIM work cell.

Conveyor and Pallets

A pallet is a tray which travels on the CIM conveyor and is designed to carry a template. To transport a part to another station, a robot places the template carrying the part on a pallet on the conveyor. The OpenCIM conveyor carries pallets in a continuous circuit from station to station. The conveyor is controlled by a PLC (programmable logic controller).

Each pallet has an ID number which is magnetically encoded in a bar on the pallet. In normal cell operation, each pallet is stopped briefly when it arrives at a station so that its magnetic code can be read. If the PLC determines that the pallet is needed at this station, it informs the CIM Manager. The pallet remains at this station until the CIM Manager sends a release command. While a pallet is stopped, the conveyor continues to transport other pallets which are moving between stations.

The location at which a pallet is stopped is called a conveyor station. Each OpenCIM station has its own conveyor station, which contains two pneumatically operated pallet stops, a magnetic pallet-arrival sensor, a magnetic pallet-in-place sensor and a set of magnetic pallet-code sensors.
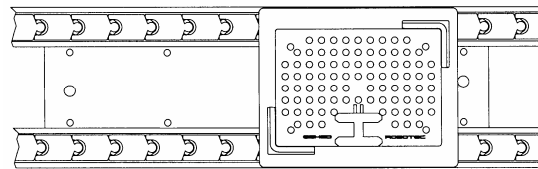
Figure 2-6: Pallet at Conveyor Station

Piston stops at each conveyor station can be raised to hold a pallet in place while the conveyor continues to cycle past the stations. The PLC controls the operation of these pneumatically driven piston stops, using input from pallet detection sensors located at the conveyor station.

The PLC keeps track of pallets which are empty and those which are carrying parts. It sends the destination station of each pallet to the PLC (default destination for each pallet is #99, allowing the pallet to continuously circle on the conveyor). Magnetic code readers at each station enable the PLC to identify the pallet ID numbers. Through a look-up-table the CIM Manager can instruct the PLC to do the following:

If ...	Then the PLC stops the pallet if ...
Pallet is empty	A template containing a part is ready to be picked up at this station.
Pallet carries an empty template	A part with no template needs to be picked up at this station.
Pallet carries a template with a part	This part needs to be delivered to this station.

If the part carried by the pallet does not require processing at the station, the pallet is allowed to continue on the conveyor.

Even though a pallet may be needed at a station, the CIM Manager may direct the PLC to release it if the robot that handles templates at this station is busy. Otherwise, a bottleneck could occur on the conveyor since other pallets would not be able to pass until the robot becomes available. The PLC would then stop the next appropriate pallet and try again.

If the robot is free it is instructed to remove the template containing the part from the pallet and place it on a station buffer. The empty pallet is then released and can continue on the conveyor, ready to pick up another template.

Conveyor Lights

Red and green lights at each conveyor station indicate the following:

Option	Description
Green On	Station is idle waiting for a pallet to arrive.
Red On	A pallet has been stopped for use at this station.
Flashing Red	An error has been reported at this station by an OpenCIM device driver (e.g. robot impact, Emergency button pressed).
Flashing Red at All Stations	All stations have stopped because someone has pressed the Emergency Stop Button.

Robots and Controllers

CIM robots move parts within a station (secondary material handling) and perform assembly operations. Robots vary in speed, payload, accuracy, range of movements (degrees of freedom), working envelope (horizontally or vertically articulated), and drive mechanism (DC servo, AC servo or pneumatic).

The following Intelitek robots may be integrated in an OpenCIM environment:

Robots controlled by Scorbase:

- SCORBOT-ER 4u
- ASRS-36u
- ASRS-36uX2
- SCORBOT-ER 9Pro
- SCORA-ER 14Pro
- HP3 with NXC100 and XtraDrive controllers

Robots controlled by ACL:

- SCORBOT-ER 5, SCORBOT-ER 5 Plus
- SCORBOT-ER 7
- SCORBOT-ER 9
- SCORA-ER 14
- PERFORMER-MK3
- Performer-SV3 with BRC controller
- MK2
- Square ASRS
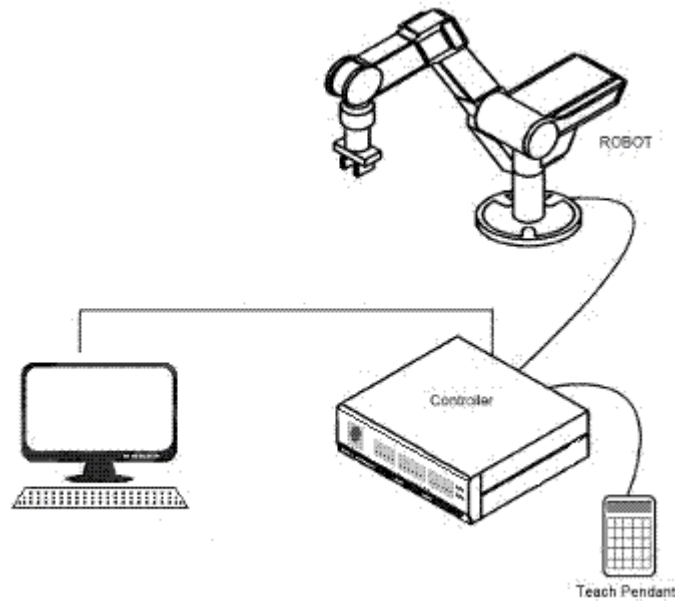
Figure 2-7: Robot, Controller, Teach Pendant, and Station Manager PC

Intelitek robots may be operated using either ACL or Scorbase as their programming language. Each programming language uses a device driver in order to communicate with the OpenCIM manager software.

Robotic programs inform the robot what path to follow and what task to perform once it reaches a destination. The controller (ACL, USB or USB PRO) provides the power supply to the robot and moves the robot by controlling the power to the motors inside the robot.

The controller can control robots and peripheral devices (e.g. barcode reader, X-Y table) simultaneously. The CIM peripheral devices are connected to the controller's auxiliary I/O ports and RS232 ports. All ports (robot, I/O, RS232) can be controlled using Robotic programs.

Processing Machines

Processing machines process parts according to processing programs stored in their memory. The CIM Manager keeps track of the programs that reside in a machine's memory and downloads a new program to process an upcoming part as needed.

OpenCIM can interface to machines that use either I/O lines or an RS232 interface to control operations such as opening/closing a door, turning on/off the machine, etc. Status lines report information, such as whether a door is open or closed and when a process is finished.

CIM Control & Optimization

To understand how the OpenCIM cell is controlled, it is necessary to look at its control elements, the communication channels that each element uses to control the devices, and the network that links the various control elements as an integrated whole.

The OpenCIM system consists of various software modules which perform command, control, and monitoring functions:

- Virtual CIM Setup Software
- CIM Manager Software with integrated modules (Storage Definition, Machine Definition, Part Definition, MRP, etc.)
- Device Managers (device drivers located on Station PC)
- Conveyor Manager (PLC device driver located on a Station PC)
- Graphic Tracking (integrated in CIM Manager and/or as external module)

OpenCIM uses a distributed control strategy as follows:

- Each Station Manager PC runs a set of device drivers that control the devices at its station.
- A PLC controls the operation of the conveyor.
- The CIM Manager provides the highest level of control and integrates the activities of the entire cell. It sends command messages to the various device drivers via the network. These units attempt to execute the command and respond with status messages describing the results.

What is Optimization?

In multiple productions (when producing more than one part in a manufacturing cycle) it is required to optimize the timing synchronization to maximize system functionality at minimum costs. This process is called System Optimization.

There are two types of system optimization methods. In the first method, the schedule is defined in advance and a detailed schedule is set for each machine. In the second method, a set of rules is defined, according to which the system acts, and during run-time, decisions are made according to this set of rules. The Optimization Approach used by OpenCIM implements this second method which enables the system to overcome failures, imprecision and inaccurate assumptions, and still provide you with an efficient system.

In a CIM system, the Time, Cost and Quality factors determine the efficiency of the overall system performance. These factors are the target functions of the different algorithms (or of the system). Generally, combinations of these three factors are used in a CIM manufacturing cycle. Additional factors, such as the machine type, can also affect the performance of the CIM cell.

The CIM Manager controls the efficiency of a manufacturing cycle, by performing the following tasks:

- **Scheduling:** The CIM Manager determines the time schedule that a part will be released from storage and in what order.
- **Dispatching:** The CIM Manager determines which part will be processed in which machine and when.

These Scheduling and Dispatching tasks are controlled by the CIM Optimization Definition, as described in Optimization in Chapter 7, CIM Manager Utility Programs.

CIM Definition Modules

The CIM Manager maintains the OpenCIM database which contains information on the physical and communication configuration of the cell, inventory of raw materials and parts, manufacturing processes, part definitions, and orders. Interactive software lets you define:

- The layout of machines and stations in the CIM cell.
- Machines and the production processes they can perform
- The bill of materials used to produce each part.
- An order which specifies the parts you want to produce.
- The contents of all storage locations.

The CIM Definition modules also allow you to back up and restore the above information.

The CIM Definition modules are usually run on the same PC used for the CIM Manager. Each of these modules creates data files in a DBF format. These files can be viewed and edited with a dBASE editor or with the DataBase Tool which can be activated from the Manager software (for details refer to Chapter 6: Operating CIM Manager). They can also be modified by a user-supplied application (e.g. using an MRP program to create an order file).

The CIM Manager

The CIM Manager performs the following basic functions:

- **Evaluates the Production Plan**: Fills in details in the production plan on how to produce the parts submitted in an order.
- **Executes the Production Plan:** Controls and monitors the CIM equipment to produce the parts as specified in the production plan.

The CIM Manager program provides centralized control of on-line production activities. It sends commands to station devices and receives responses which enable it to track the flow of parts during production.

After the production plan has been prepared, you can issue commands to start and stop production from the CIM Manager. When you start production, the CIM Manager begins sending commands over the LAN to Station Manager PCs in order to:

- Direct the flow of parts between stations on the conveyor.
- Gather at a station (e.g. in a storage rack) the parts that are needed in order to perform an assembly operation.
- Synchronize processes which can be performed concurrently and those which must be performed consecutively.

The CIM Manager runs a virtual machine that corresponds to each physical machine in the CIM. This virtual machine keeps track of the status and parts queue at the physical machine. The CIM Manager uses this information to decide when to send routing messages to bring parts to the machine.

The Station Manager

A Station Manager PC can be connected to a variety of robots and machines. It runs a separate device driver in order to communicate with the controller for each device connected to this PC. These device drivers run simultaneously in individual windows using the multitasking capabilities of MS-Windows. A PC running a set of OpenCIM device drivers is said to be acting as a station manager. These OpenCIM device drivers perform the following functions:

- Translate OpenCIM commands into instructions understood by station devices.
- Translate status information from a device into OpenCIM messages and relay these messages to the appropriate OpenCIM entities.
- Allow the user to interactively control devices such as CNC machines, robots, and other station devices.
- Download G-code programs to a CNC machine.

If desired, the station computer can be used to operate the station as a stand-alone manufacturing cell. Each device driver on a station PC has a control panel which provides this capability.

PC Requirements in OpenCIM

The OpenCIM system provides great flexibility in the way in which PCs are used around the cell. You can control the degree of distributed processing in the OpenCIM environment based on how you assign the following software modules to PCs:

- CIM Manager
- Device Manager (i.e. OpenCIM device drivers)
- Graphic Tracking

In a busy CIM cell, performance is enhanced by installing most of the above software modules on a separate PC. In this scenario, each station would have a dedicated Station Manager PC. A LAN is used to connect all of the PCs that are running OpenCIM software. The OpenCIM software modules use this LAN to exchange information.

At the other extreme, all of the above OpenCIM software modules could be loaded on one high-performance PC. Multiple OpenCIM software modules can run on the same PC in the multi-tasking environment provided by Windows. In this case, inter-module communication is internal, where services are provided by the operating system. When a single PC is used, no LAN is required, although Network Settings for the PC still have to be configured for TCP/IP.

Intermediate configurations are also possible. For example, one PC may be used to run both the CIM Manager module as well as the PLC device driver. A single Station Manager PC may be connected to devices at different stations.
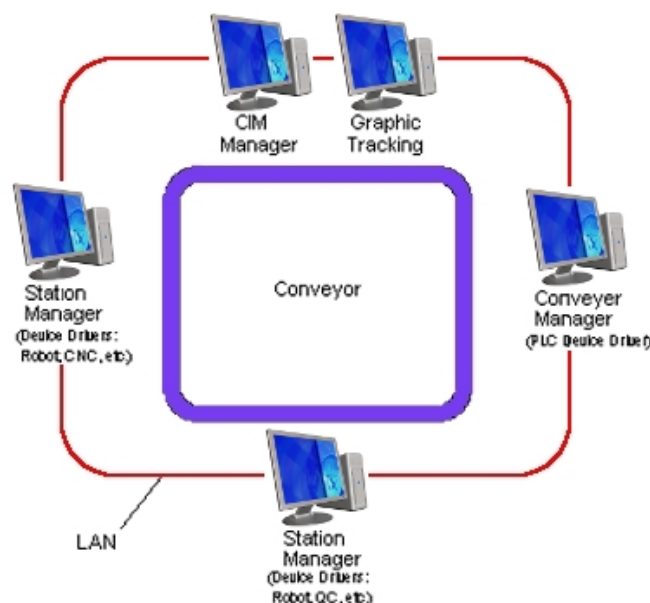
Figure 2-8: OpenCIM Modules Distributed Among PC's

Figure 2-9: OpenCIM Modules Concentrated in One PC

Recommended PC requirements for OpenCIM software are:
- Pentium 4 Dual Core with 3 GHz processor or higher
- At least 512 MB RAM (2 GB for Windows Vista/7)
- CD ROM drive
- Windows XP/Vista/7 (32/64 bit)
- Hard drive with at least 1 GB of free disk space
- Super VGA or better graphics display, minimum 256 colors
- Fully Windows compatible LAN interface card
- USB port
- Separate RS232 ports on the PC-for each hardware device which uses an RS232 port
- Mouse or other pointing device

Minimum PC requirements are:
- Pentium 4 Dual Core with 2 GHz processor
- 512 MB RAM
- CD ROM drive
- Windows XP Professional (32/64 bit)
- Hard drive with at least 1 GB of free disk space
- Super VGA or better graphics display, minimum 256 colors
- Fully Windows compatible LAN interface card
- USB port
- Separate RS232 ports on the PC for each hardware device which uses an RS232 port

Note

Supported Windows version (32/64 bit):
- Windows XP Professional
- Windows Vista Home Premium/Business/Ultimate
- Windows 7 Home Premium/Enterprise/Professional/Ultimate

Note   *If a PC is used for multiple functions, the PC must be powerful enough to keep up with the flow of information in the OpenCIM real-time environment. For example, a very high performance PC would be required to simultaneously run the CIM Manager program, control the conveyor, function as a Station Manager, and show the Graphic production display in real-time.*

Device Drivers

Each device at a station is controlled by an OpenCIM device driver program running on the Station Manager PC. A device driver translates OpenCIM messages in two directions:

- OpenCIM instruction messages into a set of commands understood by the target device.
- A response from the device into an OpenCIM status message.

After a device driver translates an instruction into a command, it sends the command to the destination machine or robot.

OpenCIM instructions can come from:

- The CIM Manager
- Other OpenCIM device drivers
- The device driver's user interface
- User application programs

When a device returns a response, the device driver translates this information into a standard OpenCIM message format. It then relays this information as follows:

- Device status information to the CIM Manager.
- Real-time production data to the Graphic Tracking module.
- Designated messages from a device to a user defined process that is monitoring this device.
- Specific messages to other device drivers.

A separate copy of a device driver is run on a Station Manager PC for each device at the station. Each device driver presents a control panel which allows you to:

- Observe the command and response messages on-screen as they are sent to and from a device.
- Issue commands interactively to a device and observe its responses on-screen.
- Capture all commands and responses in a log file if you want to analyze the behavior of a device.

Parameters which control the operation of each device driver are found in the device driver's configuration files (ACLVD1.INI, CNCVD1.INI). You can view and change these parameters by editing this file with a text editor.

Robotic Device Drivers

The robotic device driver communicates with a robotic controller, which controls robots (including the *ASRS*). This robotic device driver receives command messages from the CIM Manager to perform robotic operations. The robotic device driver then translates these requests into commands to run the corresponding robotic programs residing in the robotic controller. When the controller has finished moving the robot, it sends a confirmation message back to the device driver which forwards it to the CIM Manager.

The following robotic device drivers are used in CIM:

- **ACL Device Driver:** The ACL device driver uses an RS232 port on the Station Manager PC to communicate with the ACL controller.
- **Scorbase for Controller USB Device Driver:** The Scorbase software operates the Controller USB, using the Scorbase device driver (residing in the software).
- **Scorbase for Controller USB-Pro Device Driver:** The Scorbase for Controller USB-Pro software operates the Controller USB-Pro, using the Scorbase device driver (residing in the software).
- **Scorbase for NXC 100 and XtraDrive Controllers Device Driver**: The Scorbase for NXC100 and XtraDrive Controllers software operates the NXC100 and XtraDrive Controllers, using the Scorbase for NXC100 and XtraDrive Controllers device driver (residing in the software).
- **Scorbase for FS100 and XtraDrive Controllers Device Driver**: The Scorbase for FS100 and XtraDrive Controllers software operates the FS100 and XtraDrive Controllers, using the Scorbase for FS100 and XtraDrive Controllers device driver (residing in the software).

The primary operation performed by a robot using pick-and-place. Pick-and-place means, taking a part from one location (source) and placing it at another location (target). For example, a common pick-and-place operation involves taking a part from a template and placing it in a CNC machine. The coordinates for each pick-and-place operation are defined in advance and assigned a Location ID. The CIM Manager sends a pick-and-place command to the controller which includes two Location IDs (source and target).

The actual path a robot follows when moving from a source location to a target is defined in a robotic program residing in the robotic controller. The CIM is not involved with the complexities of robot movement. It only sends pick-and-place commands which specify the action to take, not how to take it.

In specific CIM station configurations, the robotic controller also controls the various devices (such as, barcode readers and CNC machines). In some cases, the communication between the CIM Manager and these devices is handled via the robotic device drivers by activating a robotic program that is responsible for communicating with the device. This is performed according to the ACL or USB configuration, as follows:

- **ACL Configuration:** The RS232 from the peripheral device (such as, CNC or barcode reader), is connected to the ACL controller and is controlled by an ACL program. The ACL device driver then activates this program to operate the device. For example, write specific code into the RS232 line that is connected to the device.
- **Scorbase for Controller USB:** The RS232 from the peripheral device (such as, CNC or barcode reader), is connected to the PC COM port and is controlled by the Scorbase program. The Scorbase device driver then activates the appropriate sub-routine to activate the device. For example, write specific code into the RS232 line that is connected to the device.

- **Scorbase for Controller USB PRO:** The RS232 from the peripheral device (such as CNC or barcode reader), is connected to the PC COM port and is controlled by the Scorbase for USB-PRO controller program. The Scorbase for USB-PRO controller device driver then activates the appropriate sub-routine to activate the device. For example, write specific code into the RS232 line that is connected to the device.
- **Scorbase for NXC100:** The RS232 from the peripheral device (such as CNC or barcode reader), is connected to the PC COM port and is controlled by the Scorbase for NXC 100 and XtraDrive controller program. The Scorbase for NXC100 and XtraDrive controllers device driver then activates the appropriate sub-routine to activate the device. For example, write specific code into the RS232 line that is connected to the device.
- **Scorbase for FS100:** The RS232/TCP/IP from the peripheral device is connected to the PC (COM port or LAN port) and is controlled by the Scorbase for FS100 and XtraDrive controller program. The Scorbase for FS100 and XtraDrive controllers device driver then activates the appropriate sub-routine to activate the device. For example, send/receive specific commands through the RS232/LAN line that is connected to the device.

CNC Machine Device Driver

OpenCIM uses a CNC device driver to interface with any CNC machine that uses I/O lines or an RS232 interface to receive commands and report machine status. This device driver can be adapted to work with any such CNC machine using a built-in language to write short interface routines. The CNC device driver can control a machine, read the status of a machine, send status messages to other CIM entities, and download G-code programs to the machine using an RS232 interface.

The normal sequence of events is:

- The CIM Manager instructs the robot to load a part into the CNC machine.
- The CNC device driver receives a command to process a part.
- The device driver activates the appropriate output line to turn on the machine.
- The device driver waits for the operation to complete by monitoring a status line.
- The device driver sends a status message back to the CIM Manager.
- The CIM Manager instructs the robot to remove the part in the CNC machine.

The following sample scenario demonstrates the role of the CNC device driver:

1. The CIM Manager sends a command to the CNC device driver to download a G-code file needed to machine an upcoming part.

2. When a robot is ready to insert a part into the CNC machine, its Scorbase or ACL program sends commands to the CNC machine to:
   - Open the door of the CNC machine
   - Open the chuck/vice of the CNC machine

3. The robot inserts the part into the chuck/vice. The Scorbase or ACL program sends commands to the CNC machine to:
   - Close the chuck/vice of the CNC machine
   - Close the door of the CNC machine

4. The CIM Manager waits for status lines to indicate that the part is in place ready to be machined.

5. The CIM Manager sends a signal to the CNC machine (via the CNC device driver) to begin machining the part.

6. The CNC device driver waits for a status line to indicate that the machining operation is complete.

7. The CNC device driver sends an "Operation Complete" status message to the CIM Manager. The Manager in turn sends a command to the ACL controller to signal the robot that it can now remove the part from the CNC machine.

8. The Scorbase or ACL program sends commands to the CNC device driver to:
   - Open the chuck
   - Open the door

9. The Scorbase or ACL program directs the robot to remove the part. It then sends a status message to the CIM Manager signaling that the unloading is finished.

The CNC device driver can communicate with a machine using either an RS232 interface or a special I/O board in the Station Manager PC. For an I/O interface, each status line and command line for a machine is connected to this board.

In most cases the CNC device driver uses the ACL device driver or Scorbase device driver to communicate with the machine on RS232 and I/O level.

Since CNC machines from different manufacturers have different sets of status lines and command lines, the CNC device driver uses a flexible control language called the CNC Language Interpreter (CLINT) for interacting with the machine. This language allows you to adapt the device driver to the features and wiring configuration of a specific machine by writing a set of customized routines. When a part arrives at a CNC machine, the CNC device driver receives a command to run the corresponding CLINT routine to process this part.

PLC Device Driver

The PLC device driver communicates with the programmable logic controller which directs the operation of the conveyor. This device driver receives messages from the CIM Manager about the contents and destination of the pallets traveling on the conveyor. The PLC device driver translates these messages into commands understood by the PLC.

When the PLC detects a pallet arriving or leaving a station, it sends a status message to the PLC device driver on a station PC. This device driver in turn translates the message into a standard OpenCIM format. It then broadcasts the message to the CIM Manager, the Graphic Tracking module, and any user applications which have registered for this type of message.

The control panel of the PLC device driver lists the destination of each pallet and can show which pallet is at each station. It also lets you interactively issue commands to stop a pallet at a station.

The PLC is normally connected to one of the Station Manager PCs (using an RS232 connection). This PC communicates with the CIM Manager PC using the LAN.

OpenCIM can accommodate any type of PLC. If a PLC cannot support the pallet look-up table in its memory, this information is stored on the PLC Manager PC. However, better performance results when the look-up table is stored in the PLC. This arrangement eliminates the serial communication overhead associated with performing frequent look-ups every time a pallet passes a station.

Quality Control Device Drivers

OpenCIM uses a set of device drivers to communicate with different types of quality control devices such as:

- Vision Systems (Viewflex, ROBOTVISIONpro)
- Coordinate Measuring Machine (CMM)
- Laser scan meters, calipers
- Barcode Readers (BCR)
- RFID Readers (RFIDR)

These device drivers receive instructions from the CIM Manager to perform a predefined quality control check on a part. These instructions specify the type of test to perform and the range of acceptable results. The quality control tests for each part are defined according to the instructions for each quality control device.

A quality control device driver translates OpenCIM messages into commands understood by its associated quality control device. The device driver can communicate with quality control devices attached to either a Station Manager PC or ACL controller via an RS232 interface or I/O port. When the quality control device performs a test, it sends the result back to the quality control device driver on the Station Manager PC. The device driver translates the message into a standard OpenCIM format. It then sends this status message to the CIM Manager.

The control panel of each quality control device driver lets you observe the results of each quality control test. It also allows you to interactively issue commands to a quality control device or send status messages to the CIM Manager.

Hydraulic Workstation Device Drivers

The hydraulic workstation device driver communicates with the hydraulic workstation via Advantech's I/O PCI card. This device driver receives command messages from the CIM Manager to perform robot operations and activates the press. The hydraulic device driver translates these requests into commands to run the corresponding programs residing in the HYD-2800 program on the PC. When the device driver has finished moving the robot, it sends a confirmation message back to the CIM Manager.

The hydraulic workstation device driver can also activate user-supplied programs that can be prepared using the HYD-2800 program. These programs are commonly used to control the hydraulic robot and the press.

Pneumatic WorkStation Device Drivers

The pneumatic device driver runs on a PC that controls the pneumatic station via the Vuniq software. The pneumatic station is used to supply three types of raw materials. These include two types of cylindrical parts as well as block parts.

This device driver receives command messages from the CIM Manager to supply raw material parts to the CIM system. The pneumatic device driver translates these requests into commands for the manipulator control to perform the following tasks:

- Lift the template off the conveyer and place it on the pneumatic station.
- Take the required part from the feeder and place it on the template.
- Lift the template and part off the pneumatic station and place it back on the conveyer.

When the device driver has finished supplying the part, it sends a confirmation message back to the CIM Manager.

Process Workstation Device Drivers

The process device driver runs on a PC that controls the process station via the Vuniq software. The process station is used to perform a specified chemical process on raw material parts. This device driver receives command messages from the CIM Manager to process the raw material part. The process device driver translates these requests into commands for the manipulator and process control to perform the following tasks:

- Lift the template and raw material off the conveyer and place it on the buffer.
- Lift the raw material off the template and perfom the process, specified in the Vuniq software. At the end of this process the processed part is placed back on the template.
- Lift the template and the processed part off the buffer and place it back on the conveyer.

When the device driver has completed these tasks, it sends a confirmation message back to the CIM Manager.

OpenCIM Communication Network

This section describes in detail each of the following communication networks: I/O, RS232, and LAN.
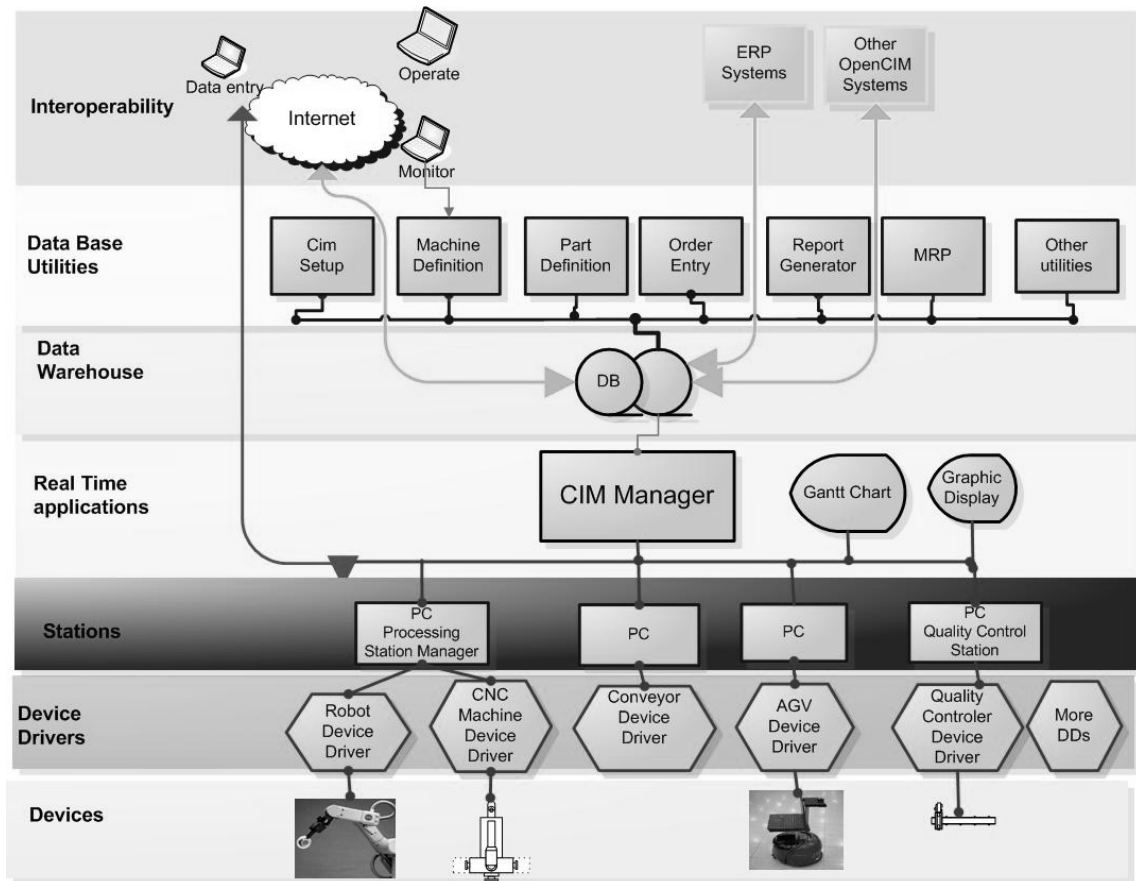
Figure 2-10: Communication Networks Used in OpenCIM

LAN

The CIM Manager and device drivers exchange command and status messages via the OpenCIM Network. This network is based on the Windows TCP/IP communication protocol. Each module (manager, device drivers) in the TCP/IP protocol has two communication sockets, the server and the client. A socket represents an endpoint for communication between processes across a network. Both the server and the client have an IP address and a port number that are unique. The OpenCIM Network transparently delivers the message to the destination application whether it is running on the same PC or on a PC connected via a LAN.

OpenCIM uses a LAN to exchange information between software modules running on separate computers. When the following software modules are configured to run on separate PCs, the LAN allows them to exchange commands and status information in real-time:

- The CIM Manager software
- The Device Driver software
- The Graphic Tracking PC
- Any PCs running user supplied applications that are interfaced to OpenCIM

OpenCIM uses the LAN to:

- Send commands from the CIM Manager to Device Drivers (e.g. data such as part ID #, task to perform, machine to use, etc.)
- Send real-time production status messages from Device Drivers to the CIM Manager.
- Allow Device Drivers to retrieve control programs (e.g. G-code) stored on the server.
- Send real-time production status messages to the Graphic Tracking software.
- Transfer CIM messages between different device drivers.
- Transfer CIM messages between devices and a user application running on a networked PC.
- Perform central backup and restore of all PCs attached to the LAN.

OpenCIM can use any LAN using a TCP/IP protocol which is supported by Windows to transfer files and send real-time messages.

RS232

An RS232 interface (also known as a serial port or com port on a PC) is a low-speed data communications port that typically transmits and receives information at the rate of 300-19,200 bits per second (bps). Data is transmitted serially, i.e. one bit at a time. There is a separate line for transmitting and receiving data.

The following OpenCIM devices use RS232 connections:

Station Manager PCs	Station Manager PCs use RS232 to: • Download programs to processing machines • Pass OpenCIM messages to/from an ACL controller • Provide a terminal interface for programming ACL controllers • Pass OpenCIM messages to/from other station devices such as QC systems.
PLC	The PLC Manager PC uses RS232 to: • Send pallet destination information to the PLC • Send commands to the PLC • Receive status messages from the PLC
Peripheral Devices	Peripheral devices can be attached to the RS232 ports of an ACL controller (e.g. barcode reader).

Inputs/Outputs

I/O connections can be used to turn production devices on and off, and to transmit binary information about the status of a device. A separate wire carries each I/O signal. I/O connections use a low voltage DC signal. The exact voltage depends on the specifications of the devices being connected.

I/O connections are used for signaling purposes only. An output signal should never be used to directly drive a piece of electrical equipment. In this case, an output signal should be buffered through a relay or other device to prevent overloading the circuit.

The following CIM devices use I/O connections:

- PLC (to raise and lower piston stops)
- Magnetic sensors at conveyor stations used to send pallet IDs to the PLC
- Processing machines (to operate the machine and report its status)
- Devices attached to an ACL controller's I/O ports (e.g. an automatic screwdriver, a pneumatic gripper for a robot, etc.)

Integration

In the previous sections you were given an overview of the entire cell. This section describes the integration of various systems and devices by considering the sequence of events when a part moves from station to station for processing.

When the user activates a production order, the CIM Manager builds a production plan. This plan includes the parts to be processed, the stations where they are to be processed, and the production activities they will undergo.

In the sample scenario presented below, a cube moves from storage in the ASRS to a CNC station where it is machined into a box. The table below provides a step-by-step description of the flow of information throughout the CIM associated with making a box. Note that operations which are listed in the same grid take place concurrently.

Message Source	Message Destination	Message Content (Command messages in normal typeface) *(Status messages in italics)*
ASRS Station		
CIM Manager	⇒PLC	Stop the next empty pallet that arrives at the ASRS station.
PLC	⇒CIM Manager	*Empty pallet has arrived at the ASRS station.*
CIM Manager	⇒Robot Controller	Use robot to remove a template with a cube from storage and place it on the pallet..
Robot Controller	⇒CIM Manager	*Template is in place on pallet.*
CIM Manager	⇒PLC	Release this pallet from the ASRS station. Stop this pallet when it arrives at the CNC station.
PLC	⇒CIM Manager	*Pallet with cube has arrived at the CNC station.*
CNC Station		
CIM Manager	⇒Robot Controller	Use robot to remove template from pallet and place it on buffer of CNC Station.
Robot Controller	⇒CIM Manager	*Template with part is waiting on buffer of CNC station.*
CIM Manager	⇒PLC	Release pallet from CNC station.
CIM Manager	⇒Robot Controller	Use robot to place part in CNC machine.
Robot Controller	⇒CIM Manager	*Part is in CNC machine.*
CIM Manager	⇒CNC Machine	Use the CNC machine to ream a hole in the cube to form a box.
CNC Machine	⇒CIM Manager	*Process complete. Box ready.*
CIM Manager	⇒Robot Controller	Use robot to place box on template in buffer.
CIM Manager	⇒PLC	Stop next empty pallet at CNC station.
Robot Controller	⇒CIM Manager	*Box is in place on template.*
PLC	⇒CIM Manager	*Empty pallet has arrived at CNC station.*
CIM Manager	⇒Robot Controller	Use robot to place template with box on pallet.

Message Source	Message Destination	Message Content (Command messages in normal typeface) *(Status messages in italics)*
Robot Controller	⇒CIM Manager	Template is in place on pallet.
CIM Manager	⇒PLC	Release pallet from the CNC station. Stop this pallet when it arrives at the Assembly Station ...

3

Safety

The OpenCIM cell is a complex system containing many different machines, each potentially dangerous if proper safety practices are not followed. Certain safe operating practices apply to all, while others are device specific. This chapter presents the general rules, followed by a brief discussion of the safety requirements of each component and contains the following sections:

- **General Safety Rules**, describes the general safety rules that must be followed in the OpenCIM cell surroundings area.
- **Robot and Controller Safety**, describes the safety rules that must be followed when using the OpenCIM robots and controllers.
- **CNC Machine Safety**, describes the safety instructions that must be followed when using the OpenCIM CNC machines.
- **ASRS Safety**, describes the safety instructions that must be followed when using the OpenCIM ASRS storage systems.
- **Conveyor and PLC Safety**, describes the conveyer and PLC safety instructions.

In addition, the User Manuals for all robots and CNC machines contain full descriptions of the safety procedures and warnings for these devices. The user is strongly urged to read these manuals before working with the devices.

Warning!
Do not approach any OpenCIM equipment before reading the safety guidelines in this chapter.

General Safety Rules

The following rules should be followed when in the vicinity of all moving machinery in the CIM cell such as robots, ASRS, or conveyor.

Warning!

Exercise caution whenever you are in the area of the CIM cell.

- Be sure you know the location of the ON/OFF switch, and any emergency shutoff switches, on the following equipment:
  - Robot controller
  - Pallet conveyor
  - CNC machines
- Be alert since any idle piece of equipment could start up suddenly. All CIM machinery can be turned on and off unexpectedly via computer control.
- Exercise special caution in the vicinity of robots since they can start up without notice and move in unexpected ways.
- Members of a group should be careful not to crowd too close around moving equipment when observing the activities at a given station.
- Do not come near a moving device when it is in operation. Be careful that hair, clothes (especially loose sleeves) and jewelry are kept away from the mechanism.
- Do not stick your fingers into a device while it is in operation; they may get caught in the mechanism.
- Keep the work area clean and free of clutter.
- Do not exceed the loading capacity of a device.
- Turn off a device before attempting adjustments, performing maintenance or measuring a part.

Robot and Controller Safety

Extreme caution must be exercised in the use of OpenCIM robots. Recklessness may cause physical harm to the operator and other people in the vicinity.

- Set up a protective screen or guardrail around the robot.
- Make sure the robot base is properly bolted to a table or pedestal. Otherwise, the robot may become unstable and topple during operation.
- Do not use physical force on the robot arm to change its position, or for any other reason.
- Be sure the robot arm has sufficient space in which to operate freely, especially during homing.
- Before connecting any input or output to the controller, and before approaching or handling the robot, make that the controller's main power switch is turned off.
- Before opening the controller housing, be sure to unplug the controller power cable from the AC power outlet. It is not sufficient to switch off the power; the power supplies inside the controller still contain dangerously high voltages.
- Before removing any fuses, be sure to turn off the controller and unplug the controller power cable from the AC power outlet.

Note

To immediately abort all running programs and stop movement of all axes (e.g. robots):

- *Press the Abort or Emergency Stop key on the teach pendant (TP), or*
- *Press the controller's red Emergency button.*
- *In ACL controller: Use the ACL command A <Enter>, or*

  in Scorbase, press the STOP icon on the toolbar.

CNC Machine Safety

The following list contains general safety instructions for the use of CNC machines. Be sure you adhere to the safety rules for the specific machines included in your cell.

- Always wear safety goggles when near the machine. Be aware that some materials, such as the brass bars, spray chips while being processed. Make sure all persons near the machine are protected.
- Keep children and visitors away. Set up the machine so that children or visitors unfamiliar with the machine cannot start it. Protect the machine against unintentional use by removing the switch key.
- Chuck parts and tools firmly and safely.
- Perform measuring and chucking work only when the machine is at a standstill.
- Remove adjusting key and wrenches even when the machine is not being used. Never attach chuck keys to a machine with a chain or similar connector.
- Always work with sharp tools.

ASRS Safety

This section describes the safety instructions that must be followed when using the OpenCIM ASRS storage systems (ASRS Carousel, ASRS^2, ASRS-36, ASRS-36u, ASRS-36uX2), each of which is described in the sections that follow.

ASRS Carousel

The following list contains safety instructions for the use of ASRS Carousel:

- Do not place your hand or any other object in or near the main drive belt, located under the lowest level of the ASRS carousel, while the device is operating. The belt is extremely dangerous and can cause severe injury.
- Make sure that the carousel has been disconnected from the AC power supply before approaching the motor and belts.

ASRS^2 and ASRS-36

The following list contains safety instructions for the use of ASRS^2 and ASRS-36

- Do not enter the robot's working envelope or touch the robot when the system is in operation.
- Use caution when moving the ASRS^2 and ASRS-36 robots by means of the teach pendant.
- To halt movement of the ACL robots which tend the ASRS^2 and ASRS-36 units, do any of the following:
- Press the Abort or Emergency Stop key on the teach pendant, or
- Press the controller's red Emergency button, or
- Use the ACL command A <Enter>.

ASRS-36u and ASRS-36uX2

The following list contains safety instructions for the use of ASRS-36u and ASRS-36uX2:

- Do not enter the robot's working envelope or touch the robot when it is in operation.
- Excercise caution when moving the ASRS-36u and ASRS-36uX2 robots by means of the teach pendant.
- To halt movement of the USB robot which tends the ASRS-36u and ,ASRS-36uX2 units, do any of the following:
- Press the Abort or Emergency Stop key on the teach pendant, or
- Press the controller's red Emergency button, or
- Use the Stop icon in the Scorbase software toolbar.

Note

Opening any of the plexiglass doors which enclose the ASRS^2 will automatically and immediately halt all movement of the ASRS robot. Upon closing the door, movement will resume.

Conveyor and PLC Safety

The following list contains the conveyer and PLC safety instructions:

- Be sure you know the location of the conveyor's power ON/OFF switch. *To immediately stop the conveyor, simply shut off this switch.*

- Keep hands and objects away from the conveyor drive unit.

- Do not tamper with the conveyor motor. Do not remove the conveyor motor covers under any circumstances.

- Do not touch or tamper with the power supply inside the PLC near the conveyor motors.

- Do not tamper with the switch and connector box for the 100/110/220/240/380V AC power supply (depending on device)

4

Installation

This chapter describes the hardware assembly processes, the wiring connections, the software installation and configuration procedures and system verification and more. It includes the following sections:

- **OpenCIM Installation Workflow**, describes the required order for installing the OpenCIM system components.
- **Hardware Installation**, describes the assembly instructions of the hardware components of the OpenCIM system.
- **Wiring,** describes the wiring connection procedures of the hardware components in the OpenCIM system.
- **Software Installation**, describes the network setup and the OpenCIM software installation and configuration.
- **Robot Positions and Homing**, describes how to teach the robot positions at the OpenCIM stations.
- **System Check**, describes the verification and adjustment procedures of the hardware components and device drivers.

OpenCIM Installation Workflow

The OpenCIM system is normally installed in the following order:

1. Hardware Assembly, described in Hardware Installation.
2. Wiring connections including network hardware, described in Wiring.
3. Software installation including software protection keys and network setup, described in Software Installation.
4. Teaching of robot positions, described in Robot Positions.
5. Checking and adjustment of devices, for standalone and system operation, described in System Check.

OpenCIM hardware configurations vary. This chapter presents the basic guidelines for setting up CIM equipment. Additional installation instructions will be provided separately for the specific equipment and configuration of your OpenCIM cell.

Hardware Installation

Before installing the OpenCIM, examine it for signs of shipping damage. If any damage is evident, contact your freight carrier, and begin appropriate claims procedures.

Make sure you have received all the items listed on the shipment's packing list. If anything is missing, contact your supplier.

For personal safety and for sufficient access to the stations from all open sides, a free area of at least one meter around each station is recommended.

Be sure you comply with all safety guidelines and warnings in the user manuals supplied with the robot, controller and other devices.

Some stations may have tables with slotted surfaces or predrilled holes to facilitate the mounting of the robots and devices.

Conveyor and Pallets

Refer to the instructions provided with your conveyor and/or OpenCIM cell.

Set up the conveyor within reach of the power supply and the air supply.

Assemble the conveyor and attach the conveyor buffers at each station.

Make sure the conveyor is assembled so that it moves clockwise or counterclockwise depending on your specific system layout.

Place the pallets supplied with the system anywhere along the conveyor with the arrow on the pallet pointing in the direction of movement of the conveyor.

Robots and Robot Controllers

The basic steps for installing a robotic system for use in the OpenCIM ce*ll are described below:*

1. Setup the robotic system according to the instructions in the User's Manual supplied with the robot/controller.
2. Interface the robotic system with relevant accessories (I/O devices and so on.)
3. Once the robotic system is functional, create a communication interface using the OpenCIM software.

For detailed instructions on connecting the robot and controller, refer to the User's Manual supplied with the robot/controller. In addition, refer to the instructions provided with your OpenCIM cell.

ASRS

Most ASRS systems are pre-assembled units that have to be placed near a conveyor station so that pallets can be loaded/unloaded. If you customize the ASRS make sure that the tending robot can reach all relevant elements and optimize the layout with respect to the tending time of the robot.

Refer to the instructions provided with your ASRS and/or OpenCIM cell.

Barcode Reader

The standard Barcode reader is normally mounted within robot reach on the conveyor next to the ASRS stop station in order to check outgoing and incoming template IDs. It requires a 5.6V DC power supply and a RS232 connection either to the Station PC or to the robot controller.

Refer to the instructions provided with your barcode reader and/or OpenCIM cell.

RFID Reader

RFID (Radio-frequency Identification) is a commonly used automatic identification technology that transmits an identification serial number using radio waves.

The standard RFID reader is normally mounted within robot reach on the conveyor next to the ASRS stop station in order to check outgoing template IDs. It requires a 12V DC power supply and an RS232 or USB port on the Station PC.

Refer to the instructions provided with your RFID reader and/or OpenCIM cell.

Pneumatic Devices

Pneumatic devices in a station (pneumatic door, fixtures, caliper, air blows, etc.) including position sensors are normally interfaced through the I/Os of the robot controller. They can also be controlled by a separate PLC if requested by the customer. In any case, pneumatic devices require the appropriate supply of compressed and conditioned air.

Refer to the instructions provided with the specific device and/or OpenCIM cell.

Palletizing Racks and Buffers

Normally, buffers are mounted on the conveyor but, for a specific station layout, it is possible to attach them to a table. Palletizing racks are used as a flexible storage that can be adapted to various parts by using sets of different sized pins. Make sure the pins are arranged identically for each part of the same type in the palletizing rack.

Refer to the instructions provided with the OpenCIM cell.

Templates

Templates are used to transport parts within the system. Templates, which have a grid of holes together with different sized pins, allow flexible adjustment for different parts in a similar way to the palletizing racks. Make sure the pins are arranged identically on all templates which will hold identical parts.

Refer to the instructions provided with the OpenCIM cell.

Wiring

Wiring depends on the actual stations, machines and devices included in your OpenCIM installation. Most systems are installed by a trained engineer who will supply you with specific wiring documentation (communication layout) to enable you to easily understand and maintain the system.

Refer to the documentation and wiring instructions provided with the OpenCIM cell.

Network

It is possible to operate the OpenCIM on any desktop or laptop PC which complies with the required hardware specifications in which a network board (adapter) has been installed, providing that you have installed the software driver which enables the adapter to work with the Windows Operating System. For further details, see Network Setup. Network wiring, which is documented in the communication layout, is performed through a hub.

Software Installation

This section describes how to set up the OpenCIM network as well how to install and configure the various OpenCIM software components such as the OpenCIM projects, the web viewer and more.

Network Setup

Once Windows has been installed and all PC devices (CD ROM drive, mouse, graphic card, network card, and so on) have been set up on each PC that is to be used in the OpenCIM network, set up the network as described in the following procedure:

To set up the network:

1. Define the Network Workgroups and computer names. Computer names are case-sensitive. It is recommended that you use names such as these:

Network Workgroup	CIM
CIM Manager PC	CIM-MANAGER
Station PCs	CIM-PC1, CIM-PC2, etc. on which the logical workstations (WS1, WS2, etc.) of the CIM are located

2. Set the TCP/IP network protocol as your communication protocol. Select the IP address as follows: If your network Workgroup is part of a global network (i.e. is connected to a server) choose the option: Obtain an IP address automatically.
   If your network workgroup is local (i.e. not connected to a server) set the IP address, for example, as 200.1.1.1 and increment the last digit for each additional PC.
   For all PCs in the CIM, specify the same subnet mask, for example, 255.255.0.0. For details, contact your network administrator.

3. If you are using more than one PC, it is a good idea to verify that the PCs are connected to the network.

To see the names of the connected PCs, follow one of the procedures outlined below.

- *In Windows XP*:

   Click **Start** | **My Network Places** | **Entire Network** (In the *Other Places* section area on the left of the window.)

- *In Windows Vista/7*:

   Click Start | Network

   If you don't see the PC names, wait a few seconds and press **F5**. If the names are still not displayed, check the network setup again.

Installing the OpenCIM Software

After setting up the network, the next step is to install the OpenCIM software. The following procedure describes how to install one of the OpenCIM products (OpenCIM, OpenCIM Offline, OpenFMS). You can cancel the setup process at any time by pressing Cancel.

Notes

By default, the software will be installed in the path X:\program files\intelitek\opencim (where X is the system drive) . If you already have a previous version of the software in a directory of this name, select a different directory name for the new installation. Installation of multiple products on the same PC is permitted provided that you have a license for each product.

To install the OpenCIM software:

1. Close all open applications and insert the OpenCIM software CD into the CD ROM drive.

   - Autorun begins the installation process as soon as you insert the CD unless you hold the SHIFT key while you insert the CD.

   - If you have turned off Autorun, you must specify the CD ROM drive from which to run the installation setup file which will begin the installation. From the Windows Start menu, choose **Run**. Enter:

     X:\install\setup [where X represents the CD ROM drive].

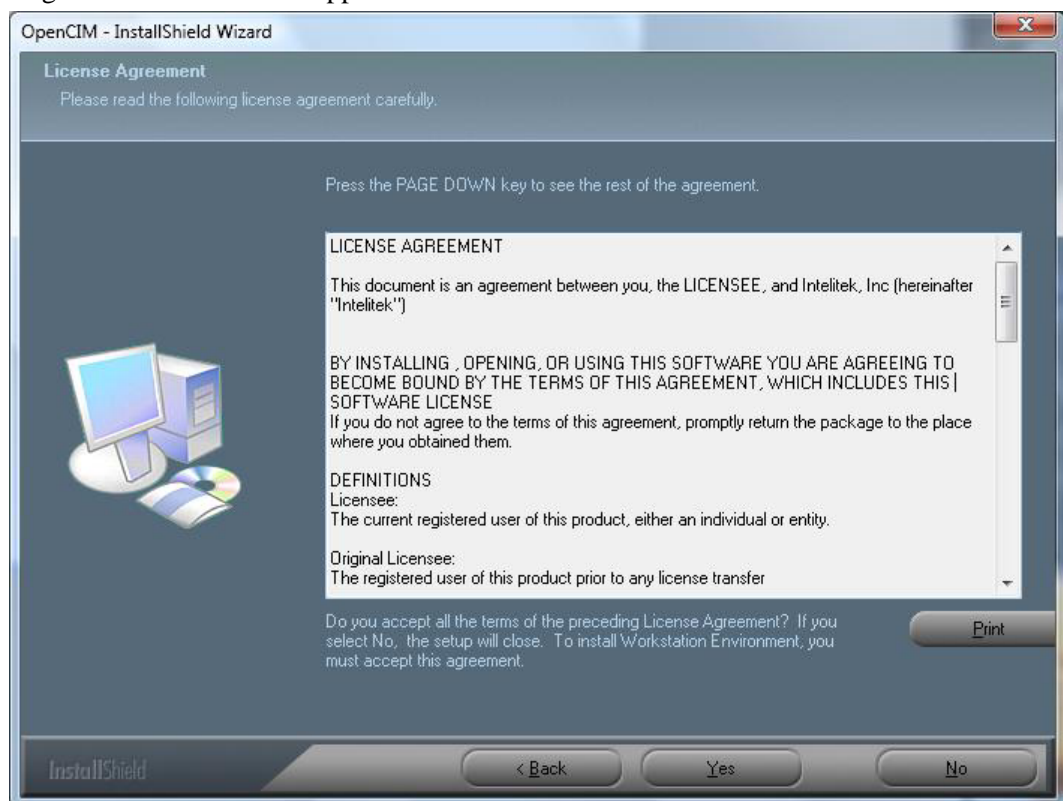2. In the Welcome window of the Installation Wizard choose **Next**. The License Agreement window will appear.

Figure 4-1: License Agreement Window

3. Review the Intelitek software license agreement. You must accept the terms of this agreement in order to complete the installation. If you do not accept the agreement, you cannot proceed with the installation. To accept, click **Yes**. The Installation Mode window will appear.
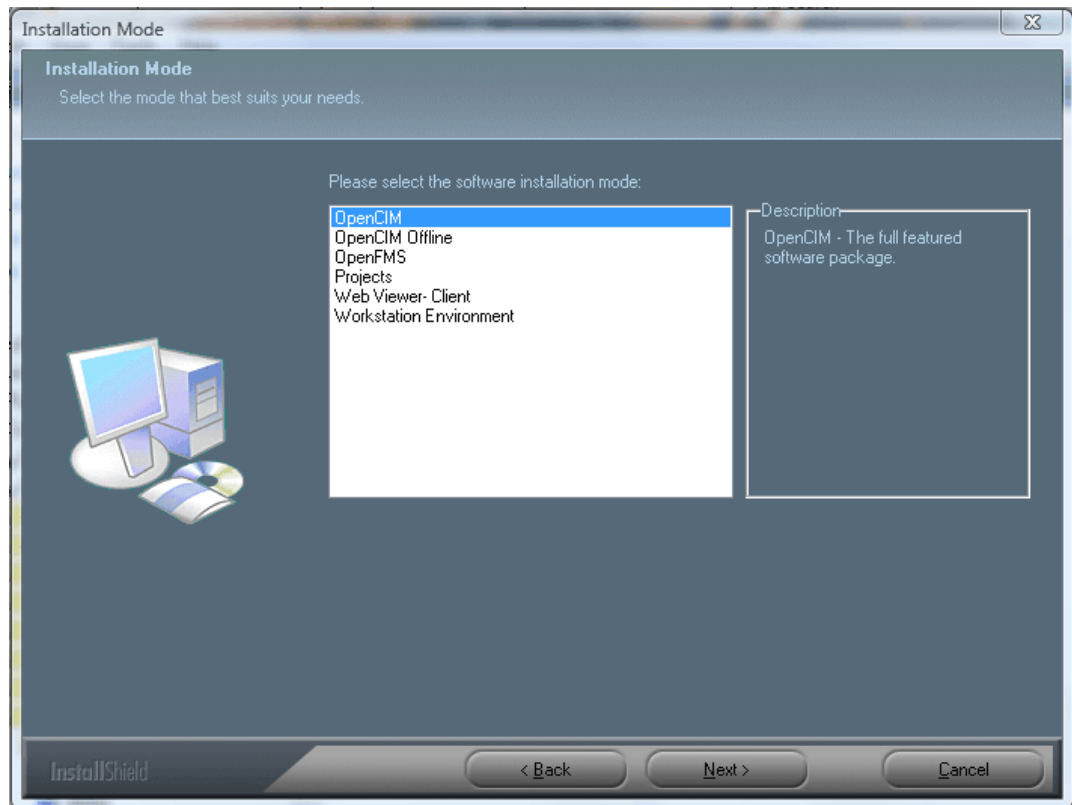

Figure 4-2: Installation Mode Window

4. Select the OpenCIM product that you want to install and click **Next**. The Project Selection window will appear.

Note *The WorkStation Environment option is used (by Intelitek technical support personnel) to install system components necessary to run device drivers from a PC which is not the CIM Manager PC. For further details, see Software for WorkStation PCs.*
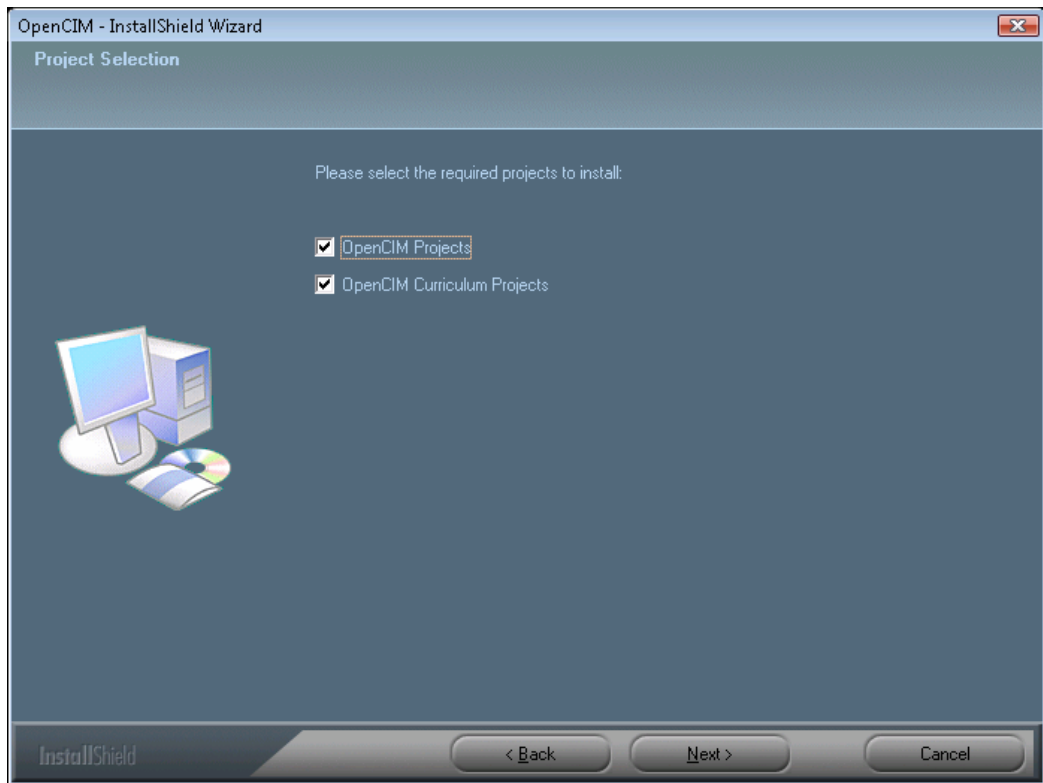

Figure 4-3: Project Selection Window

5. Select the required projects to install.

6. Select the required option as follows:

- Select **OpenCIM Projects** to add the default OpenCIM projects to your existing OpenCIM installation.

- Select **OpenCIM Curriculum Projects** to add the OpenCIM Curriculum Projects to your existing OpenCIM installation.

Note *You can select one or both of the above.*

7. Click **Next.** The Web Viewer Configuration window will appear.
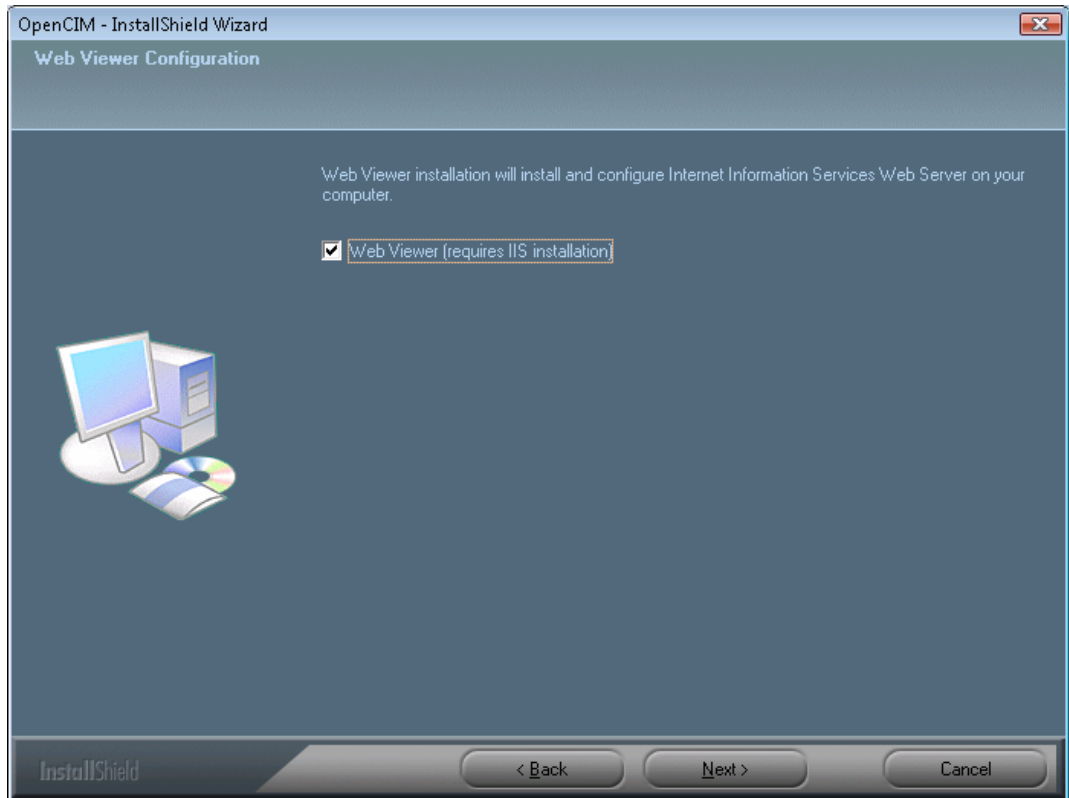
Figure 4-4: Web Viewer Configuration Window

8. Check the Web Viewer box and click **Next** to add the Internet Information Services (IIS) Web Server (necessary for the web viewer), and to configure the Web Server to work with the Web Viewer. The Choose Destination Location window will appear.

Note   *In XP, for the Web Server to be installed, the installation CD must be inside the computer.*
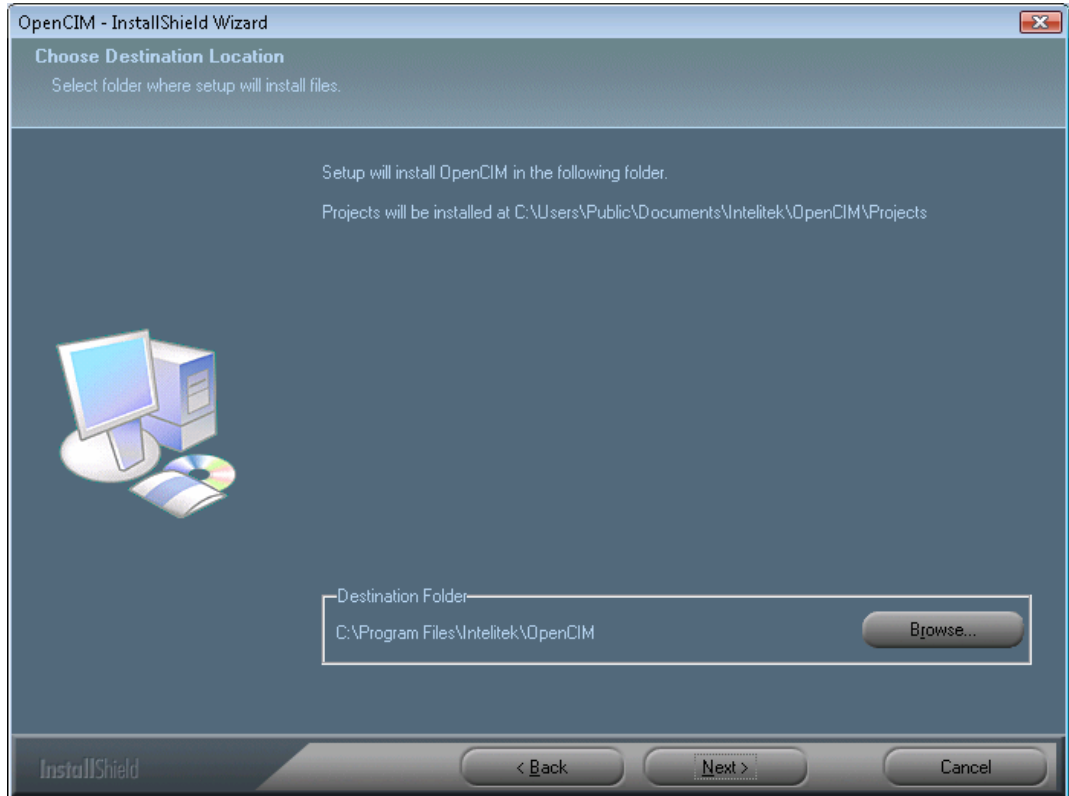

Figure 4-5: Choose Destination Location Window

9. Click **Next** to install to the default folder, or click **Browse** and choose another folder. The Select Program Folder window will appear.
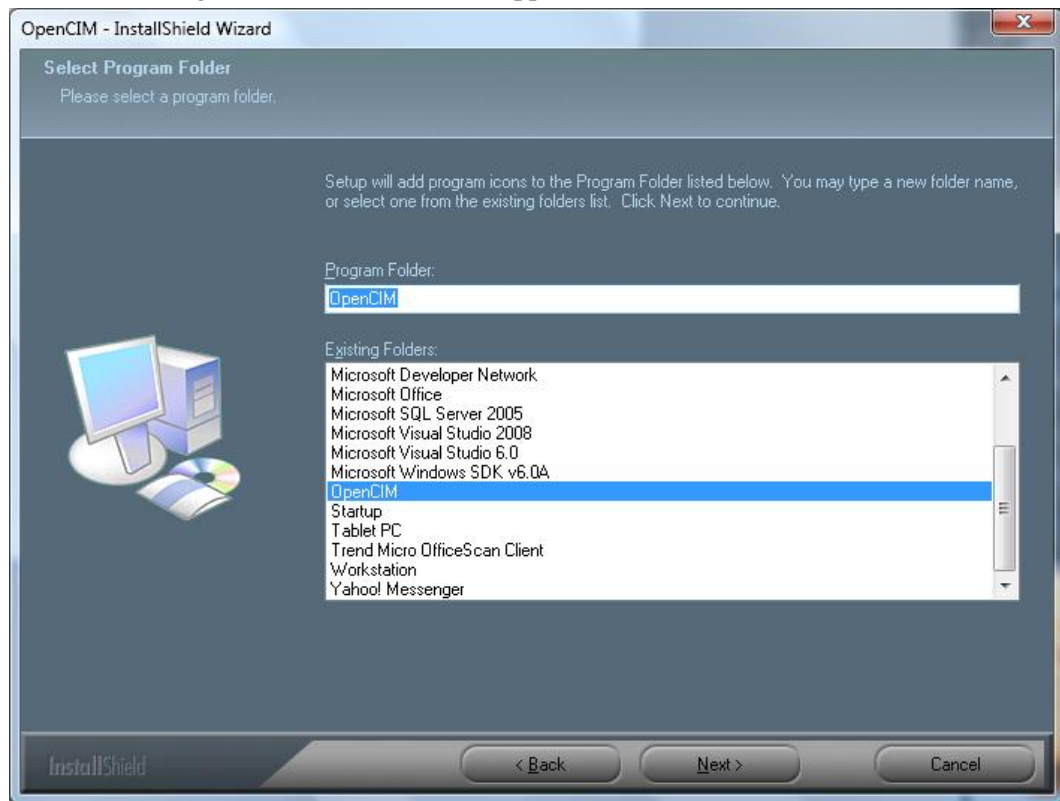
Figure 4-6: Select Program Folder Window

10. Click **Next** to accept the default Program Folder name. (You may also type a new folder name or select another folder from the existing folders list.)

11. The installation progress bar is displayed on the screen. When prompted by the on-screen instructions, click Ok to install the Project Manager launch icon on your desktop.

12. The Installation Complete window is displayed when installation has been successfully completed as shown in Figure 4-7.
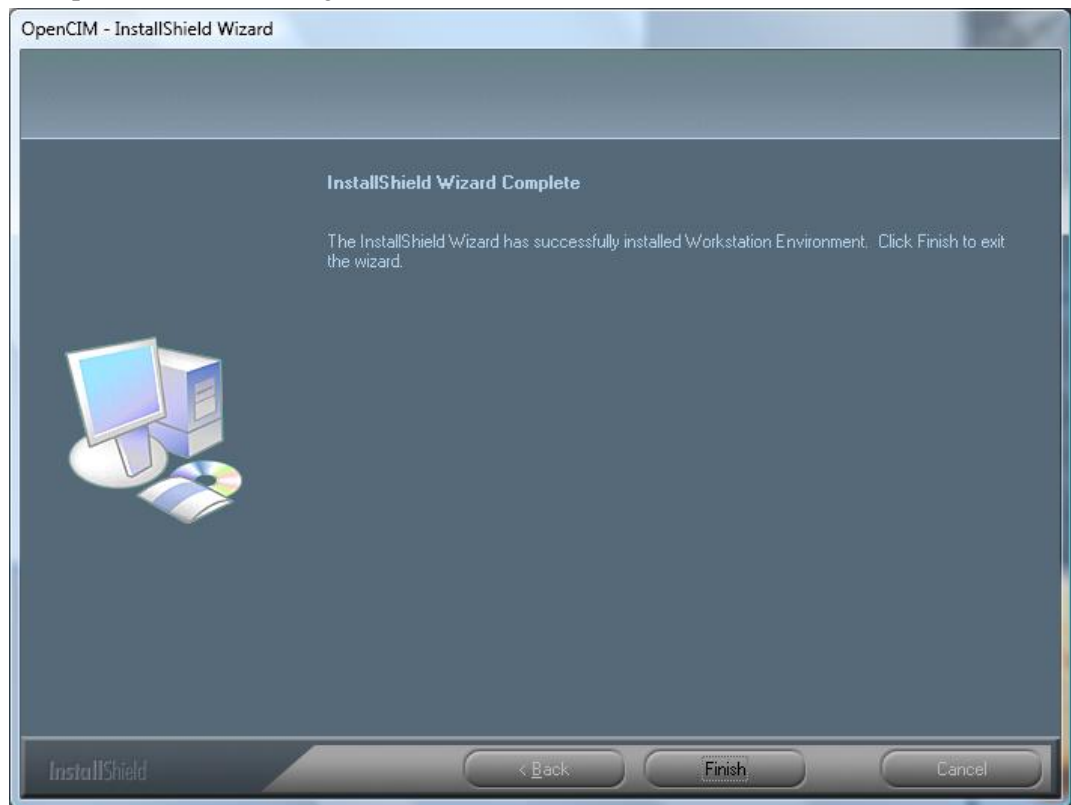
Figure 4-7: Installation Complete Window.

13. Click **Finish** to exit the InstallShield Wizard.
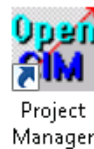
The Project Manager launch icon is added to the Desktop.

Project Manager can also be initialized by clicking on Project Manager which can be accessed by clicking **Start | All Programs | OpenCIM**.

The OpenCIM Applications, **CIM Manager** and **CIM Setup** are both activated from the **Project Manager** application which can be launched from the desktop shortcut icon or from the OpenCIM popup menu

Refer to *Chapter 5, CIM Project Manager* for further details.

A shortcut to the folder containing the projects that were installed is also added to the Desktop and the Start menu.

Installing OpenCIM Projects

The OpenCIM Installation wizard enables you to add OpenCIM Projects to an existing installation, as described in the following procedure.

To install OpenCIM Projects:

1. Close all open applications and insert the OpenCIM software CD into the CD ROM drive.

   - Autorun begins the installation process as soon as you insert the CD unless you hold the SHIFT key while you insert the CD.

   - If you have turned off Autorun, you must specify the CD ROM drive from which to run the installation setup file which will begin the installation. From the Windows Start menu, choose **Run**. Enter:

     X:\install\setup [where x represents the CD ROM drive].

2. In the Welcome window of the Installation Wizard choose **Next**. The License Agreement window will appear.
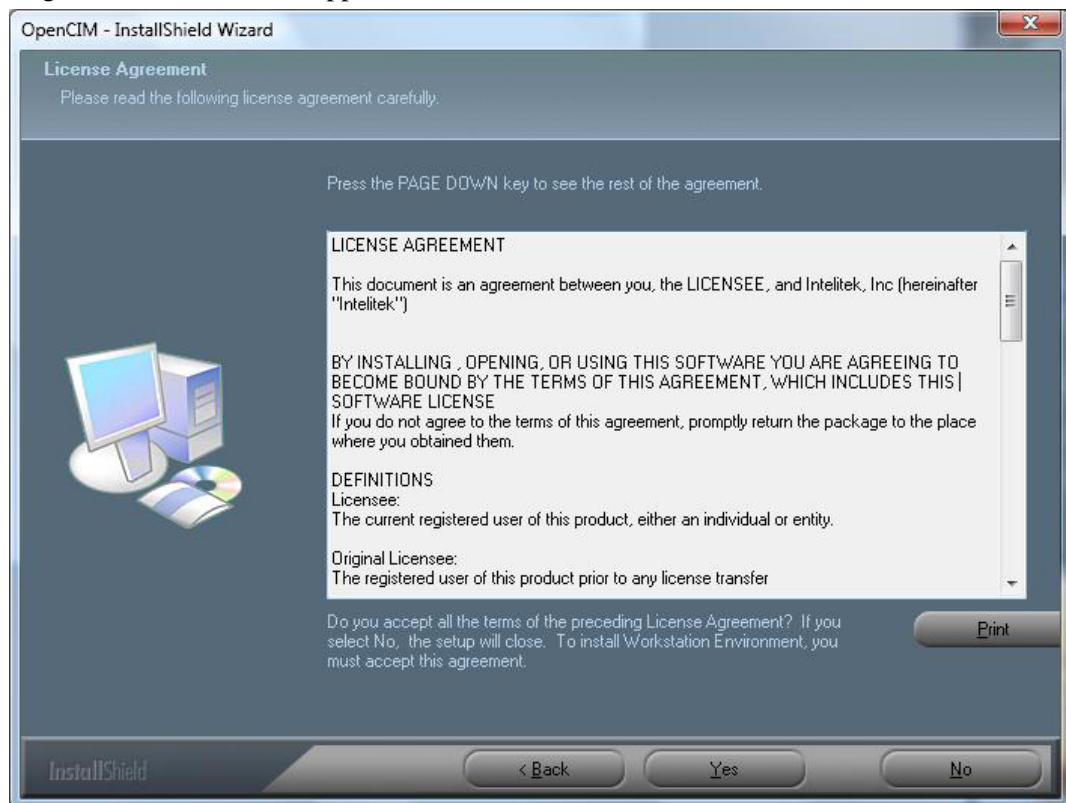
Figure 4-8: License Agreement Window

3. Review the Intelitek software license agreement. You must accept the terms of this agreement in order to complete the installation. If you do not accept the agreement, you cannot proceed with the installation. To accept, click **Yes**. The Installation Mode window is displayed.
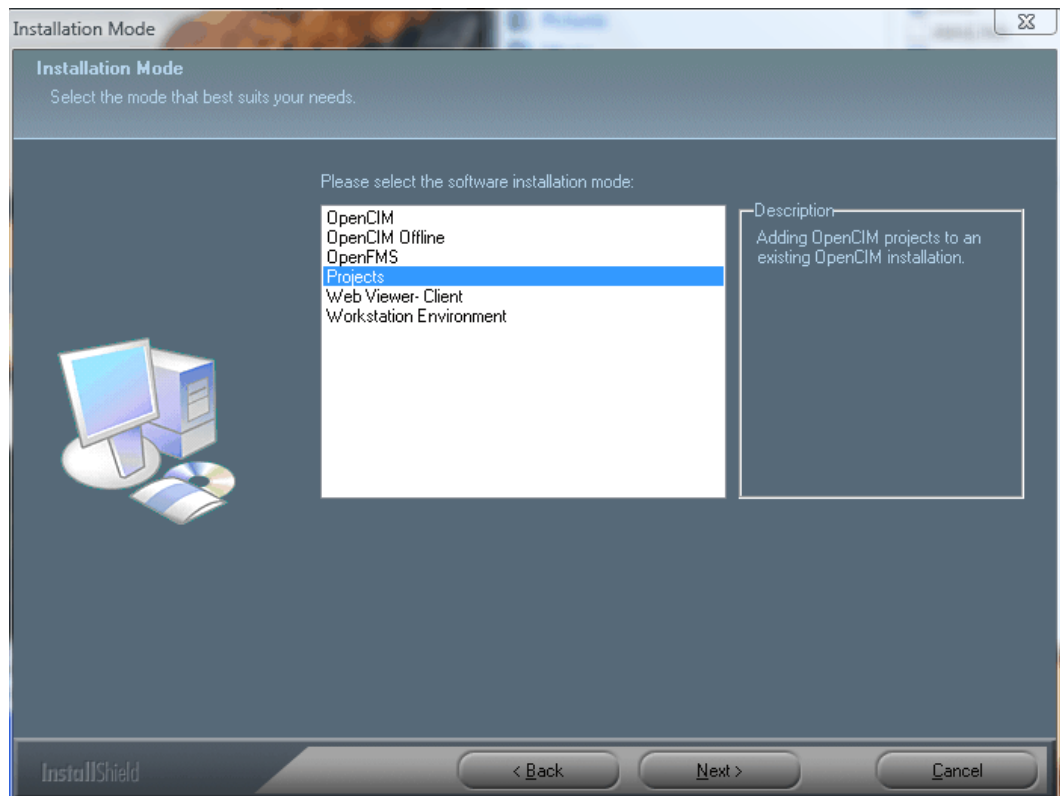
Figure 4-9: Installation Mode Window. Installing Projects.

4. Select Projects and then choose **Next**. The Project Selection window is displayed.
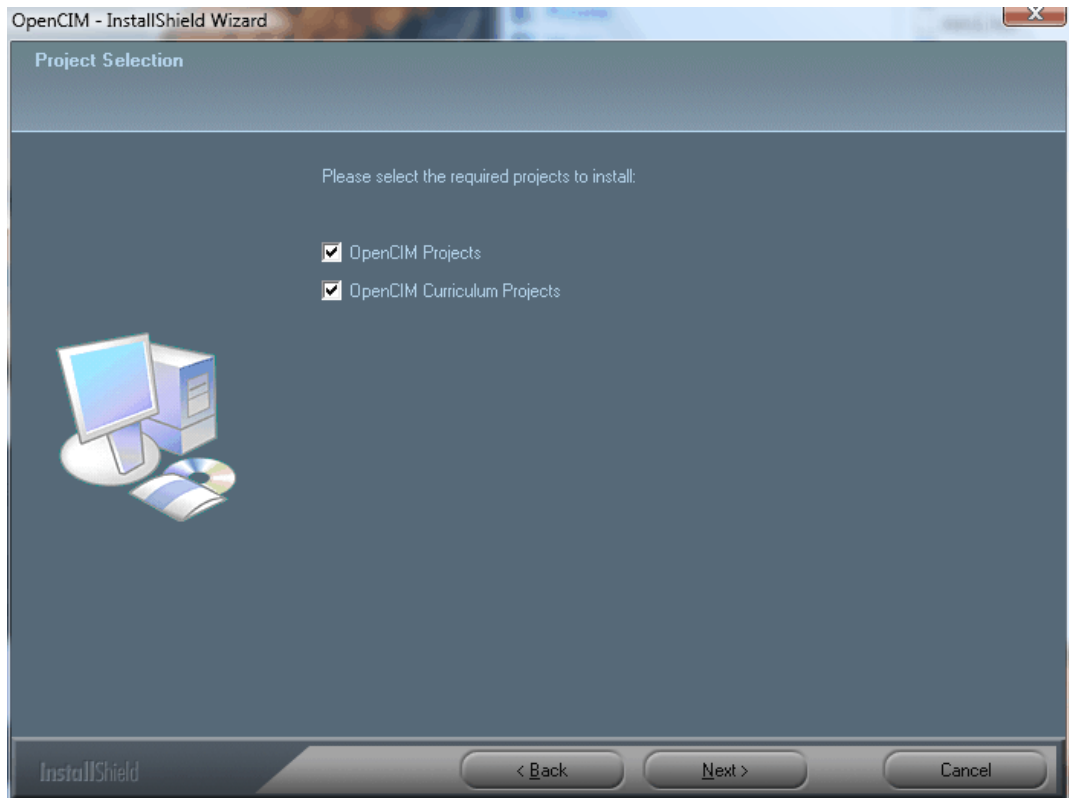
Figure 4-10: Project Selection Window

5. Select the required option as follows:

- Select **OpenCIM Projects** to add the default OpenCIM projects to your existing OpenCIM installation.

- Select **OpenCIM Curriculum Projects** to add the OpenCIM Curriculum Projects to your existing OpenCIM installation.

Note *You can select one or both of the above.*

6. Click **Next**. The Project Installation Mode window is displayed containing the software installation modes that contain the project or projects that were chosen in the previous step).
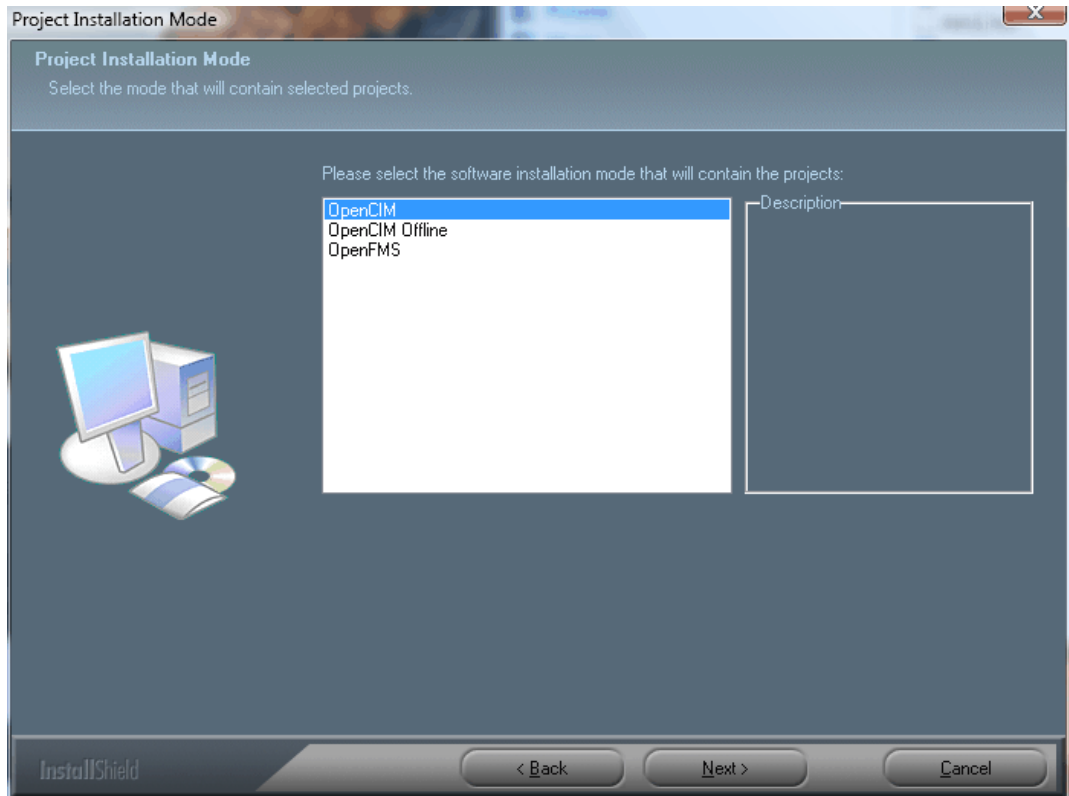
Figure 4-11: Project Installation Mode Window

7. Select the appropriate project installation mode and Click **Next**. The OpenCIM Installation wizard begins to install the selected projects onto your computer. When the installation is complete. The InstallShield Wizard Complete window is displayed.

8. Click **Finish** to exit the Wizard. The selected projects will be displayed in the **Archive** tab, the next time you activate the **Project Manager** application.

Notes

The default destination for the projects is:

Windows XP: [default system drive (for example C:)]\Documents and Settings\All Users\Documents\Intelitek\OpenCIM\Projects

Windows Vista/7: [default system drive (for example C:)]\Users\Public Documents\Intelitek\OpenCIM\Projects

Manual Web Viewer Installation

It is recommended that you choose to install and configure the Internet Information Server (IIS) automatically by checking the box in the Web Viewer Configuration window which appears during the OpenCIM installation procedure.

The following section describes how to install the Internet Information Server (IIS) and configure the Web Viewer manually.

After installing the OpenCIM software, the Web Viewer can be installed and configured manually as shown in the steps outlined below:

- **Installing the IIS**, describes the installation procedure for the Internet Information Server (IIS).
- **Configuring the Web Viewer**, describes the configuration of the web server in Windows XP and Windows Vista/7.

Installing the IIS

In Windows XP:

To install the IIS manually in Windows XP, you are required to insert the Windows XP Operating System CD into the CD drive and follow these steps:

1. From the Windows Start menu select Settings | Control Panel | Add or Remove Programs
2. Select Add/Remove Windows Components

The Windows Components Window is displayed as shown in Figure 4-12.
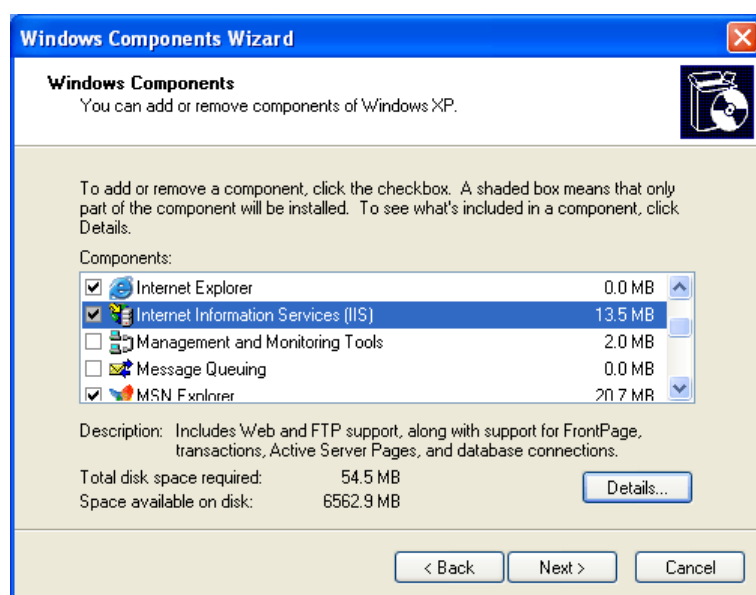
Figure 4-12: Windows Components Window

3. Select the **Internet Information Services (IIS)** checkbox and click **Next** to proceed to the next step where you will configure the components.

4. When the configuration is complete the final window is displayed.

Figure 4-13: Windows components Wizard-Final Window

5. Click **Finish** to complete the installation.

In Windows Vista/7:

1. From the Start menu select Control Panel | Programs | Programs and Features | Turn Windows Features On or Off (In the Tasks area on the left of the window.)

2. Click **Continue** on the security dialog box or if you are an administrator, enter your administrator account information and click **Continue.** The Windows Features window is displayed.
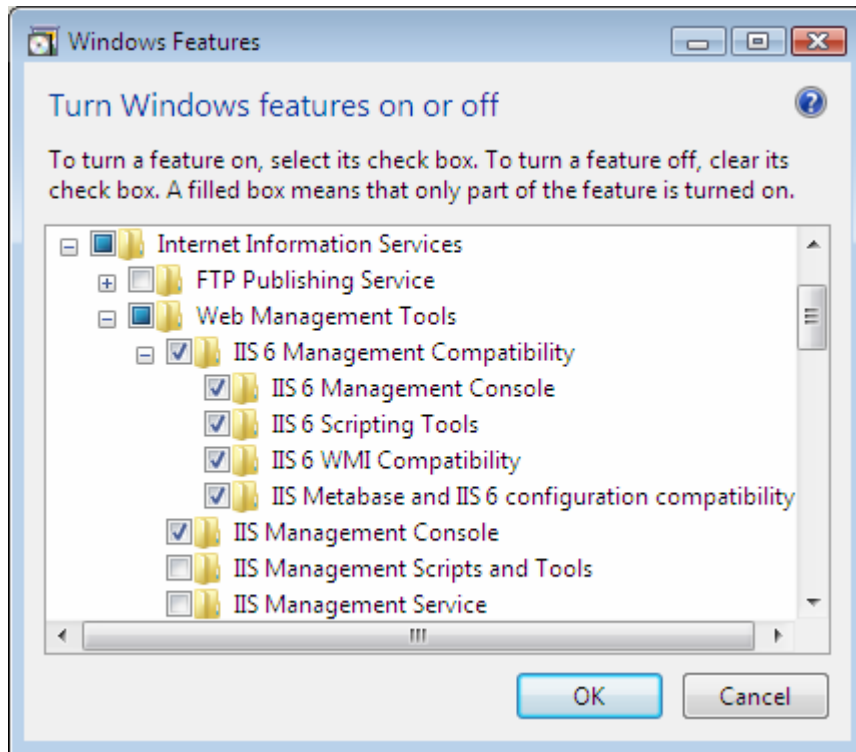
Figure 4-14: Windows Features Window – Internet Information Services

3. Expand **Internet Information Services | Web Management Tools | IIS 6 Management Compatibility. V**erify that all its branches are checked as shown in Figure 4-14.

4. Expand Internet Information Services | Web Management Tools | World Wide Web Services | Application Development Features and check the following branches:

- ASP
- ISAPI Extensions

The screen should now appear as follows:
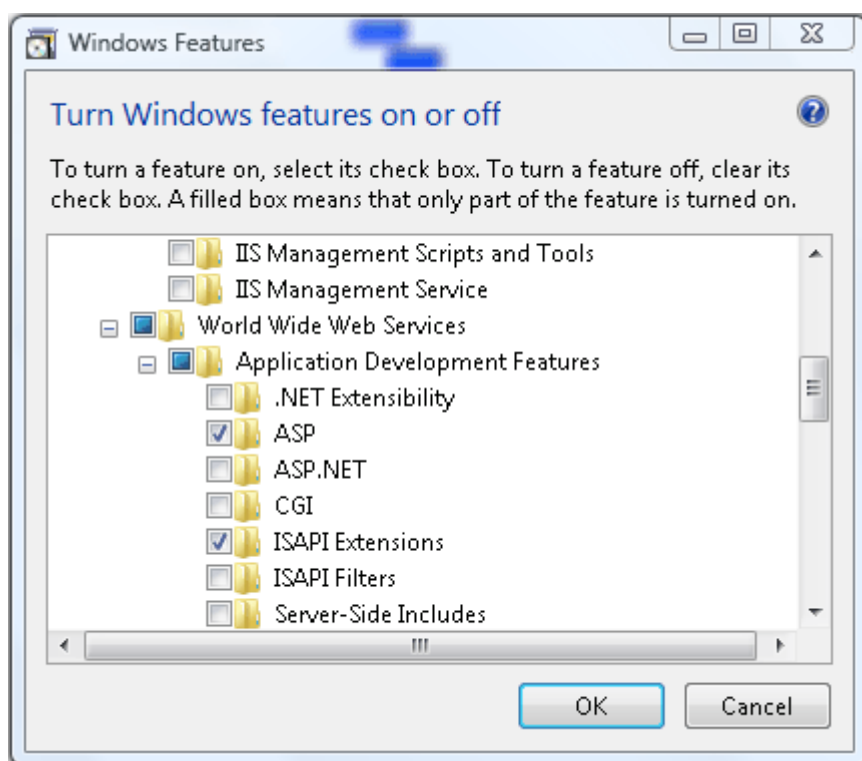
Figure 4-15: Winows Features Window- World Wide Web Services

5. Click **OK**.

6. After the IIS installation is complete, you can add an Internet Information Services icon on your desktop, as follows:

   - From your windows **Start** menu, select **Control Panel** | **System and Maintenance** | **Administrative Tools** The Administrative Tools window is displayed.

   - Right-click [image] **Internet Information Services IIS Manager.** From the displayed popup menu select **Create Shortcut**. When prompted, click **Yes** to create a shortcut on your desktop.

To restart the IIS press the ▶ *Start button. To stop the IIS server press the* **Stop** ■ *button.*

Note

Configuring the Web Viewer

After installing the IIS manually, the next step is to configure the Web Viewer. When reinstalling or upgrading the Open CIM application, the configuration procedures described in this section are not required.

This section describes how to configure the Web Viewer in Windows XP, Vista and 7.

In order for the Web Viewer to work when the firewall is enabled, Port 80 in the firewall must be opened.

Note

Configuring the Web Viewer in Windows XP, Vista and 7

The following procedure describes how to configure the web viewer in the Windows XP, Vista or 7 operating systems.

To configure the web viewer in Windows XP/Vista/7:

1. From your windows **Start** menu, select **Run**. The Run window is displayed.

2. Type **dcomcnfg** in the Open field and click **OK**. The Component Services window is displayed as shown in Figure 4-16.

In Windows Vista/7, a security dialog box will be displayed. Click Continue (if you are not the Administrator you will be prompted to enter your user name and password to continue).
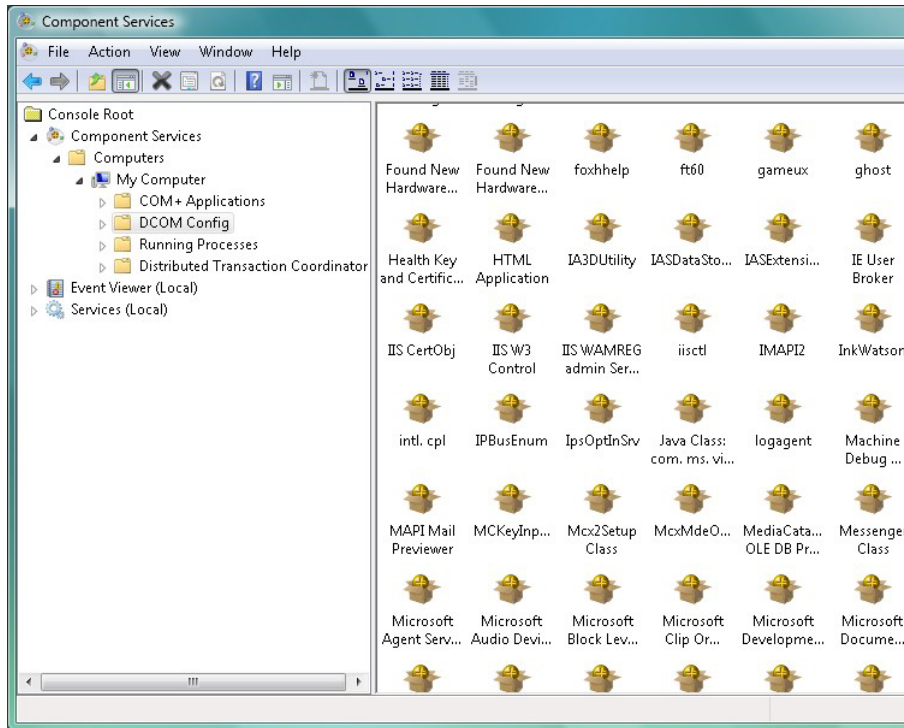
Figure 4-16: Component Services Window

3. Browse to **Component Services | Computers | DCOM Config** in the tree menu in the left pane.

4. Right-click **Manage Document** and select **Properties**. The Manage Document Properties window appears displaying the **General** tab.
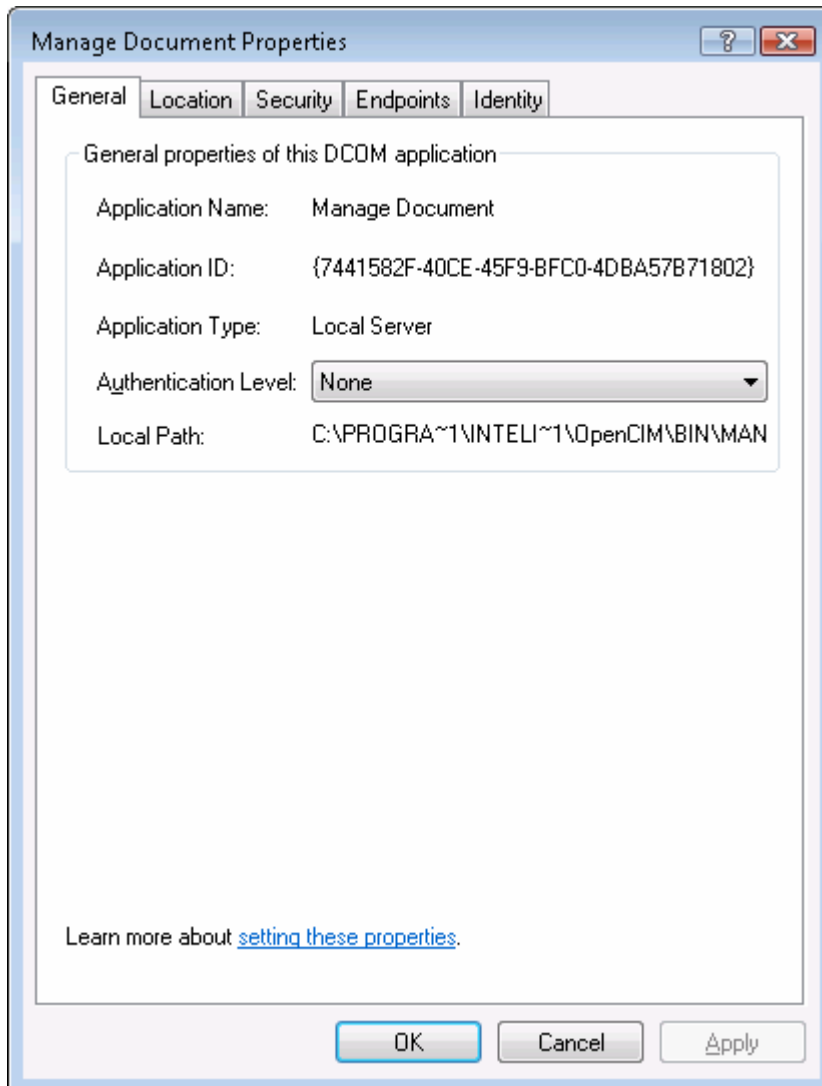
Figure 4-17: Manage Document Properties Window- General Tab

5. From the Authentication Level dropdown list, select **None**.

6. Select the **Identity** tab, as shown in Figure 4-18.
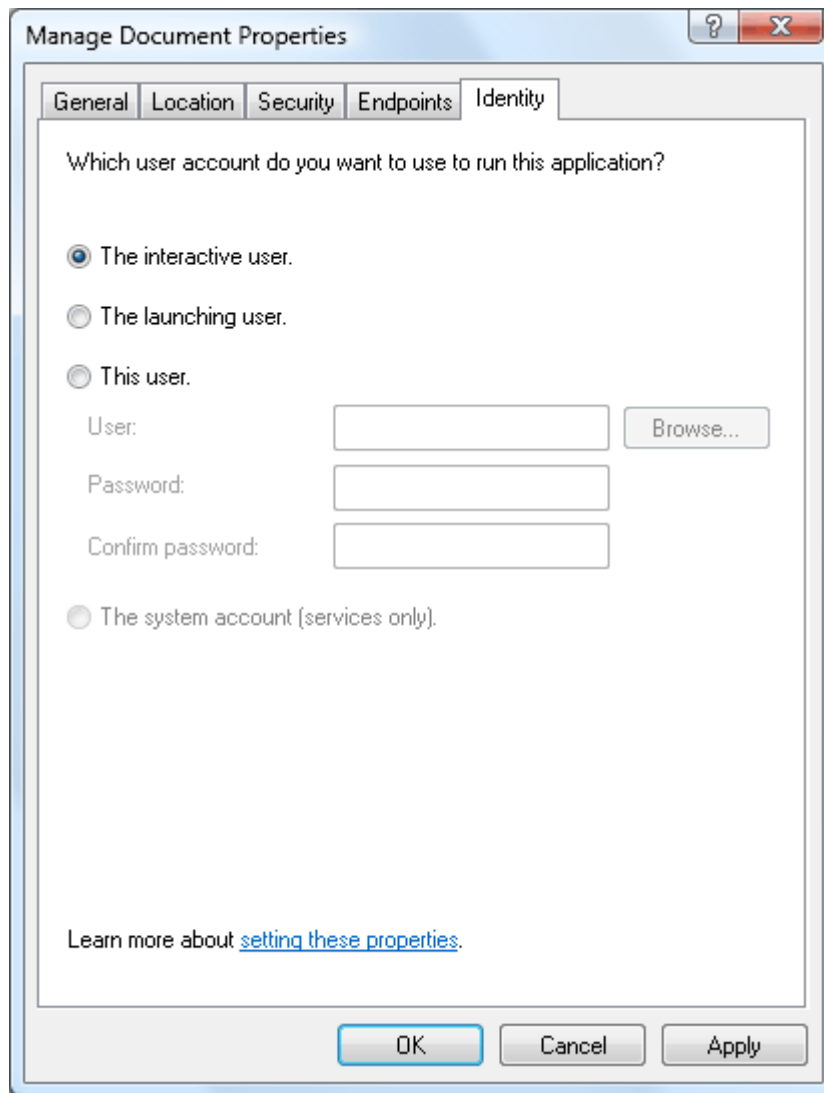
Figure 4-18: Manage Document Properties Window-Identity Tab

7. Select **The interactive user** and click **Apply** and then **OK** to close the Management Document Properties window.

To change security settings, repeat steps 1-5 above, and proceed as follows:

1. Select the Security tab as shown in Figure 4-19:
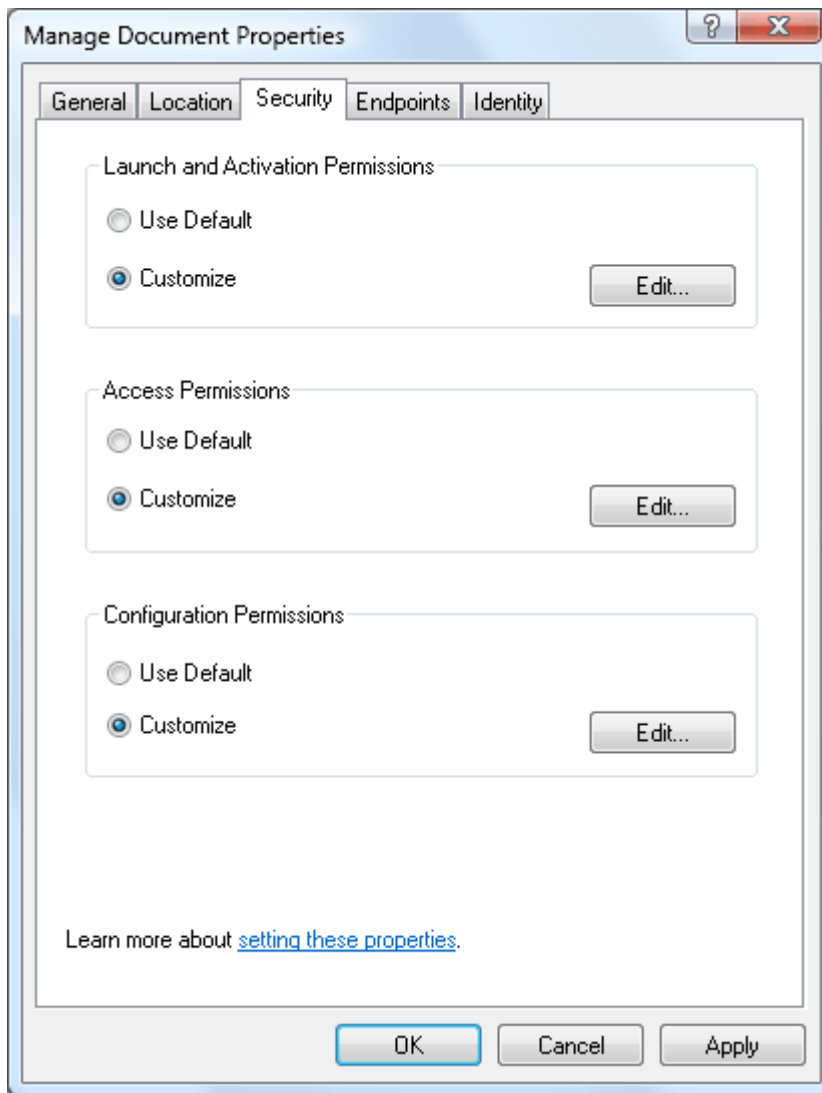
Figure 4-19: Manage Document Properties-Security Tab

2. In the Launch and Activation Permissions area, select **Customize** and click **Edit**. The Launch and Activation Permission window is displayed. (In Windows XP, this window is called the Launch Permission window).
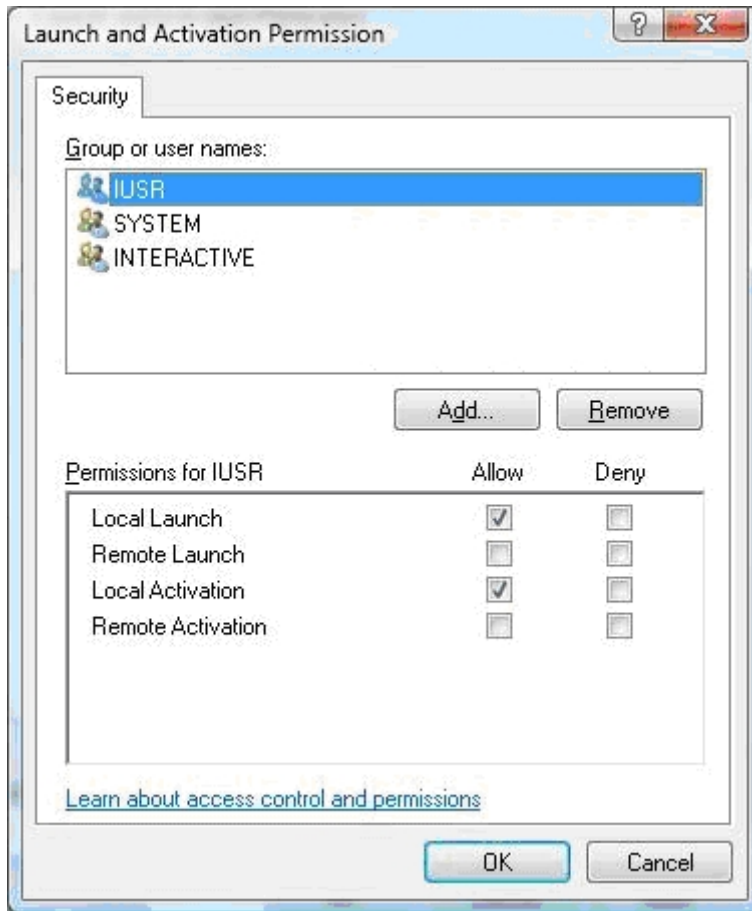
Figure 4-20: Launch and Activation Permission Window

3. Select the **Allow** check boxes for Local Launch and Local Activation.

4. Click Add | Advanced | Find Now

5. Scroll through the list of search results to find the following user:

   • In Windows XP: IUSR_<your computer name>.

   • In Windows Vista/7: IUSR

6. Double click the name.

7. Click **OK**.

8. Return to the Manage Document Properties window. In the Access permissions area in the Security Tab, select **Customize** and click **Edit**.
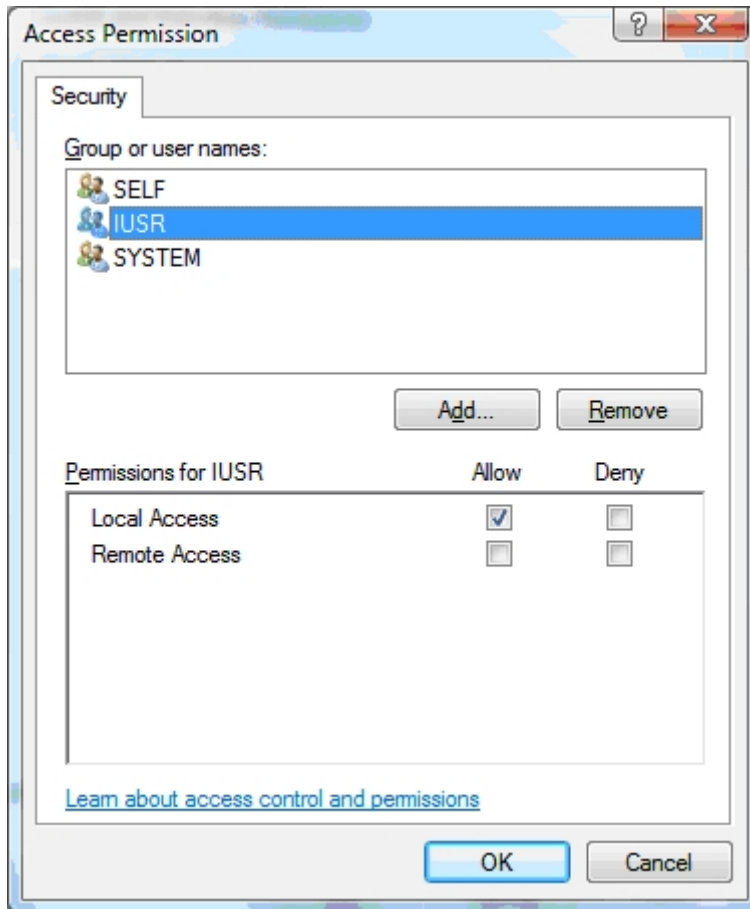
Figure 4-21: Access permissions Window

9. Repeat steps 4 to 7.
10. Select **Allow** for Local Access in the Access Permissions window.
11. Click **OK**.
12. Restart your computer, in order for the changes to take effect.

Before you can activate the Web Viewer, you must activate the Web Viewer in the CIM

Modes window, accessed from the Modes [icon] button on the CIM Manager toolbar, as described in Modes Dialog Box in Chapter 6, Operating CIM Manager. In addition, the HTTP protocol must be enabled in order to access the Web Viewer.

Software for Workstation PCs

In order to be able to use OpenCIM modules on a station PC, you must install the Workstation software from the OpenCIM installation CD.

Note   *For workstation PCs which require Viewflex and/or Scorbase software, it is highly recommended that they be installed before the installation of the Workstation Environment.*

They can however be defined manually within the project directory by modifying the configuration files which they use. Refer to Chapter 12 OpenCIM Loader: DDLoader.EXE for more details.

Note   *Microsoft .Net Framework 4.0 is required on the client PC for the use of the OpenCIM RFID driver, and will be installed automatically if it is not already installed on the workstation PC.*

Installing the Workstation Software

To install Workstation software:

1. Close all open applications and insert the OpenCIM software CD into the CD ROM drive.

   - Autorun begins the installation process as soon as you insert the CD unless you hold the SHIFT key while you insert the CD.

   - If you have turned off Autorun, you must specify the CD ROM drive from which to run the installation setup file which will begin the installation. From the Windows Start menu, choose **Run**. Enter:

X:\install\setup [where x represents the CD ROM drive].

2. In the Welcome window of the Installation Wizard choose **Next**. The License Agreement window will appear.
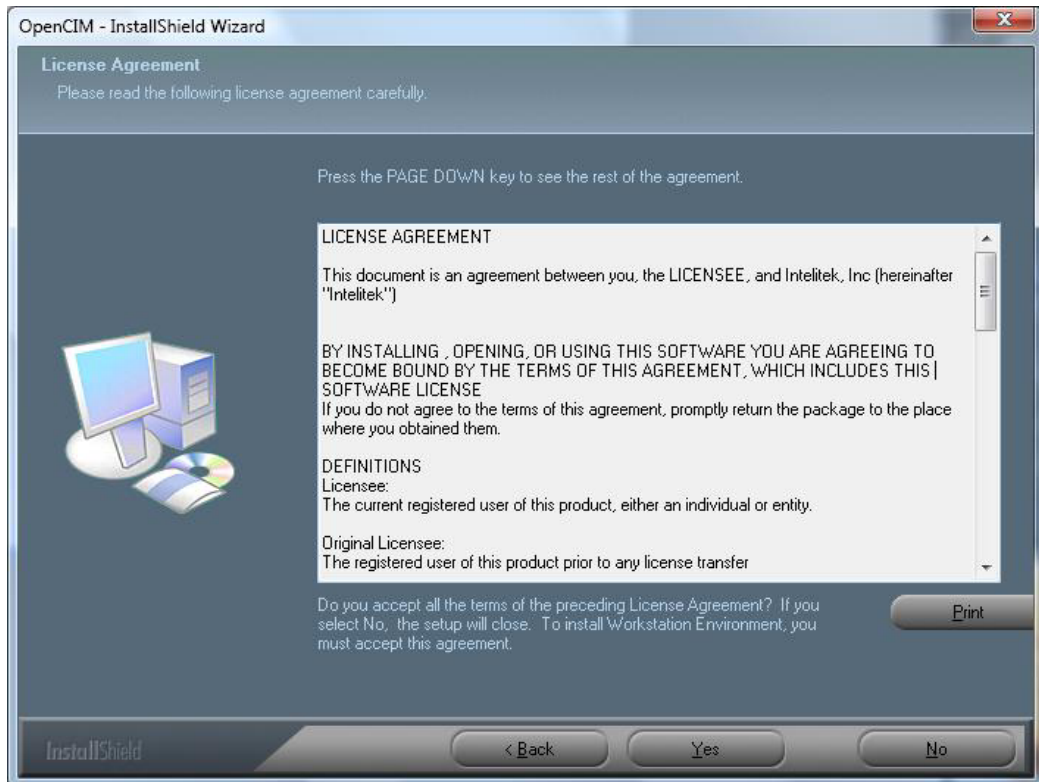
Figure 4-22: License Agreement Window

3. Review the Intelitek software license agreement. You must accept the terms of this agreement in order to complete the installation. If you do not accept the agreement, you cannot proceed with the installation. To accept, click **Yes**. The Installation Mode window will appear.
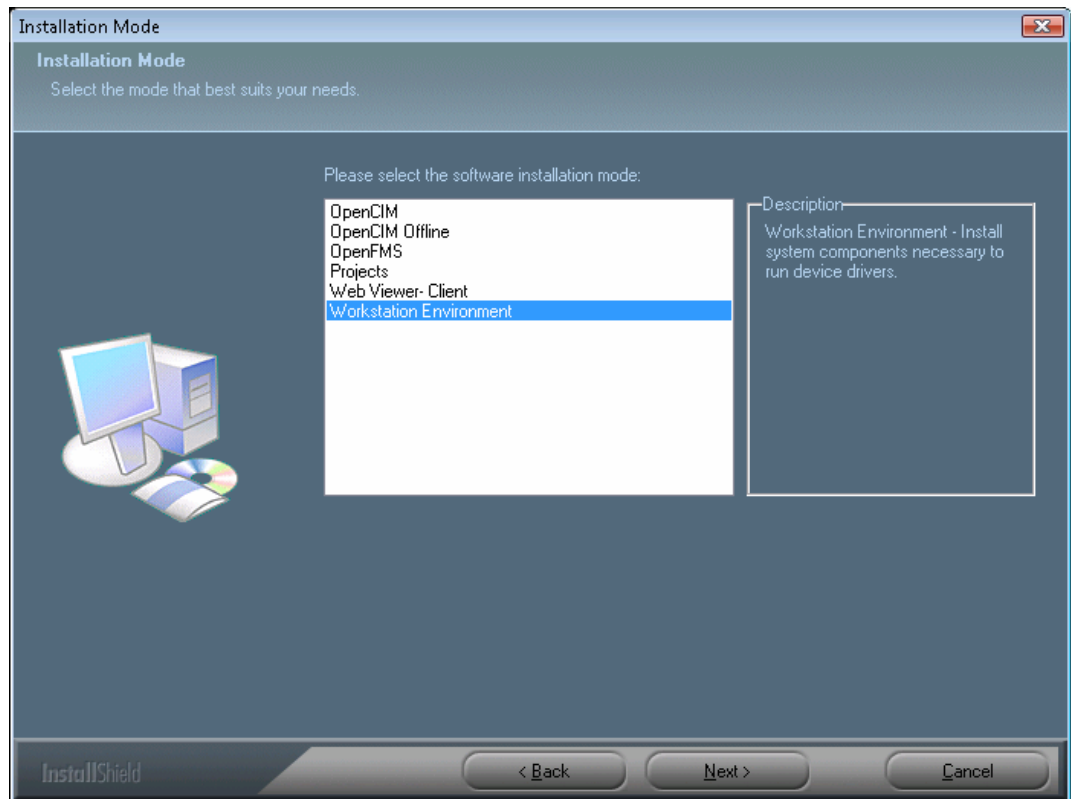
Figure 4-23: Installation Mode Window

4. Select **Workstation Environment** and click **Next**. The Choose Project window will appear.
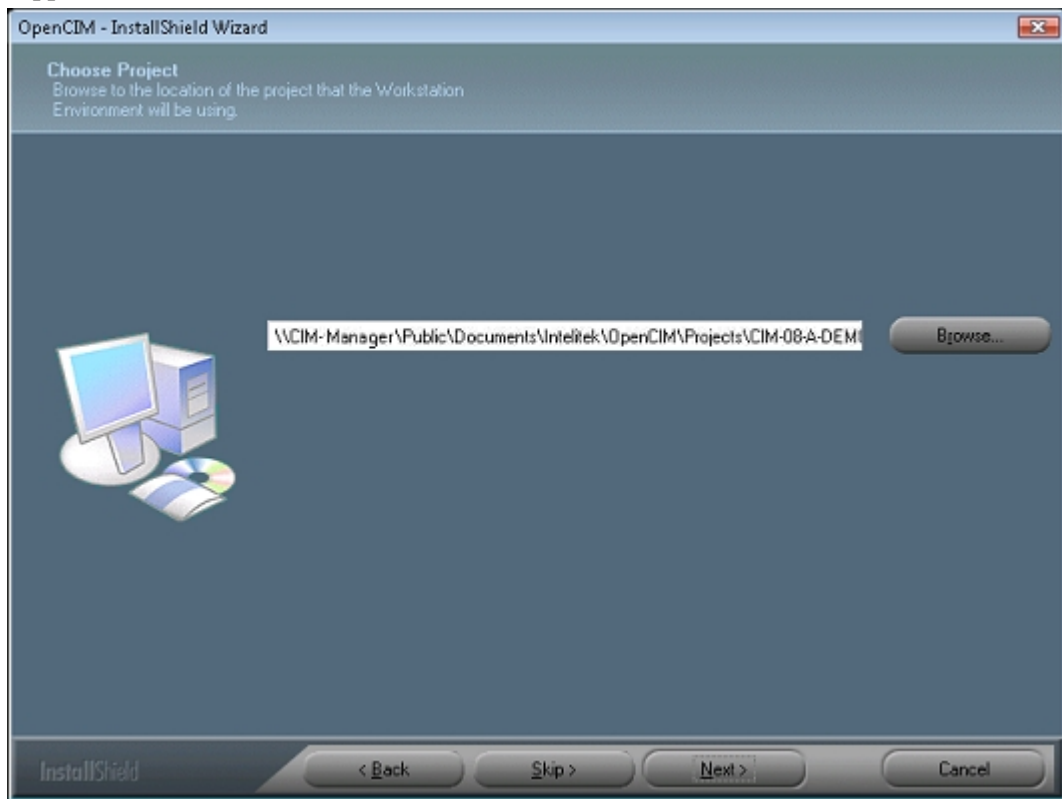
Figure 4-24: Choose Project Window

5. Browse to the location of the project that the workstation environment will be using and click **Next**. The Select a Workstation window will appear.
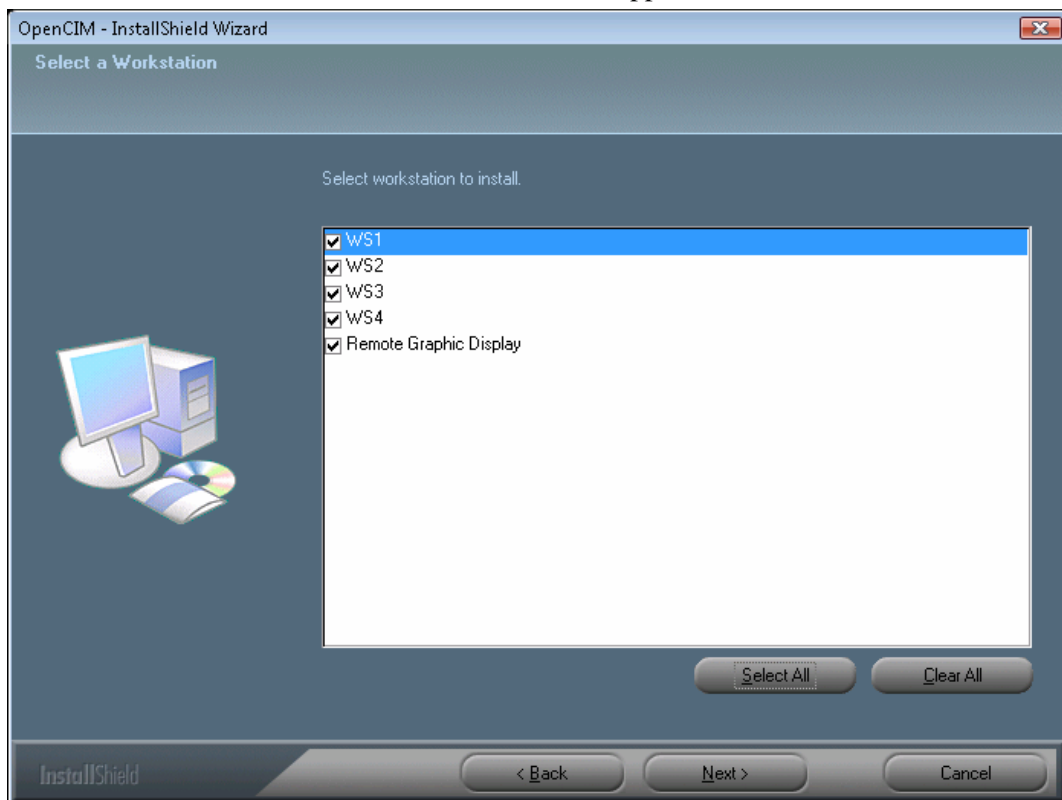
Figure 4-25: Select a Workstation Window

6. Select the workstation to install, and Remote Graphic Display if desired (Refer to *Chapter 6: Operating CIM Manager* for information on configuring the Remote Graphic Display). Several Workstations may be installed on one computer. Click **Next**.

- If the workstation you are installing uses Viewflex, the Viewflex Path window will appear. Refer to the next step.
- If the workstation you are installing uses Scorbase, the Scorbase Path window will appear. Refer to step 8.
- If the workstation does not require Viewflex nor Scorbase, skip to step 9.
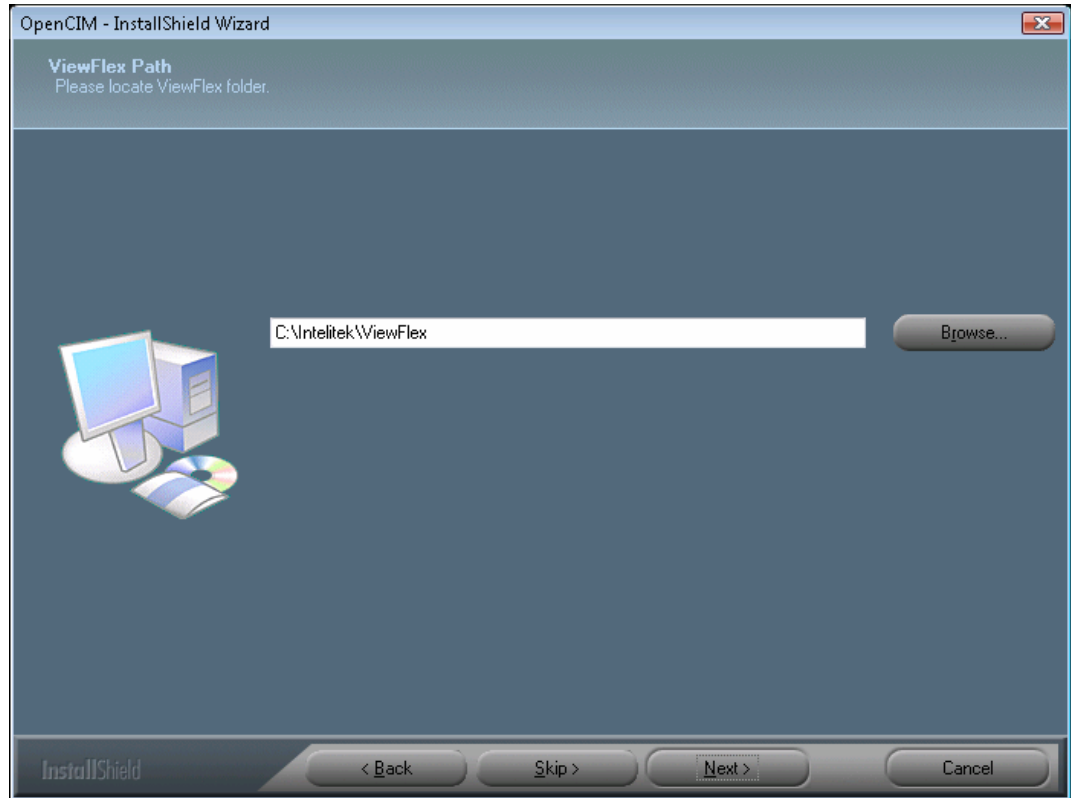
Figure 4-26: Viewflex Path Window

7. Browse to the location of Viewflex on your system and click **Next**. The Viewflex Path window will appear.

Note   *If you have not yet installed Viewflex, click Skip to continue the installation process.*
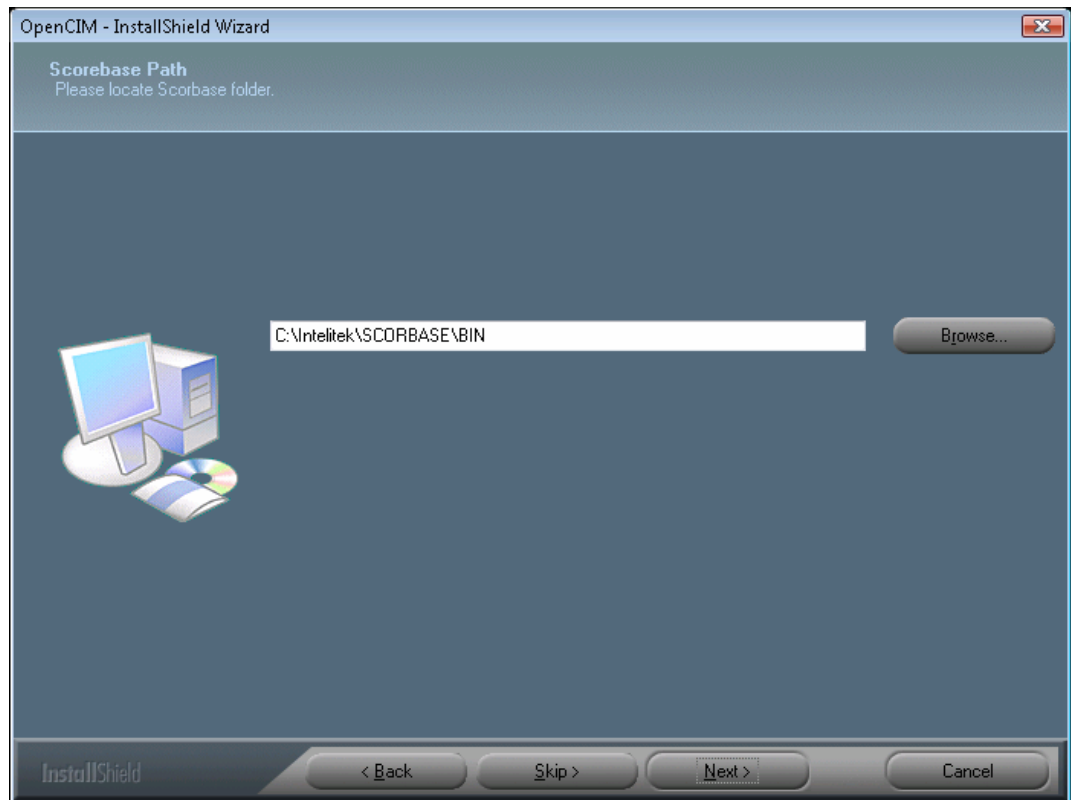
Figure 4-27: Scorebase Path Window

8. Browse to the location of Scorbase on your system and click **Next**. The Choose Destination Folder window will appear.

𝒢𝒰
Note   *If you have not yet installed Scorbase, click Skip to continue the installation process.*
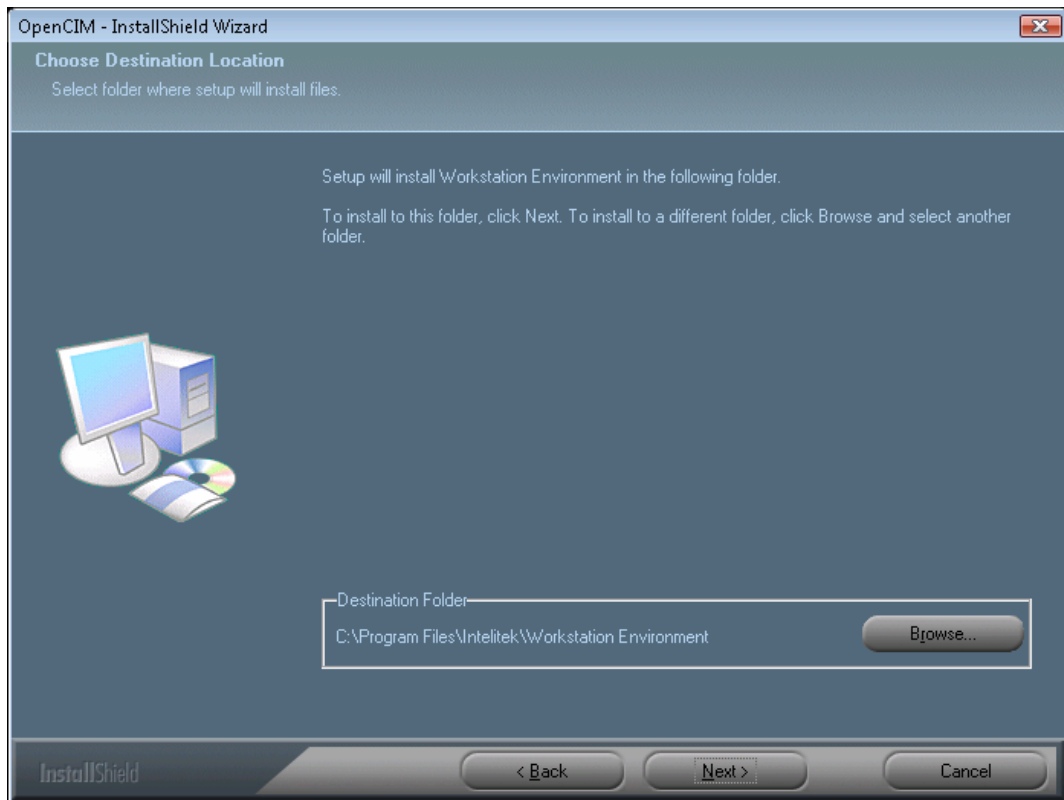
Figure 4-28: Choose Destination Location Window

9. Browse to the folder in which you want to install the Workstation Environment files and click **Next**. The installation will begin.

Shortcuts for the DDLoaders for the installed Workstations, and for the Remote Graphic Display if it was installed, will be added to the Desktop and to the Start menu.

Configuring Workstations

To use the OpenCIM modules on a station PC, ensure that you perform the following procedures:

- **On the CIM Manager and all Workstations,** configure the firewall for communication with other Workstations in the local network.

- **On all Workstations,** enable all workstations in the local network to access the Manager PC Drive in which the projects are installed by sharing the projects folder.

- Note: In order to work with the Remote Graphic Display with CIM custom parts displayed correctly on the Workstation, the following folder must be shared:

- In Windows XP: [default system drive (for example C:)]\Documents and Settings\All Users\Documents\Intelitek\OpenCIM\.

- In Windows Vista/7: [default system drive (for example C:)]\Users\Public\Documents\Intelitek\OpenCIM\.

Firewall Configuration

If the firewall of a Workstation is enabled, ports must be opened for them for the Workstation to be used in the local network. Open the Map.ini file to see which ports to open. For more details on this file, refer to Chapter 12: Inside OpenCIM.

In order for the Web Viewer to work, Port 80 must also be opened in the firewall.

Enable Access to CIM Manager

In Windows Vista/7, before you begin you must verify that the settings displayed in Figure 4-29 are on, to enable you to set the folders for sharing.
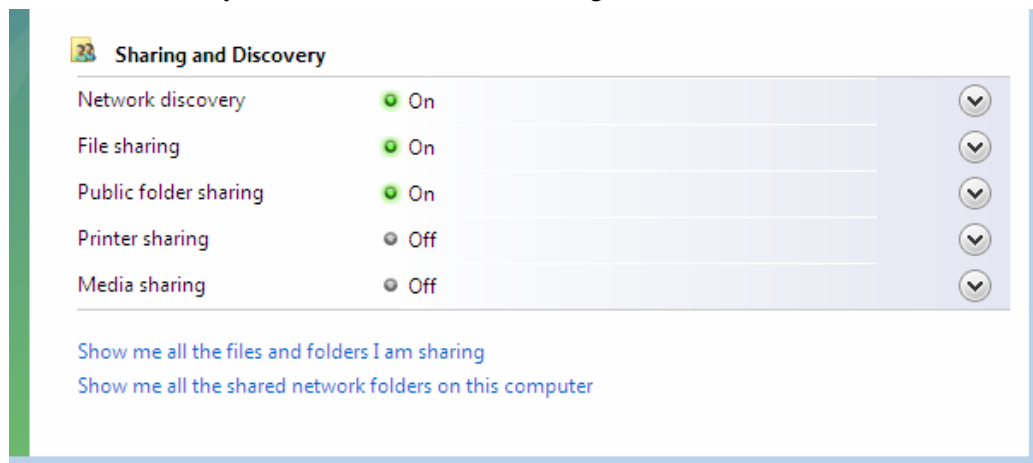
Figure 4-29: Sharing and Discovery Window

To verify that the correct options are on, follow these steps:

1. Click **Start | Control Panel | Network and Internet | Network and Sharing Center**.

2. In the Sharing and Discovery area, ensure that the first three items are on, as illustrated in Figure 4-29.

The projects are installed in the following locations:

- In Windows XP: [default system drive (for example C:)]\Documents and Settings\All Users\Documents\Intelitek\OpenCIM\Projects.

- In Windows Vista/7: [default system drive (for example C:)]\Users\Public\Documents\Intelitek\OpenCIM\Projects.

In Windows XP for example, to share folder X:

1. Select **My Computer**.

2. Find folder X

3. Right-click on the folder and select **Sharing And Security**. The folder's Properties window is displayed.

4. Select **Share this folder**.

5. Click **Permissions**. The folder's Permissions window is displayed.

6. Check the **Full control** option to allow other users to modify the files.

7. Click **OK**.

8. In the Properties window, click **OK**.

In Windows Vista/7:

1. Click **Start | Computer**.

2. Browse to and right-click the required folder.

3. Select **Properties** | **Sharing** | **Advanced Sharing** (if you are not the administrator you will need to enter the user name and password).

4. Click **Continue**.
5. Select **Share this folder**.
6. Confirm that the Share name is correct.
7. Click **Permissions**.
8. Check the **Full Control** option.
9. Click **OK.**
10. In the Advanced Sharing window click **OK.**
11. In the Properties window click **Close**.

On the Station PCs, map the drive of the Manager PC (where the projects are installed) through the network, as follows:

1. Select and right-click **My Computer**.
2. Select **Map Network Drive**.
3. Select a free drive and write the path for the Manager's projects directory.
    For example: Drive E: <path> \\Manager\Projects.
4. Check **Reconnect at Logon** and click **OK**.

On the Remote Graphic Display Station PC, map the drive of the Manager PC (where the OpenCIM shared directory is located) through the network, as follows:

1. Select and right-click **My Computer**.
2. Select **Map Network Drive**.
3. Select a free drive and write the path for the Manager's projects directory.
    For example: Drive E: <path> \\Manager\OpenCIM.
4. Check **Reconnect at Logon** and click **OK**.

Other Software

An OpenCIM system can include many devices which require additional software (e.g., Vision Systems, CNC machines, CAD/CAM, etc.).

Refer to the documentation and software installation instructions provided with each device.

ACL Controller Configuration

The following procedure is valid only for ACL controller types A and B. For other controllers, refer to the documentation and software installation instructions provided with each device.

At each station which contains an ACL controller, you will need to configure the controller and download the file ALL.CBU from the Station Manager PC to the ACL controller. This file contains all programs, positions and parameters required for controller operation in the OpenCIM environment.

To configure the controller from the ATS main screens:

1. Press **<Ctrl>+F1** to configure the controller.

2. Press **Y** to confirm the prompt to configure the controller. You are then prompted by a short series of Controller Configuration options. *Refer to the ATS Reference Guide provided with your ACL controller for complete instructions on configuring the controller.*

Once you have confirmed the configuration, ATS will perform the configuration procedure. You can ignore the message about the missing SETUP.PAR parameter file.

3. When the **>** prompt appears, press **[Shift] + F10**. The ATS Backup Manager screen opens.

4. Perform the following selections and entries:

```
Backup directory:
```

```
In Windows XP: [default system drive (for example
C:)]\Documents and Settings\All
Users\Documents\Intelitek\OpenCIM\Projects\microcim\ws1\rob
ot1
```

```
In Windows Vista/7: [default system drive (for example
C:)]\Users\Public\Documents\Intelitek\OpenCIM\Projects\micr
ocim\ws1\robot1.
```

Make sure the path correctly shows the working directory defined during the configuration, as shown in this example.

```
Backup / Restore: ALL
```

5. Use the arrow keys to highlight **ALL** and press **[Enter].**

```
During Restore: ERASE.
```

6. Use the arrow keys to highlight ERASE and press **[Enter].**

```
File name: all
```

7. Type **ALL** and press **[Enter]**.

8. Press **[Enter]** again. Press **F5** to RESTORE from disk.

9. Press **Y** to confirm all prompts to overwrite and erase.

Scorbase and Scorbase PRO Configuration

At each station which uses Scorbase, you will need to configure the software to match the robot and peripherals that the Controller will control.

To configure Scorbase, select **Options | Hardware Setup** from the Menu bar in Scorbase. Configure the settings in the Hardware Setup dialog box to match the robot and peripherals (and gripper if appropriate), that will be controlled by the Controller. Once you have

configured the Hardware Setup, click the **Search Home** icon to home the robot.

Note: Scorbase must be in On-line mode before configuring the Hardware Setup. To change to On-line mode, select **Options | On-line** from the Menu bar**.**

For more information on configuring Scorbase, refer to the Scorbase for SCORBOT-ER 4u and Scorbase for ER 9Pro/ER 14Pro User Manuals.

Robot Positions and Homing

The actual location of robot positions will differ depending on the actual stations, machines and devices included in the OpenCIM installation. You will therefore need to record (teach) new coordinates for the positions which were downloaded to the controller.

For further details on defining the positions, refer to the instructions supplied with your OpenCIM installation. In addition, refer to the User Manual that is provided with the robot or controller.

The procedures in this section describe how to home the robot from the ACL and USB controllers. For other controllers, refer to the documentation and software installation instructions provided with each device.

Homing the Robot from the ACL Controller

The following procedure describes how to home the robots from the ACL controller (types A and B). The teaching of robot positions is performed from the *ATS*. You must home the robot before you teach positions, as described in the following procedure.

To home robot from the ACL controller:

1. Enter the ACL command: `RUN HOMES`

2. Wait until the robot has completed the homing twice. All positions used in the OpenCIM belong to the vector CIM[n]. The size of the vector can vary from station to station. The standard size is n = 500. To teach the robot the positions required for the application, you must attach this vector to the teach pendant. Enter the ACL command: `ATTACH CIM`

3. When you have finished recording the positions, use the ATS Backup Manager to save the programs, positions and parameters to disk. *Save each item as a separate file*. Make the following selections and entries:

```
Backup directory:
```

In Windows XP: `[default system drive (for example C:)]\Documents and Settings\All Users\Documents\Intelitek\ OpenCIM\Projects\microcim\ws1\robot1`

In Windows Vista/7: `[default system drive (for example C:)]\Users\Public\Documents\Intelitek\OpenCIM\Projects\micr ocim\ws1\robot1.`

```
(for example)
Backup / Restore: ALL
File name: All
```

4. Press **[Enter]** again. Press **F3** to SAVE to disk

```
Backup / Restore: PROGRAMS
File name: programs
```

5. Press **[Enter]** again. Press **F3** to SAVE to disk.

```
Backup / Restore: POSITIONS
File name: position
```

6. Press **[Enter]** again. Press **F3** to SAVE to disk.

```
Backup / Restore: PARAMETERS
File name: parameter
```

7. Press **[Enter]** again. Press **F3** to SAVE to disk.

Homing the Robot from Scorbase

This procedure describes the basic instructions for homing the robot from the Scorbase software. For additional information on homing and recording the positions, refer to the Scorbase User Manual.

To home the robot from Scorbase:

1. Activate the Scorbase software, by double clicking the Scorbase SCORBASE icon on your desktop.

2. From the Scorbase Main Window select **Options | On-line** to activate on-line mode and click the icon to home the robot. Refer to the Scorbase User Guide for additional details on homing and recording the robot positions.

3. Select **File | Open Project** and navigate to the required project file for this station. For example:

   - *In Windows XP:*

     \\CIM-MANAGER\Documents\intelitek\OpenCIM\projects\M-CIM-SCBS\WS1\ROBOT1-program sample\Station1.WS.

   - *In Windows* Vista*/7*:

     \\CIM-MANAGER\Public\Documents\Intelitek\OpenCIM\Projects\M-CIM-SCBS\WS1\ROBOT1-program sample\Station1.WS.

Notes

*Select the **Reload last Project at Startup** from the **Options** menu, to load this project automatically the next time you load Scorbase.*

System Check

The system check depends on the actual stations, machines and devices included in the OpenCIM installation. Refer to documentation and instructions supplied with your OpenCIM installation.

Check the following hardware and device drivers and make sure they are functioning properly:

- Power Supply
- Conveyor
- Robots (for ACL controller use the ACL program DEBUG)
- PLC Device Driver
- Robotic Device Drivers
- CNC Device Driver
- Quality Control Device Drivers
- Pneumatic Device Driver
- Hydraulic Device Driver
- Process Device Driver

5

Project Manager

This chapter describes the Project Manager application that launches the Virtual CIM Setup and CIM Manager. It enables users to manage their own projects, and enables administrators to manage the projects in the archive. It includes the following sections:

- **Accessing the Project Manager**, describes how to access the Project Manager application.
- **Project Manager Main Window**, describes the main components of the Project Manager interface.
- **Project Manager User Mode,** describes the various project management tasks that can be performed by the user. These include, adding, importing and exporting projects and so on.
- **Project Manager Administration Mode,** describes the additional tasks performed by the administrators enabling them to manage the projects displayed in the archives.
- **Installing the Web Viewer Server**, describes how to install the Web Viewer Server from within the Project Manager application.

Accessing the Project Manager

After the OpenCIM installation is complete, the Project Manager icon is displayed on your windows desktop and Start menu.

To log in to the Project Manager application:

- From the windows Start menu, select **All Programs** | **OpenCIM** | **Project Manager** .

OR

- Click the OpenCIM Project Manager icon on your desktop.

The CIM Project Manager window appears displaying the **User Projects** tab, as shown in *Project Manager Main Window* in the following section.

Project Manager Main Window

The CIM Project Manager Main window is displayed, as follows:
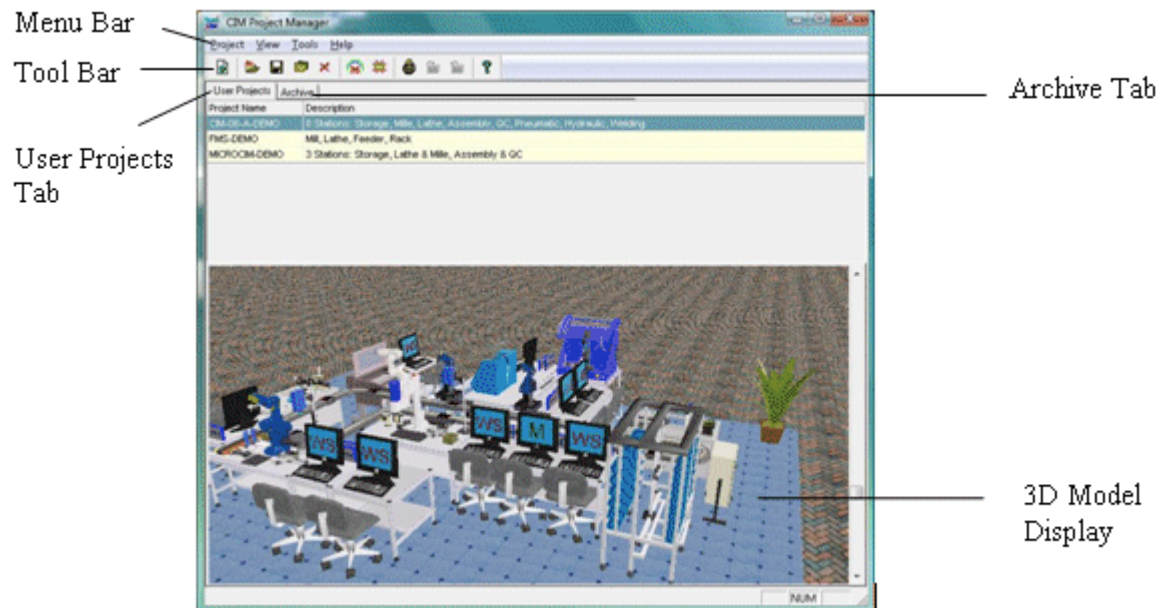
Figure 5-1: Project Manager Main Window

The CIM Project Manager window, shown above, contains the following elements, each of which is described in the sections that follow.

- Project Manager Menu Bar
- Project Manager Toolbar
- Archive Tab
- User Projects Tab
- 3D Model Display Area

Notes

*A 3D model of the selected project is shown below the list of project names and descriptions when **3D Model** is selected from the View Menu. See View Menu below.*

Project Manager Menu Bar

The Project Manager Menu bar contains four menus, each of which is described in detail in the sections that follow.

Project Menu

The following table contains a brief description of each option in the Project menu:

Option	Description
New	Enables you to add a new project to the list.
Import	Enables you to import an existing CIM Project from a specified directory. This option is enabled from the User Projects tab only.
Save As	Enables you to save the selected project under a new name. The new project is displayed in the list of projects in the User Projects tab.
Load All	Loads all the projects existing in the Projects directory into the User Projects tab. This feature is used, for example, when upgrading your OpenCIM, enabling you to load all previous projects that were created previously. This option is enabled from the User Projects tab only.
Back Up	Enables you to export a project to a specified directory for backup purposes. You can export projects from both the User Projects and Archive tab.
Delete	Removes a project permanently from the list as well as from your computer. You must have administrator access rights to remove projects from the Archive.
Exit	Exits the project manager application.

View Menu

The following table contains a brief description of each option in the View menu.

Option	Description
Toolbar	Shows or hides the OpenCIM toolbar.
3D Model	Shows or hides a 3D view of the selected project, in the 3D Model Display area.

Tools Menu

The following table contains a brief description of each option in the Tools menu.

Option	Description
CIM Manager	Activates the CIM Manager application, with the selected project from the User Projects list.
CIM Setup	Activates the CIM Setup application, with the selected project from the User Projects list.
Login as Administrator	Enables you to log in as an administrator.
Move to Archive	Moves the selected project from the User Projects tab to the Archive tab. Enabled for administrator's only.
Move to User Projects	Moves the selected project from the Archive tab to the User Projects tab. Enabled for administrators only.

Help Menu

The following table contains a brief description of each option in the Help menu.

Option	Description
Help	Displays the OpenCIM online help.
About	Displays the About Project Manager window contaning the current software version information.

Project Manager Toolbar

The toolbar icons provide quick access to the most commonly used functions in the CIM Project Manager window and appear as follows:

The following table contains a brief description of each option on the CIM Project Manager toolbar:

Option	Description
	New: Enables you to add a new project to the list.
	Import: Enables you to import an existing CIM Project from a local directory. This option is enabled from the User Projects tab only.
	Save As: Enables you to save the selected project under a new name. The new project is displayed in the list of projects in the User Projects tab.
	Back Up: Enables you to export a project to a specified directory for backup purposes. You can export projects from both the User Projects and Archive tabs.
	Delete: Removes a project permanently from the list as well as from your computer. You must have administrator access rights to remove projects from the Archive.
	CIM Manager: Activates the CIM Manager application, with the selected project from the User Projects list.
	CIM Setup: Activates the CIM Setup application, with the selected project from the User Projects list.
	Administrator: Enables you to log in as an administrator.
	Move to Archive: Moves the selected project from the User Projects tab to the Archive tab. Enabled for administrator's only.
	Move to User Projects: Moves the selected project from the Archive tab to the User Projects tab. Enabled for administrators only.
	Help: Displays the OpenCIM online help.

User Projects Tab

The User Projects tab appears automatically when you access the Project Manager application. The User Projects tab contains a list and description of the user's projects and enables you to perform the following:

- Create new CIM projects.
- Import projects from an external directory into the user projects list.
- Export projects from the user projects list to an external directory for backup purposes.
- Perform various editing options, such as, save as, delete, move to archive and more.
- Access the CIM Setup or CIM Manager applications.

The User Projects tab appears as follows:
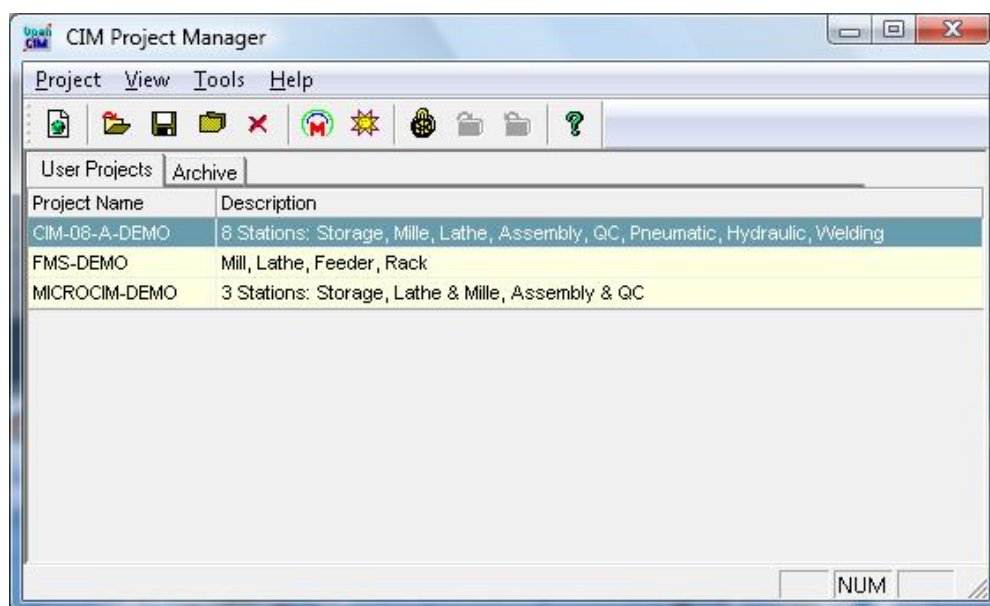
Figure 5-2: CIM Project Manager – User Projects Tab

3D Model Display Area

When View | 3D Model is selected from the menu bar, the selected project is displayed in a three dimensional view in the 3D model display area. You can zoom in and zoom out, rotate the image and adjust the viewing angle of the image, using the viewing options described in CIM Manager's graphic display module described Graphic Display and Tracking in Chapter 6, Operating CIM Manager.
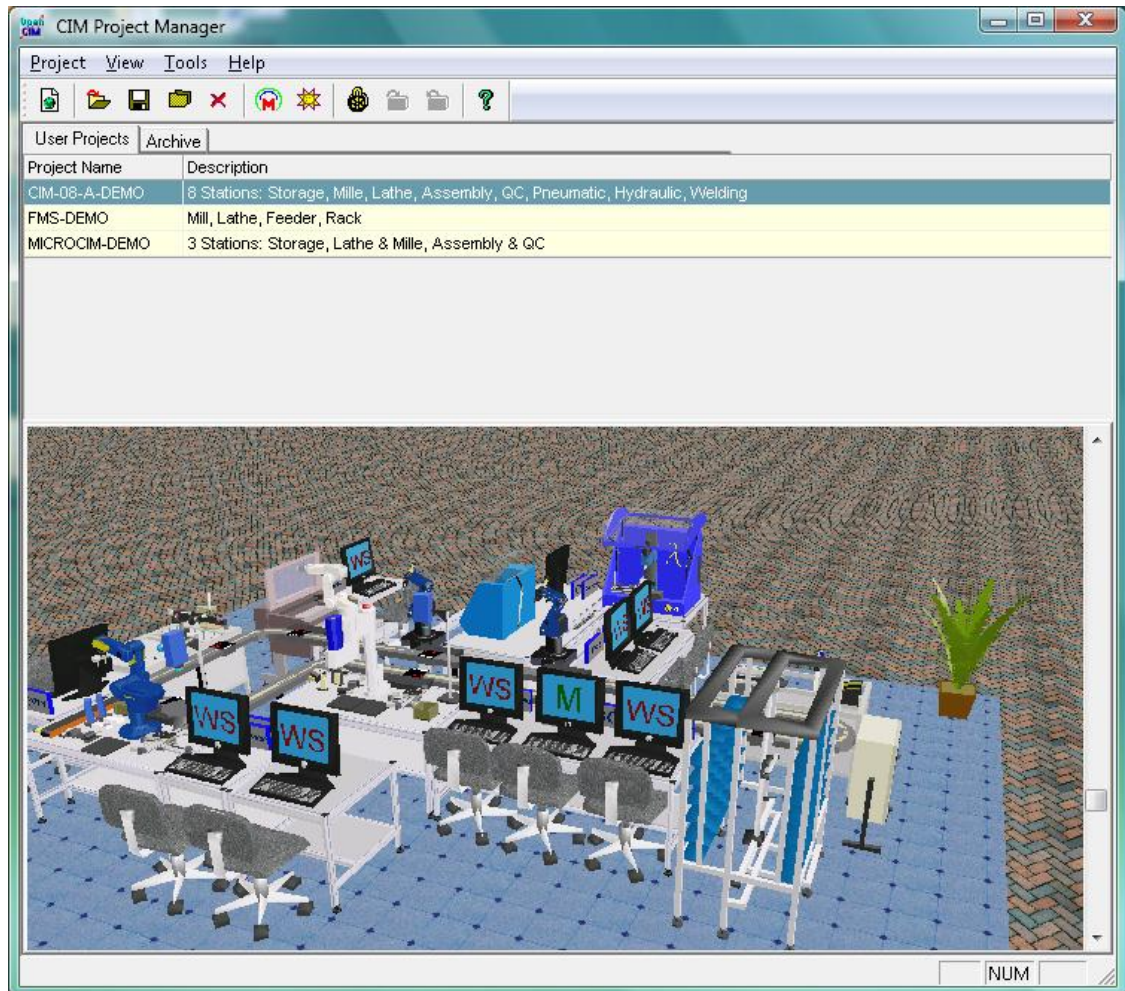
Figure 5-3: CIM Project Manager – User Projects Tab 3D Model View

User Projects Right Click Menu

The User Projects right-click menu provides alternate quick access to some of the most commonly used functions in the User Projects tab of the CIM Project Manager window. To access the menu, select a project and then right click on the project name. The menu appears as follows:

```
CIM-08-A-DEMO

Edit Description

CIM Manager
CIM Setup

Save As...
Back Up...
Delete...
```
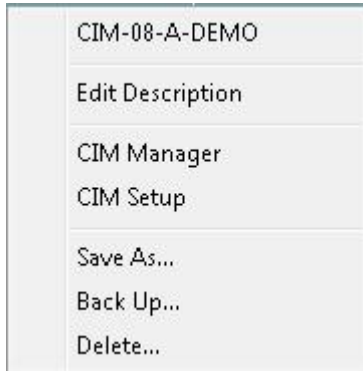
The options in the right-click menu shown above are described in the Project Manager Toolbar and Project Manager Main Window sections of this chapter.

Archive Tab

The Archive tab contains a list and description of all the predefined projects that are provided with the OpenCIM software and enables you to perform the following:

- Copy projects to the user projects list (users and administrators)
- Save projects to an external location for backup purposes. (users and administrators)
- Perform various editing options, such as delete, move to user projects and more (administrators only).

The Archive tab appears as follows:
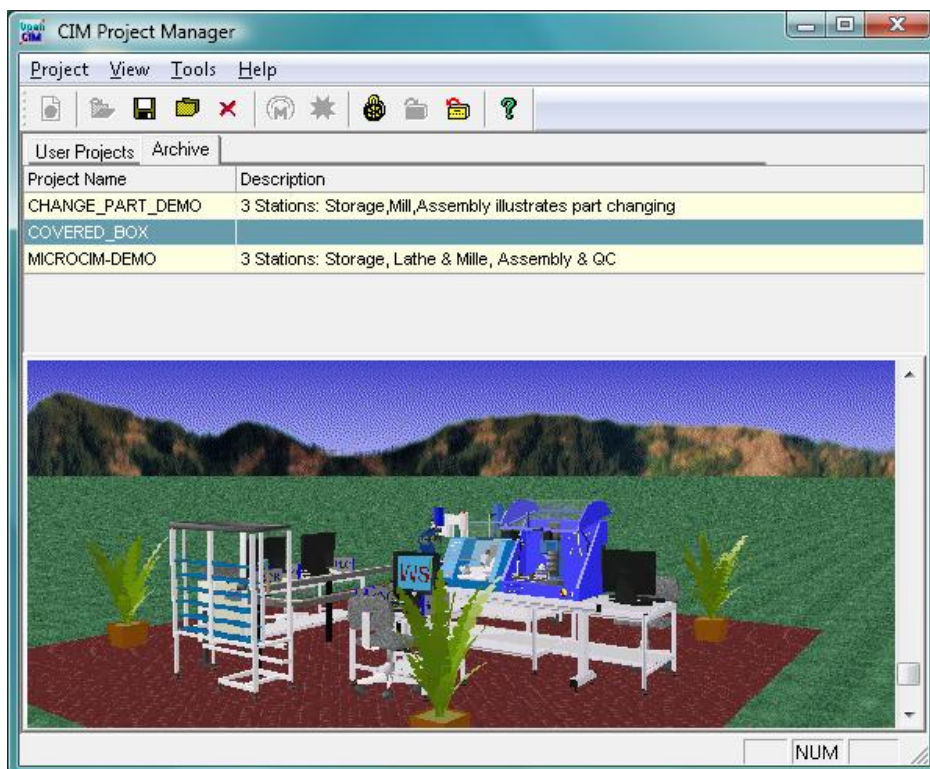
Figure 5-4: CIM Project Manager – Archive Tab

Archive Tab Right-Click Menu

The Archive Tab right-click menu provides alternate quick access to some of the most commonly used functions in the Archive tab of the CIM Project Manager window. To access the menu, select a project and then right click on the project name. The menu appears as follows:

The options in the right-click menu shown above are described in the Project Manager Toolbar and Project Manager Main Window sections of this chapter.

Project Manager User Mode

The Project Manager application enables users to manage their CIM projects from the User Projects tab. The tasks that the user can perform include adding projects, importing and exporting projects, copying projects and more.

Adding Projects

You can add new projects from the User Projects tab in the CIM Project Manager window. After you have created your project you can then activate the CIM Setup or CIM Manager applications for the selected project.

To add projects:

1. In the User Projects tab, click **New** [icon] on the toolbar. The Save As dialog box is displayed.

Figure 5-5: Save As Dialog Box

2. In the Project name field, enter the name of the new project, and click **OK**. The new project is added to the list of projects in the User Projects tab.

You can now select the new project in the User Projects tab and then select the required option, as follows:

- **CIM Setup** [icon]**:** Activates the Virtual CIM Setup application to enable you to edit your simulated CIM cell of the project. (For further details refer to *Chapter 7, OpenCIM Setup*).

- **CIM Manager** [icon]**:** Activates the CIM Manager application, enabling you to centrally control all the activities of the OpenCIM cell. (For further details refer to *Chapter 6: Operating CIM Manager*).

Importing Projects

The Project Manager enables users to import existing CIM projects into the User Projects tab from a specified directory.

To import projects:

1. From the User Projects tab, select **Import** [icon]. The standard Open dialog box is displayed. Navigate to select the file to import and click **Open**. The Open dialog box is displayed.

2. Navigate to the required directory and select the project (containing the ***.O2C** format) that you want to import.

3. Click **Open**. The selected project is displayed in the list of projects in the User Projects tab.

Exporting Projects

The Project Manager enables users to export an existing CIM Project from the User Projects or Archive tab to a specified directory, for future reference.

To export projects:

1. From the User Projects tab, select the project that you want to export.

2. Select **Backup** [icon] from the toolbar. The standard Save As dialog box is displayed. Navigate to select the directory to export and click **Save**. The project will be saved in the selected directory.

Loading All Projects

The Project Manager enables users to load all the projects that exist in the OpenCIM directory into the User Projects tab. This option is enabled from the User Projects tab only

To load all projects:

From the User Projects tab, select **Project | Load All**.   All the projects that exist in your Projects directory on your computer will be displayed in the User Projects tab. This feature is used for retrieval purposes and can be useful when upgrading your OpenCIM, enabling you to load all the previous projects created previously.

Copying Projects

The Project Manager enables users to create a copy of the selected project by saving it under a new name. The new project is then displayed in the list of projects in the User Projects tab.

To copy projects:

1. From the User Projects or Archive tab, select the project that you want to copy, and click **Save As** [icon]. The Save As dialog box is displayed:
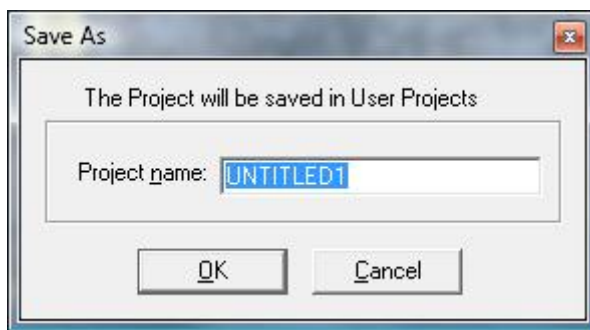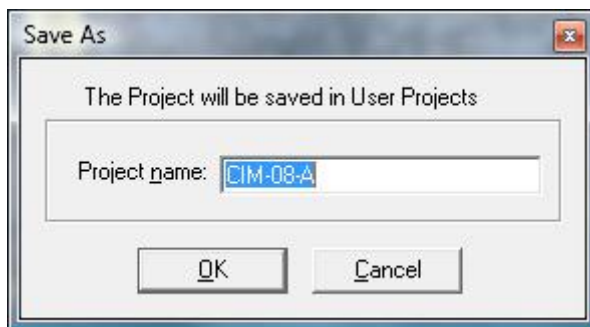

Figure 5-6: Save As Dialog Box

2. In the Project name field, enter the new project name and click **OK**.

3. The new project is added to the list of projects in the User Projects tab.

The Save As option always displays the new project to the User Projects tab (even when you select this option from the Archive tab).

Note

Removing Projects

When removing projects from the User Projects tab, you must act cautiously since the project is removed permanently from your computer. Only administrators can remove projects from the Archive tab.

To remove projects:

1. From the User Projects tab, select the project that you want to remove.

2. Click [×] **Delete**. The selected project is removed from the User Projects list.

Accessing CIM Manager

The CIM Manager application enables users to centrally control all the activities of the selected CIM cell. For further details refer to Chapter 6, Operating CIM Manager.

To access the CIM Manager:

1. From the User Projects tab, select the project for which you want to access the CIM Manager.

2. Select [M] **CIM Manager** to activate the CIM Manager application, enabling you to control all the activities of the OpenCIM cell.
   (For further details refer to *Chapter 6, Operating CIM Manager).*

Accessing CIM Setup

The CIM Setup application enables users to create and modify the CIM Setup of the selected cell. For further details refer to Chapter 8, OpenCIM Setup.

To access the CIM Setup:

1. From the User Projects tab, select the project for which you want to access the CIM Setup.

2. Select [✦] **CIM Setup** to activate the CIM Setup application, enabling you to edit your simulated CIM cell of the selected project.
   (For further details refer to *Chapter 8, OpenCIM Setup).*

Project Manager Administration Mode

In addition to all the tasks described in the Project Manager User Mode, the Project Manager application also enables administrators to manage the projects displayed in the archives. These include, moving projects from the user projects list to the archives and from the archives to the user list and more.

Defining Administrator Access Rights

After entering your administrators' password, (described below) you are granted additional access rights to the system, enabling you to add, remove, import and export projects from the Archive tab and more. You must enter this administrators' password each time you enter the Project Manager application.

To log in as an administrator:

1. Click [icon] on the toolbar. The Administrator dialog box is displayed.
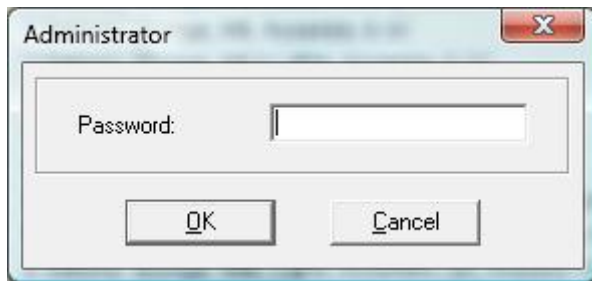
Figure 5-7: Administrator Dialog Box

2. In the Password field, enter **mypassword** and click **OK**. You now have administrator's access rights.

Moving Projects to Archive

Administrators can move projects from the User Projects to the Archive. This is used for example when you want to define a project as read-only.

To move projects to the Archive:

From the User Projects tab, select the project to move to the Archive tab and

click [icon] **Move to Archive**. The selected project is removed from the User Projects tab and displayed in the Archive tab.

Moving Projects to User Projects tab

Administrators can move projects from the Archive tab to the User Projects tab. This is used for example when a project is no longer required in the archive, but is required for the users.

To move projects to the user projects tab:

From the Archive tab, select the project to move to the User Projects tab and

click [icon] **Move to User Projects**. The selected project is removed from the Archive tab and is displayed in the User Projects tab.

Installing the Web Viewer Server

1. To install the Web Viewer Server select **Tools | Web Viewer Server Setup** from the Menu bar. Click **Yes** when prompted to begin the installation and configuration of the Web Viewer Server.

2. For the Web Server to be installed in XP, the Windows Installation CD must be inside the computer.

3. In Windows Vista/7, click **Continue** when prompted to provide administrator details (if you are not logged as administrator to the computer) or to confirm installation of the Web Viewer Server.

   For more information on installing the Web Viewer Server, refer to Chapter 4: Software for Workstation PCs.

6

Operating CIM Manager

This chapter describes how to operate the CIM Manager which is used for operating the OpenCIM system and controlling production. The CIM Manager application is accessed from the Project Manager, as described in Chapter 5, CIM Project Manager. It includes the following sections:

- **Accessing the CIM Manager**, describes how to access the CIM Manager application.
- **CIM Manager Main Window**, describes the main components of the CIM Manager interface.
- **CIM Operational Modes**, describes the operational modes of the CIM Manager application.
- **CIM Manager Views**, describes the various view screens in the CIM Manager application for tracking the production process.
- **CIM Scheduler**, introduces the CIM scheduler, enabling you to view various production schedules.
- **Graphic Display and Tracking**, introduces the Graphic Display module displaying real-time, 3D animations of the operations being performed in the CIM cell.

Accessing the CIM Manager

The CIM Manager is accessed from the Project Manager main window enabling the user to centrally control all the activities of a selected OpenCIM cell.

To access the CIM Manager application:

From the Project Manager Main window, shown in Chapter 5: Project Manager, click CIM Manager on the toolbar. The CIM Manager Main window is displayed, as shown in CIM Manager Main Window in the following section.

CIM Manager Main Window

The CIM Manager Main window appears, as follows:
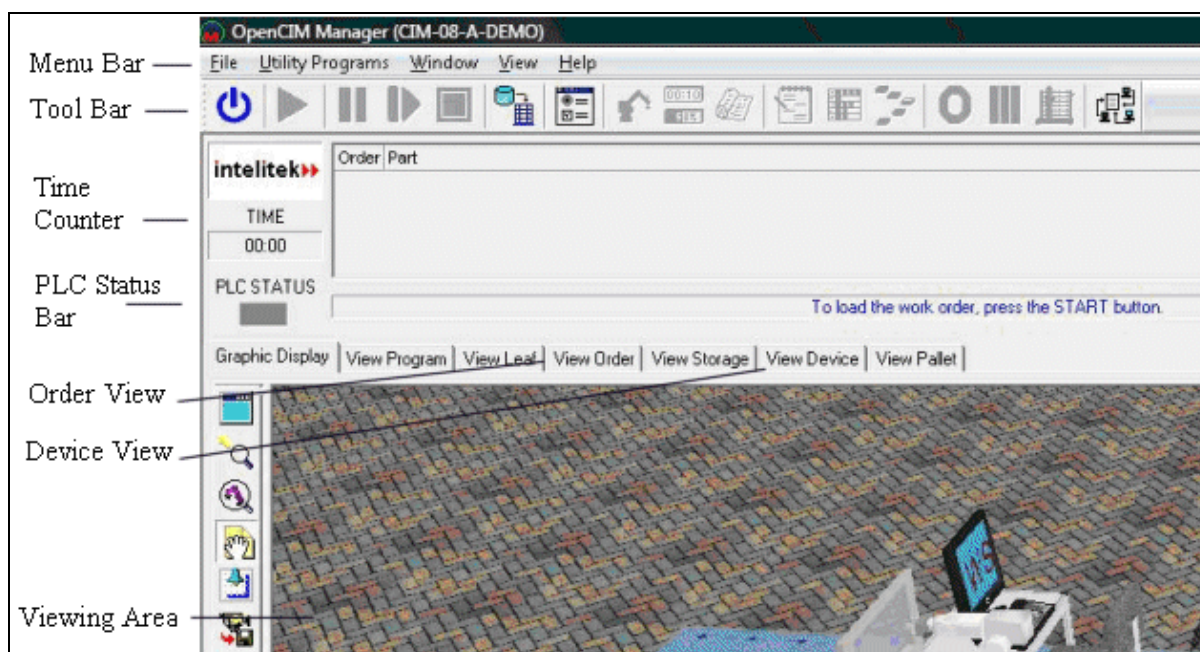
Figure 6-1: CIM Manager Main Window

The CIM Project Manager window, shown above, contains the following elements, each of which is described in the sections that follow.

- CIM Manager Menu Bar
- CIM Manager Toolbar
- Time Counter
- PLC Status Bar
- Viewing Area
- Order View
- Device View
- Status Bar

CIM Manager Menu Bar

The CIM Manager Menu bar contains five menus, each of which is described in detail in the sections that follow.

File Menu

The following table contains a brief description of each option in the File menu:

Option	Description
Default Storage	Restores a predefined configuration of the storage from the backup database file.
Start	Loads the production work order (A-Plan). Opens communication channel. This sends a command to reset (INIT) all device drivers. The run arrow turns blue and the stop button turns red, indicating that they are available for use. The production plan will appear in the Program View screen.

Option	Description
Run	Starts executing the A-Plan. CIM production begins. The pause button turns blue, indicating that it is available for use.
Continue	Resumes operation after production has been paused.
Stop	Stops production. It can be used as emergency button.
Modes	Displays the Modes dialog box, as described in *Modes Dialog Box*.
Database Tool	Displays the CIM Database browser. Recommended only for advanced users.
Exit	Exits the OpenCIM Manager application.

Utility Programs

The following table contains a brief description of each option in the Utility Programs menu:

Option	Description
MRP	Displays the CIM MRP window, enabling you to create a list of customers, define the products ordered by each customer, and generate a manufacturing order.
Machine Definition	Displays the CIM Machine Definition window enabling you to define the machines and the specific processes that the machines will perform.
Optimization Definition	Displays the CIM Optimization Manager enabling users to select machine queue algorithms and define their weight.
Part Definition	Displays the CIM Part Definition window, enabling you to define the parts that the CIM cell can manufature, including available parts and the parts that need to be manufactured. These include: Supplied Parts, Product Parts and Phantom Parts.
Performance Analysis	Displays the CIM Performance Manager for viewing and analyzing information generated form the manufacturing cycle.
Report Generator	Displays the CIM Report Part Definition window, enabling you to generate and print various reports from the database. These include, part definition reports, machine definiton reports and more.
Scheduler Gantt	Displays the CIM Scheduler window, enabling you to plan, coordinate and track variuos production schedules. For further details refer to *CIM Scheduler*.
Storage Manager	Displays the CIM Storage Manager window which manages and keeps track of parts in storage and informs the system of the part location.

Window

The following table contains a brief description of each option in the Windows menu:

Option	Description
Device	Displays the Device View window containing a lists of all the devices in the CIM cell.
Event	Displays the Event View window containing a list of events that will be generated by OpenCIM simulation engine when **Run** is activated.
Leaf	Displays the Leaf View window containing the production activities occurring in the CIM cell.

Option	Description
Log	Displays the Log View window contains a list of all the messages that have been sent and received by the CIM Manager.
Machine Queue	Displays the parts that are currently in the queue to the various machines for processing.
Order	Displays the Order View window containing the current manufacturing order information.
Pallet	Displays the Pallet View window containing a list of the pallets in the CIM cell and each pallet's current status.
Program	Displays the Program View window, containing the A-Plan (the production work order) of the CIM cell.
Storage	Displays the Storage View window, containing the current location of parts in the CIM cell.
Message History	Displays the Message History window, containing three types of messages: External Messages, Internal Messages, CIM Warnings.

View

The following table contains a brief description of each option in the View menu:

Option	Description
Toolbar	Shows or hides the CIM Manager toolbar.

Help

The following table contains a brief description of each option in the Help menu:

Option	Description
About	Displays the About OpenCIM Manager window contaning the current software version information.
Help	Displays the OpenCIM Online help.
Registration	Displays the registration dialog box enabking you to perform various registration options, such as obtain your software license from Intelitek's website, or by e-mail, fax or phone.

CIM Manager Toolbar

The toolbar icons provide quick access to the most commonly used functions in the CIM Manager window and appear as follows:

The following table contains a brief description of each option on the CIM Manager toolbar:

Option	Description
	Start: Loads the production work order (A-Plan). Opens communication channel. This sends a command to reset (INIT) all device drivers. The run arrow turns blue, and the stop button turns red, indicating that they are available for use. The production plan will appear in the Program View screen.

Option	Description

Figure 6-2: Production Control Bar

Option	Description
▶	**Run**: Starts executing the A-Plan. CIM production begins. The pause button turns blue, indicating that it is available for use.
❚❚	**Pause**: Halts operation at any time; causes the CIM Manager to stop sending commands to the device drivers and then wait until the Continue button (which has turned red) is pressed. All device drivers complete the current command.

🔖 *Note*

*CIM devices do not stop immediately when you click the **Pause** button. Each device will complete its current operation before it stops.*

❚▶	**Continue**:Resumes operation after production has been paused.
■	**Stop:** Stops production. It can be used as an emergency button.
	Reset Storage: Restores a predefined configuration of the storage from the backup database file.
	CIM Modes: Displays the Modes dialog box, as described in *Modes Dialog Box*.
	Device View: Displays the Device View window containing information referring to all the robots and machines in the CIM cell.
	Event View: Displays the Event View dialog box that lists the events generated by the OpenCIM simulation engine.
	Leaf View: Displays the Leaf View window, providing a detailed description of the production activities of the CIM cell.
	Log View: Displays the Log View window that contains a log of all the messages that have sent and recieved by the CIM Manager.
	Pallet View: Displays the Pallet View window, containing the pallet information in the CIM cell, such as descriptions, status and more.
	Machine Queue View: Displays the Machine Queue View showing the parts that are currently in the queue to the various machines for processing.
	Program View: Displays the Program View window, that contains a copy of the A-Plan or production work order.
	Storage View: Displays the Storage View window containing every location defined in the CIM system.

Option	Description
	Message History: Displays the Message History dialog box with three types of messages: External and Internal Messages and CIM Warnings.

CIM Manager Time Counter

The CIM time counter indicates the time elapsed since the onset of the production cycle.

PLC Status Bar

The PLC Status Bar, informs us whether or not the connection is active.

Viewing Area

The Viewing area enables you to monitor various aspects of the production cycle on a real-time basis by selecting one of seven tab views. By default, the Graphic Display tab is selected and the viewing area displays 3D graphic simulation of the CIM production cycle. For further details on the tabs displayed in the Viewing area, refer to Graphic Display and Tracking.

Order View

The Order View, located below the toolbar in the left portion of the window, displays data regarding the order of parts and their production status.

Device View

The Device View, located below the toolbar in the right portion of the window, displays data regarding the activity taking place in the devices during the production process.

Status Bar

The application's status bar, located at the bottom of the window, displays the status and location of the application, such as the current operation mode and the location of the WSO.ini file used by the manager.

Information Bar

The Information Bar displays general messages that occur during production, such as Order is in progress and so on.

CIM Operational Modes

The CIM Manager can operate in the following modes:

- **Simulation Mode:** The CIM Manager does not communicate with device drivers. This mode does not require either hardware or device drivers.
- **Real Mode:** The CIM Manager communicates with all device drivers, whether or not hardware is in use. This mode requires that *all device drivers which are needed for a specific application (for a specific product order)* be loaded, so that the CIM Manager can transmit and receive messages.

Since the CIM Manager affects operation of the CIM cell hardware by communicating with the device drivers (and not directly with the hardware), the CIM Manager can operate in *real mode* even if the hardware has not been activated, or even if no hardware exists.

The CIM modes are described in the following table:

CIM Manager Mode of Operation	Device Driver	Hardware
Simulation	Not required.	Not required
Real Mode	All device drivers must be loaded.	Not required. Hardware may be activated, or it may be simulated by the device drivers, at some or all stations.

CIM Modes Dialog Box

The MODES Dialog Box is displayed by clicking the [icon] CIM Modes icon on the toolbar. This dialog box enables you to define the current modes that are active in the CIM Manager, such as, whether the CIM Manager is working in Real Mode or Simulation Mode, whether the Web Viewer is activated and so on. The modes defined in the MODES dialog box are per project.
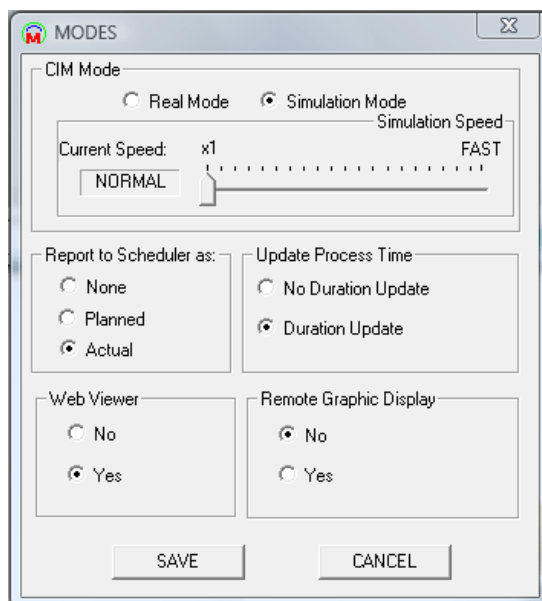
Figure 6-3: Modes Dialog Box

CIM Mode	**Real Mode:** In this mode there is message interchange between Manager and Device Drivers.
	`Simulation Mode:` The production runs on the Manager. There is no message interchange between any devices. You can set the production speed for the simulation, where 1 is the slowest and 100 is the fastest.
Report to Scheduler as	**None**: In this mode, the CIM Manager does not send messages to the CIM Scheduler.
	Planned: In this mode, the CIM Manager sends messages as planned to the CIM Scheduler (generally, when CIM Manager is operating in Simluation Mode).
	Actual: In this mode, the CIM Manager sends actual messages to the CIM Scheduler (generally, when CIM Manager is operating in Real Mode).
Update Process Time	**No Duration update:**Does not update the duration of any process defined in Machine Definition.
	Duration Update: Updates the duration of any process defined in Machine Definition. The duration is the actual time that a machine has taken to complete a process.
Web Viewer	Specifies whether or not the web viewer is activated. Select the required option, as follows:

- Yes: Activates the Web Viewer.
- No: Disables the Web Viewer.

For information on installing the web viewer server refer to chapter 4.

For information on installing the web viewer client refer to chapter 10.

Remote Graphic Display	Specifies whether or not status messages are sent from the devices in operation to the Graphic Display module so that the display is updated accordingly.

> Yes - Messages are sent to the Remote Graphic Display.
>
> No - Messages are not sent to the Remote Graphic Display.
>
> By default the Manager is configured to support only one Remote Graphic Display.
>
> To enable support for more than one Remote Graphic Display:

1. Open Notepad by clicking **Start | Accessories | Notepad**.
2. Select **File | Open** and browse to [default system drive (for example C:)]

> - Windows XP:
>
> \Documents and Settings\All Users\Documents\Intelitek\OpenCIM\Projects\<project name>\setup\opencim.ini.
>
> - Windows Vista/7:
>
> \Users\public\Documents\Intelitek\OpenCIM\Projects\ <project name>\setup\opencim.ini.

3. Find the NUMCIMULSOCKETS in the [networking] key and change it to the required number of cim simulations.
4. Click **Save.**

✍ Note

If the Graphic Display is not activated on any PC, click **No,** otherwise the manager will work very slowly.

Refer to Chapter 4: Software for Workstation PCs for information on installing the Remote Graphic Display.

Once installed, the Remote Graphic Display can be launched via

the shortcut that is added to the Desktop.

To begin producing an order, do the following:

❶
❷
❸
Procedure
Starting Production

1. Start all OpenCIM device drivers by clicking on the **Device Driver Loader** icon at each Station Manager PC. (Skip this step if you intend to work in Simulation mode.)
2. Select either Real Mode or Simulation Mode from the Modes dialog box.
3. Reset the Storage by clicking the Refresh Storage icon.
4. In the CIM Manager, click the **Blue** button.
5. Click the **Run** button to start executing the production plan.

⚠ Warning!

For safety reasons, when operating the CIM in Real Mode, you must use the actual hardware EMERGENCY buttons to halt the system in an emergency.

Working in Simulation Mode

To operate the CIM cell in simulation mode, you must verify that the simulation mode is selected in the CIM Manager.

To operate the CIM cell in simulation mode:

1. From the Project Manager application, select the required project and click ![icon] **CIM Manager**. The CIM Manager main window is displayed, as shown in *CIM Manager Main Window*.

2. Select the **Modes** ![icon] icon on the toolbar. The MODES dialog box is displayed.

3. Select **Simulation Mode** option, and if required adjust the simulated speed and define additional options, as described in the *CIM Modes Dialog Box*, and click **Save.**

4. Verify that the Graphic Display tab is selected and click the **Reset Storage** ![icon] button on the CIM Manager toolbar.

5. If required, you can select **Utility Programs | Scheduler Gantt** to view the production schedule.

6. Click the **Start** ![icon] button to start, and click **Run** ![icon]. The selected CIM Cell is now running in simulation mode.

Working in Real Mode

To operate the CIM cell in real mode, you must verify that the real mode is selected in the CIM Manager.

![warning icon] *Before starting actual production, make sure you are in compliance with all the safety measures detailed in Chapter 3, Safety.*

Warning!

To operate the CIM Cell in real mode:

1. Remove any templates on the conveyor and at station buffers.
2. Remove any parts left at stations: in a robot's gripper, in a machine and on storage racks.
3. Load parts into the ASRS and into any feeders.
4. Turn on all hardware: PCs, controllers, CNC machines, etc.
5. Make sure all PCs have been activated.
6. From the PC of each Station Manager click the **Loader WS1** icon (for example). The CIM Device Driver Loader window is displayed.
7. In the **Simulation** column, select the mode in which you want to load the device drivers by selecting or deselecting the column and click the **Start** ![icon] button.
8. At each station, home the robot and initialize all the equipment.
9. On the PC that contains the CIM Manager, perform the following:

1. From the Project Manager application, select the required project and click ![CIM Manager icon] **CIM Manager**. The CIM Manager main window is displayed, as shown in *CIM Manager Main Window*.

2. Select the **Modes** ![Modes icon] icon on the toolbar. The MODES dialog box is displayed.

3. Select the **Real Mode** option, and if required define additional options, as described in the *CIM Modes Dialog Box*, and then click **Save**.

4. If required, you can select **Utility Programs | Scheduler Gantt** to view the production schedule.

5. Click the **Start** ![Start button] button to start, and click **Run** ![Run button]. The selected CIM Cell is now running in real mode.

Note

> *You can turn on OpenCIM workstation PCs and hardware in any order. There is no mandatory boot-up sequence. You can also reboot a PC as long as it is not in the middle of an operation or communicating with the CIM Manager. If you reset a PC, you do not need to reset other workstation PCs connected to the OpenCIM network. When the PC boots up, its applications will resume communication with other PCs on the OpenCIM network.*

CIM Manager Views
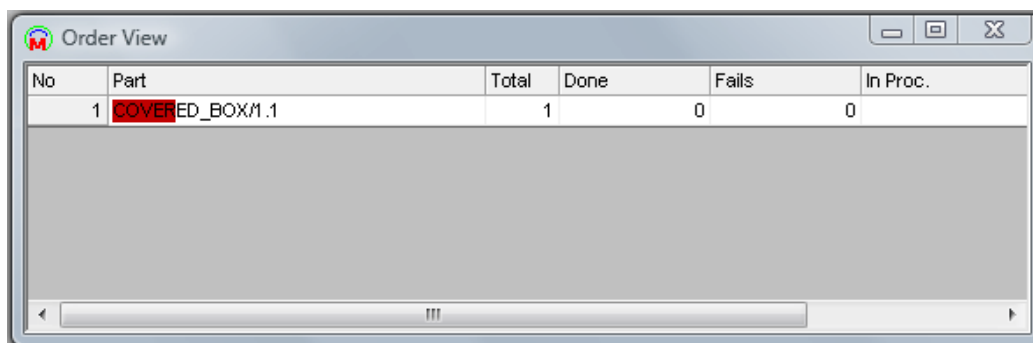
During the manufacturing process, you can track production by looking at up to eight different view screens, as follows:

- Program View
- Order View
- Storage View
- Device View
- Log View
- Machine Queue View
- Pallet View
- Leaf View (only after Run is pressed)
- Event View (only after Run is pressed)
- Message History

Click the appropriate icon on the toolbar to open the desired View screen, or select the desired View from the alphabetical list in the Windows drop-down menu. Alternatively you can replace the Graphic Display in the lower half of the Manager with the desired View screen (with the exception of Log and Event) by clicking the appropriate tab.

Order View

The Order View is a copy of the Manufacturing Order. It is the most basic of the available views.
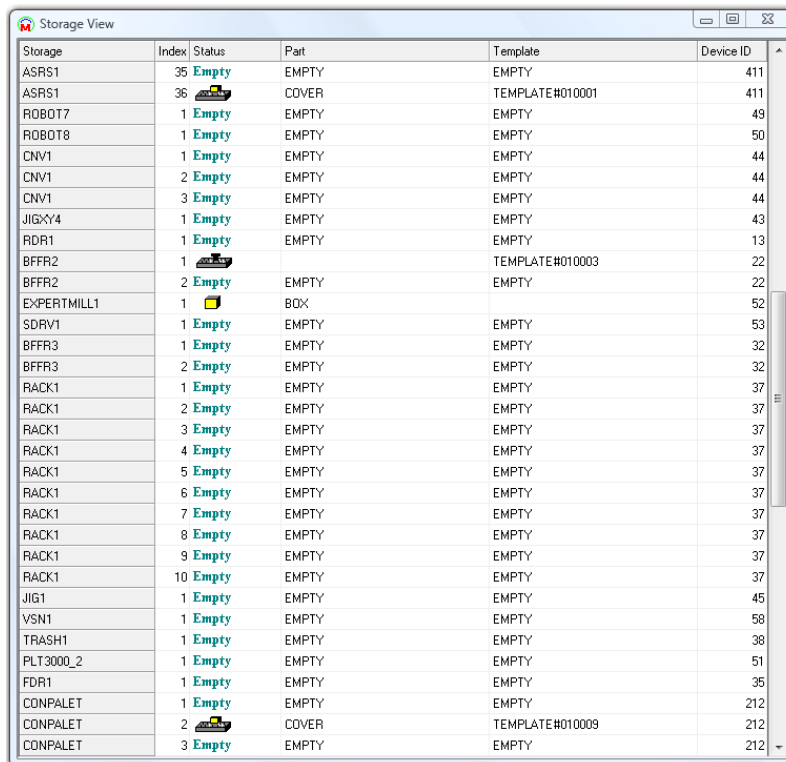
Figure 6-4: Order View

The following is an explanation of each column in the Order View.

No.	Line number from Manufacturing Order.
Part	Name of part; as defined in Part Definition Form used in Manufacturing Order. The grid containing the part name is progressively filled in red from left to right, indicating the production status of the part which is being manufactured.
Total	Total number of parts to be produced, as defined in Manufacturing Order.
Done	Number of parts that have been completed. Updated during production.
Fails	Number of parts that have failed inspection. Updated during production.
In Process	Number of parts that are being manufactured. Updated during production.

Storage View

The Storage View resembles the Location Status Report (see Chapter 5). This view is a detailed listing of every location defined in the CIM system.

Figure 6-5: Storage View

The following is an explanation of each column in the Storage View.

Storage
: A list of all the locations in the CIM cell.

Index
: Indicates the exact location on a device which has more than one location for a part. For example, the conveyor (CVN1) has three indices, one for each station; the robot identified as ROBOT7 has only one index; the ASRS1 has an index for each of its cells.

Status
: Graphic illustration of the contents of the location, as defined in the PART and TEMPLATE columns. For example, EXPERTMILL1 has a part named BOX.

PART
: Status of the specified location: either Empty or the Name of the part if it exists at the location.

TEMPLATE
: Status of the specified location: either Empty or the ID of the template if it exists at the location.

Device ID
: Device ID number defined in the Virtual CIM Setup (or assigned by the CIM Manager during production).

Program View

The Program View is a copy of the A-Plan, or production work order. You can track the current status of production by watching the Program View. This screen shows the commands that the CIM Manager executes to produce an order. These commands are executed in bottom-up order.

Figure 6-6: Program View

The following is an explanation of each column in the Program View screen.

Level	This hierarchy number indicates the level in the Part Definition tree for each ordered product. Operations at the same level can occur in parallel (except an ONFAIL process).
Part	Unique name used to identify the subpart currently under production.
Action	The A-Plan command or user-defined process that the CIM Manager executes to produce a part.
Subpart	The part or object which the A-Plan action operates on.
Target	The destination where this subpart is to be delivered.
Index (#)	Parameters used by this command or process.
P1	Shows the current production status. The number of shaded Part columns corresponds to the total number of parts ordered.

When a part is being produced, one of the following symbols appears at the current stage of production:

↵	Command sent, waiting for acknowledgment.
ON	Device has begun processing this part (device driver has responded with Start message).
OFF	Device finished processing this part (device driver has responded with Finish message).
■	The blue box indicates operation completed (device driver has responded with End message).
WAIT	CIM Manager is waiting for another operation to complete before sending this command.

Device View

The Device View is a complete list of every robot and machine (including QC devices) in the CIM cell and a description of the current action being performed by each.

Figure 6-7: Device View

The following is an explanation of each column in the Device View.

Device
: Name of the device or machine, as defined in the Virtual CIM Setup.

Status
: When a part is being produced, one of the following symbols appears at the current stage of production:

RUN Command sent, waiting for acknowledgment.

Start Device has begun processing this part (device driver has responded with Start message).

Finish Device finished processing this part (device driver has responded with Finish message).

End Device ended processing this part (device driver has responded with End message).

Stop Device is ready for next command.

Load Device is loading the processing program from the Backup or the Database.

Action
: The movement or operation command which is currently being executed by the device. For robots, the action is commonly the placement of a part. For machines, the action is usually the name of the process (as defined in the Machine Definition form).

Station
: The number which identifies the workstation where the device is located.

ID
: The Device ID number, as defined in the Virtual CIM Setup.

Log View

The Log View is a transcript of the Leaf View. It details all messages which have been sent and received by the CIM Manager.
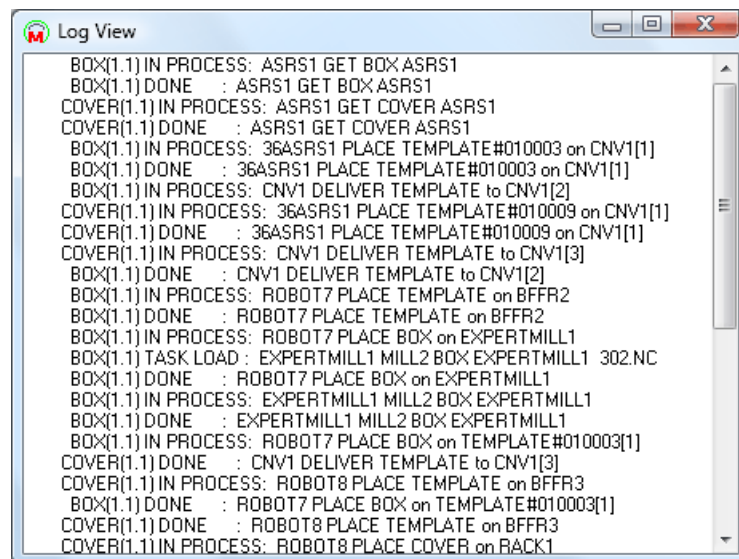
Figure 6-8: Log View

You can control the amount of information that is displayed by editing the CIM Manager INI file. By default, the system is set to display only IN PROCESS and DONE messages, which allow you to see which commands have been sent and which have been executed.

Machine Queue View

The Machine Queue View displays the parts that are currently in the queue to the various machines for processing.
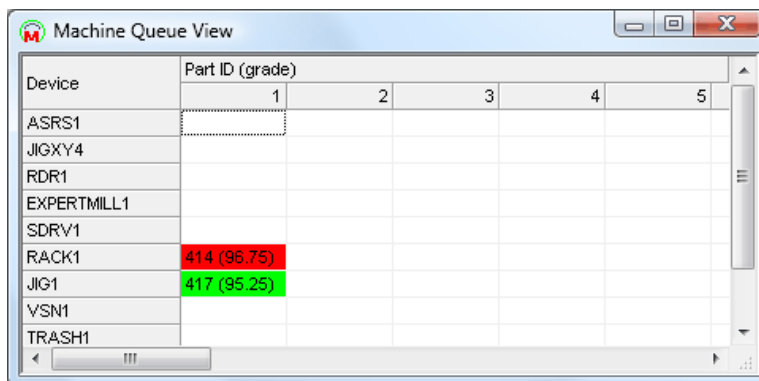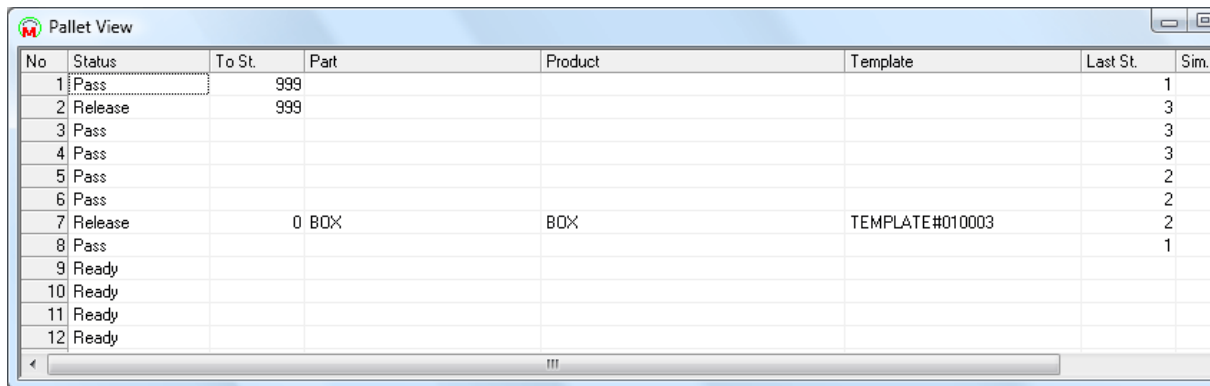
Figure 6-9: Machine Queue View

The following is an explanation of the columns in the Machine Queue View:

Device.	Lists the devices in the CIM cell.
Part ID (grade)	Lists the ID and grade value of the color coded parts that are in the queue to the specific device

Pallet View

The Pallet View is a complete list of every pallet in the CIM cell and a description of its current status.

Figure 6-10: Pallet View

The following is an explanation of each column in the Pallet View.

No.	Identification number of the pallet
Status	Describes the status of:
	Ready Pallet has not yet reached a station.
	Pass Pallet is moving; has passed through the last station.
	Stop Pallet has been stopped at a station to be unloaded.
	Stop[Free] Pallet has been stopped at a station to be loaded.
	Released Pallet has been released from a station.
To Station (To St)	Number of the next workstation which pallet will reach. If pallet status is Free, the destination is Station 999.
Part	Name of part or subpart being carried by pallet.
Product	Name of final product to which part belongs.
Template	Identification number of the template being carried by the pallet.
Last Station (Last St)	Number of the last workstation which pallet has passed through.
Sim Place	"Simulated position"; a sector location on the conveyor, as used in the simulated graphic display.

Leaf View

The Leaf View provides a detailed description of the production activities of the CIM cell, describing the current operation being performed on each item and the operation that will immediately follow.

Figure 6-11: Leaf View

The following is an explanation of each column in the Leaf View.

Subpart of Part | Name of the part and the name of the final product to which it belongs.

Action
>Next Process | The action currently in progress (upper line) and the next process to be performed on the part. For example:
MILL2 = process defined in the Machine Definition form
BOX = part name.
EXPERTMILL1 = name of machine which will perform operation, as defined in the Machine Definition form.

Status | When a part is being produced, one of the following symbols appears at the current stage of production:

↵ Command sent, waiting for acknowledgment.

ON Device has begun processing this part (device driver has responded with Start message).

OFF Device finished processing this part (device driver has responded with Finish message).

■ The blue box indicates operation completed (device driver has responded with End message).

WAIT CIM Manager is waiting for another operation to complete before sending this command.

Part ID | An internal ID index for the part, generated by the CIM Manager.

Bar Code | The ID number of the template which is carrying the part.

Leaf ID | An internal ID index generated by the CIM Manager.

L1... Ln | Additional information about other "leaves".

Event View

The Event View is used only when the CIM Manager is operating in simulation mode; it contains data only after the **Run** button is pressed.

The Event Queue is a list of events that will be generated by OpenCIM's simulation engine, in order to ensure proper functioning of the simulation.
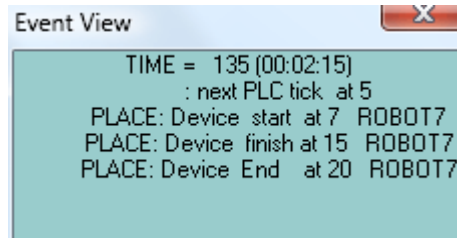
Figure 6-12: Event View

For example:

`TIME = 135 (00:02:15)`	Indicates amount of time that has passed (135 seconds, or 2 minutes, 15 seconds) since the Run button was pressed.
`PLACE: Device start at 7 ROBOT7...` `PLACE: Device finish at 15 ROBOT7`	Indicates Robot 7 will send a Start message in 7 seconds and a Finish message in 15 seconds.
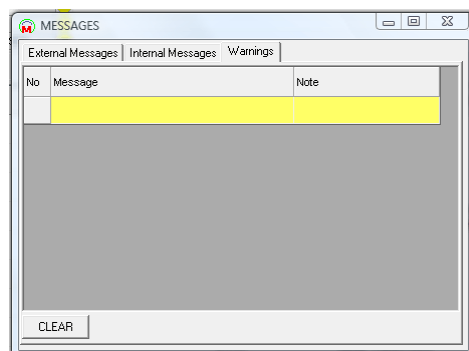
Message History

Figure 6-13: Message History Dialog Box

The Message History View allows you to view three types of messages:

External Messages: All messages that the CIM Manager sends to drivers via the TCP/IP protocol and vice versa.

Internal Messages: Enables you to check CIM Manager setup parameters, e.g. path to database files, path to utilities, simulation speed, etc.

CIM Warnings: All warnings issued by the CIM Manager. The last message is highlighted in yellow. When a warning message is issued, the Message History dialog box opens. The most frequently issued warning message is: "Part not currently available. Update storage".

CIM Scheduler

The CIM Scheduler allows you to view various production schedules and determine the most efficient one. The Scheduler is a Gantt utility that displays the exact timing and scheduling of the different phases of production.
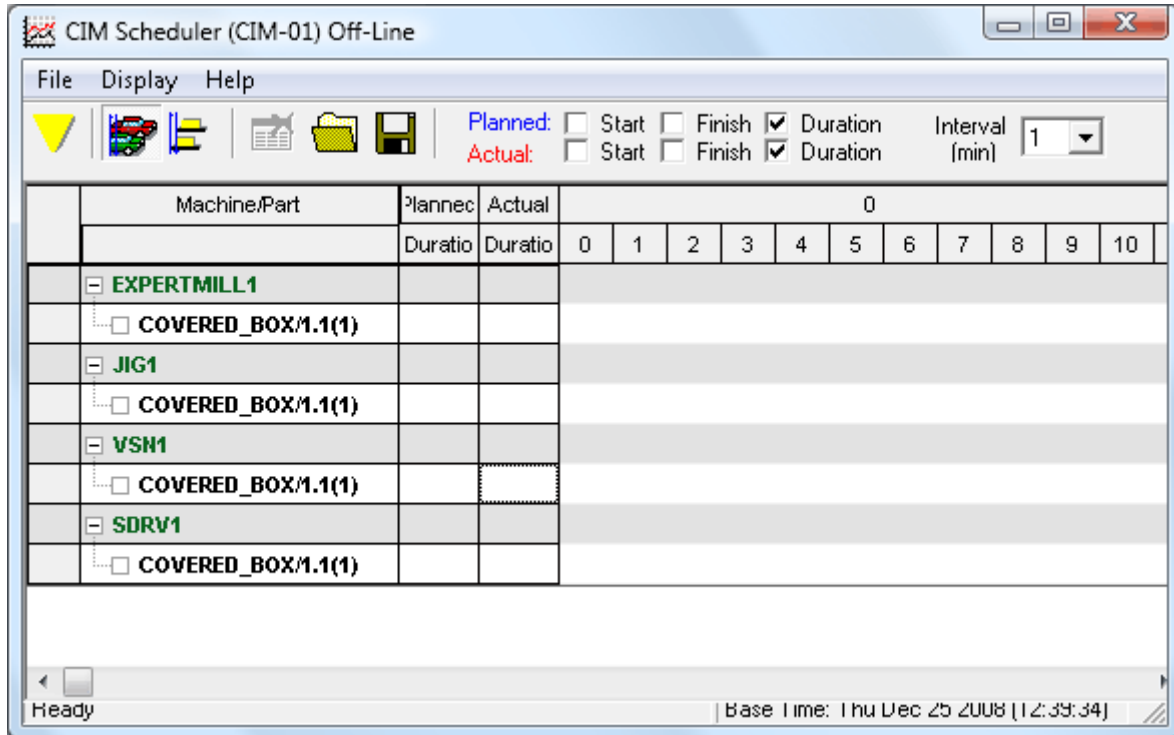
Figure 6-14: CIM Scheduler-Gantt Chart

The Scheduler can display two kinds of production schedules:

- **Planned**: This schedule is normally produced and displayed when the CIM Manager is operating in Simulation mode. Tracking mode must be activated from the CIM Mode dialog box. For further details, see *CIM Modes Dialog Box*.

- **Actual**: This schedule is normally produced and displayed when the CIM Manager is operating in Real mode. The Tracking mode is deactivated.

The left side of the Scheduler screen is a textual description, while the right side is a graphic representation (Gantt chart) of the production schedule.

Click and drag on the vertical lines in the table to increase and decrease column widths.

CIM Scheduler Toolbar and Menu Bar Options

The following table contains a brief description of each option in the CIM Scheduler toolbar in addition to the corresponding menu bar item:

Online/Offline: When the Scheduler operates on-line, it displays data from the CIM Manager. (No data is displayed until the CIM Manager commences production.).

When the Scheduler operates off-line, it displays information from Manager.Data is displayed as either Actual or Planned, depending on the definition in the [Scheduler] section of the CIM Manager's INI file.

Alternatively, select **Display | Online/Offline** from the CIM Scheduler menu bar.

Sort by Machine: Shows the activities of machines and the parts they process.

Alternatively, select **Display | Sort by Machine** from the CIM Scheduler menu bar.

Shows the progress of parts, and the machines which process them.

Alternatively, you can also select **Display | Sort by Part** from the CIM Scheduler menu bar

Enables you to clear the scheduler data .(Enabled in Online mode only.)

Alternatively, select **File | Clean** from the CIM Scheduler menu bar.

Displays the Load Data dialog box, enabling you to load additional production schedules.(Enabled in Offline mode only.).

Alternatively, select **File | Open** from the CIM Scheduler menu bar.

Displays the Save as dialog box enabling you to save the current production schedule.

Alternatively, select **File | Save As** from the CIM Scheduler menu bar.

Display Options
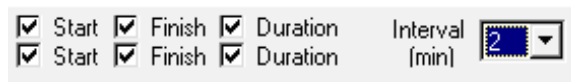
Figure 6-15: Scheduler Display Options

Zoom: Value of time interval compression in Gantt chart display.

Show: Checked items are displayed in the textual display.

Creating a Planned Production Schedule

To create a planned production schedule, do the following:

❶
❷
❸
Procedure

Generating a Planned
Production Schedule

1. Activate the CIM Manager**.**

2. From the Modes Dialog Box, activate the tracking mode.

3. Reset storage.

4. Activate the CIM Scheduler.

5. Click **Start**.

6. Click **Run**. Wait for the CIM Manager to complete an entire production cycle.

☞ *Tip:*

> *To speed up the simulation, change the value of the simulation speed in Modes Dialog Box.*

After you have generated a planned schedule, you can run the CIM Manager in real mode in order to track and display the actual schedule, and see how it compares with the planned schedule.

Graphic Display and Tracking

The OpenCIM Graphic Display and Tracking module provides a real-time 3D graphic display of a working OpenCIM cell, displaying the movement of pallets on the conveyor based on the status messages it receives as each pallet passes a conveyor station. The Graphic Tracking module estimates the position of pallets as they travel between stations and updates its display accordingly. It synchronizes its display with the actual pallet position every time a pallet passes a conveyor station.

Status Messages

When a device performs an operation on the part, its device driver transmits status messages to the CIM Manager reporting the outcome The CIM Manager forwards these messages to the Graphic Tracking module, which then updates its display accordingly.

Examples of these messages include:

- **Command Response Messages:** A device driver responding to a command sent from the CIM Manager, or a device driver responding to a command sent from another device driver (such as, a CNC device driver responding to commands sent by a robot's ACL or Scorbase device driver to open and close its door).
- **Pass Messages:** The PLC device driver sending a *Pass message* indicating that a pallet that is not needed at this station has just gone by. Pass messages are generated only to allow the Graphic Tracking module to update its conveyor display, and are not used by the CIM Manager or any other CIM entity.

Updating the Display

The CIM Manager relays the status messages to the Graphic Tracking module, which then updates its display accordingly. The display can show the following examples:

- **Parts**, as they move from device to device (such as, a robot picking up a part from a template and putting it in a CNC machine).
- **Pallets**, moving around the conveyor.

The screen display includes detailed representations of station elements such as computers, controllers, CNC machines, and robots as shown in the following figure. This module updates its display in response to real-time status messages emanating from the CIM Manager and active device drivers.

The Graphic Tracking PC can be used in the following modes:

- **Real Time Mode,** enabling you to observe the flow of parts around the CIM cell.
- **Simulation Mode**, enabling you to observe the results of different production strategies on-screen without actually operating the CIM equipment.

The graphic display module appears in the Graphic Display tab, as follows:
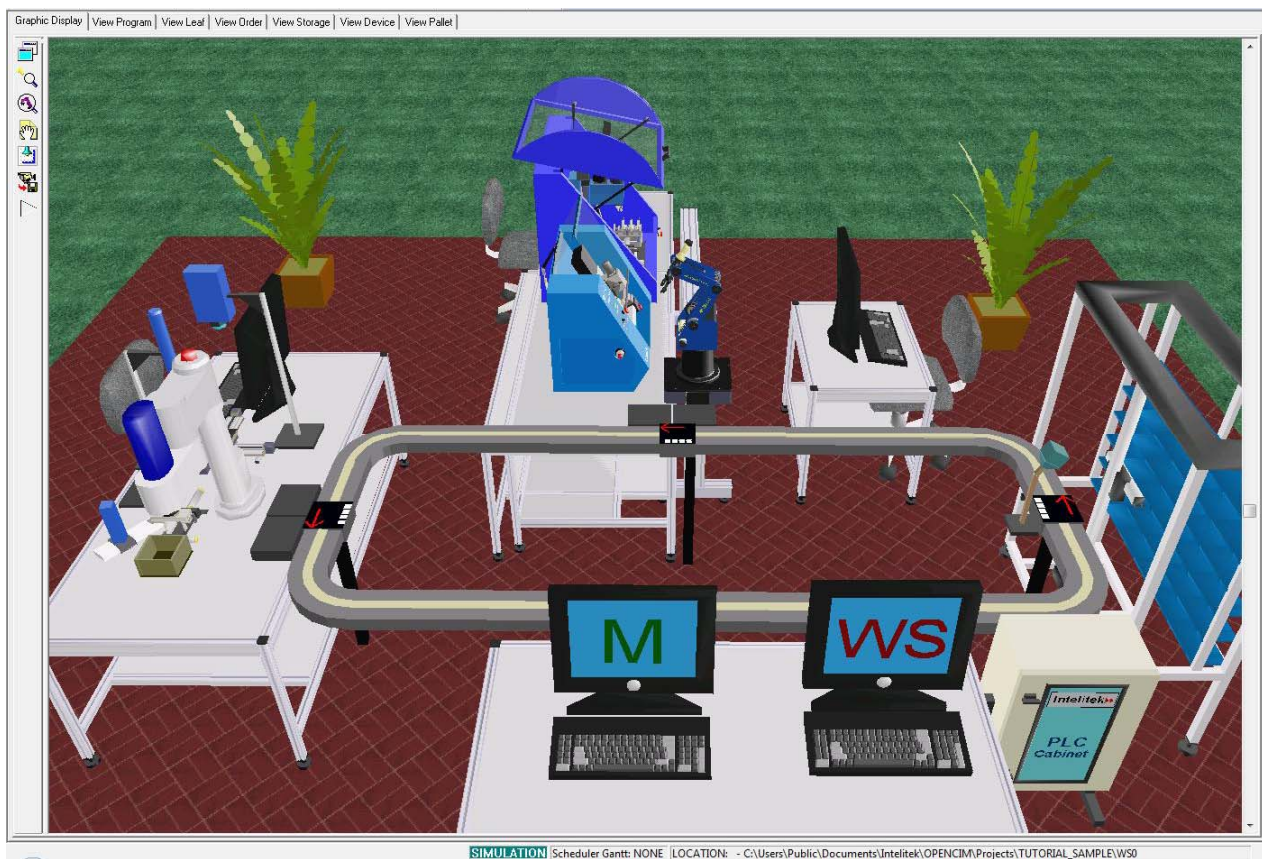
Figure 6-16: Graphic Display on Manager

Graphic Display Toolbar

The following table contains a brief description of each option in the Graphic Display toolbar:

Option	Description
	Maximize/Restore: Enables you to toggle between maximizing and restoring the Graphic Display tab.
	Redirect Camera: Defines the position that will be in the center of the image.
	Follow Me Camera: Enables you to focus on the location of specific part during a production cycle**.**
	Drag Scene: Enables you to pan the CIM cell left, right, up and down.
	Top View: Places the camera on top of the cell at the center of the image
	Save Camera Position: Saves the current position of the graphic dispay screen until the next time you enter the CIM Manager application.
	Toolbar View: Displays or hides the IDs of the objects, templates, parts, and pallets, each of which is described below.
	Show Name:Displays the names of the objects that currently exist in the CIM cell.
	Show ID:Displays the object IDs that exist in the current CIM cell.
	Show Pallets: Displays the pallet numbers that are currently on the conveyer.
	Show Templates: Displays the IDs opf the templates in the CIM cell.
	Show Parts: Displays the IDs of the parts in the CIM cell.

In addition, the following options enable you to change the view of the CIM cell:

- **Zoom In/Zoom Out:** Zooms in and out of the image by pressing the right mouse button and moving it forward or backward.
- **Rotate the Image:** Rotates the view of the image by pressing the right mouse button and moving it to the right/left.
- **Moving the Camera Up/Down:** Use the window's scroll bar to adjust the viewing angle of the image.

Alternatively, the viewing angle can also be adjusted by scrolling the mouse wheel.

The Graphic Display of a working OpenCIM cell is displayed in the CIM Manager window and it can also be displayed on another PC (WebViewer application and Remote Graphic Display.) It is possible to display three different 3D views at the same time on the same screen as shown in Figure 6-17.
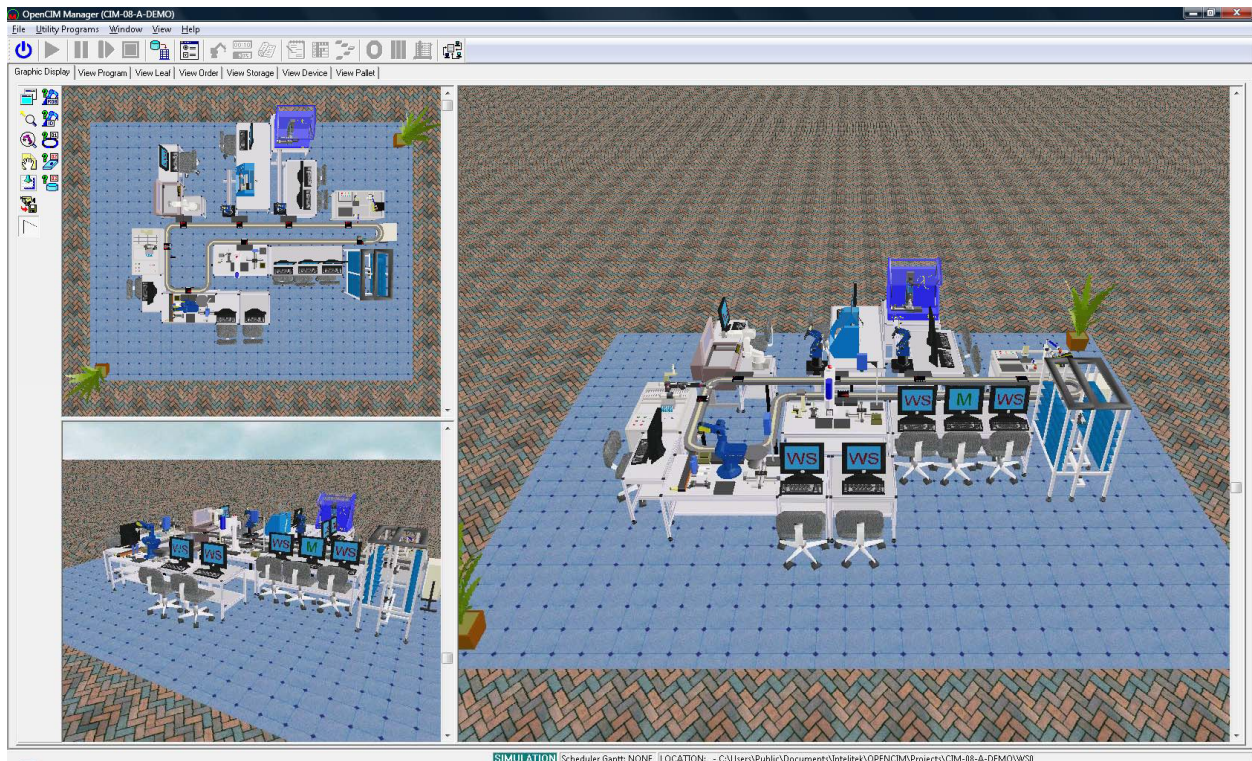
Figure 6-17: Graphic Display

Tracking the Production Process

The following procedure describes how to graphically track the production in the OpenCIM system:

❶
❷
❸

Procedure

Graphic Tracking of Production

1. Launch Project Manager . To do so, select the project and click the CIM Manager icon in the tool bar.

2. Click the Blue **Load** button. Click **Run** (Blue arrow).

3. Now you can observe the operations performed in the CIM cell in both Graphic Displays: the Manager and in the 3D views.

Manipulating the Graphic Display Views

The Graphic Display module offers two types of views, an overhead view and an elevated side view. This procedure describes how to manipulate these views according to your requirements.

Note *When the Simulation window opens, it always displays the view that was displayed when you last closed either the Graphic Display or the Virtual CIM Setup window.*

To manipulate the views:

❶
❷
❸
Procedure

Manipulating the Graphic Display

1. To change the angle of the overhead scene, place the cursor on the vertical scroll bar and drag it up and down. (It is recommended that you click on the vertical scroll bar up and down arrows.)

2. To rotate the scene, place the cursor anywhere on the screen and:
   - Click the right mouse button and drag to the right to rotate the display counterclockwise.
   - Click the right mouse button and drag to the left to rotate the display clockwise.

3. To zoom the scene, place the cursor anywhere on the screen and:
   - Click the right mouse button and drag up to zoom in.
   - Click the right mouse button and drag down to zoom out.

Changing the Focus of the Graphic Display

The following procedure describes how to change the focus of the Graphic Display.

To change the focus of the Graphic Display:

❶
❷
❸
Procedure

Changing the Focus of the Graphic Display

1. Click on **View | Redirect Camera**.

2. Click any object in the scene. It now becomes the center point for the display manipulation. The view changes to an overhead scene (if it is not already), which you can now manipulate, as described above.

The *Text* menu allows you to select the kind of captions you want to include in the graphic display. Only one kind of text can be selected at a time.

None	No text. Select **None** to remove the currently displayed caption. You may then select another kind of text. (Note that there is no checkmark in the menu to indicate your selection.)
Name	Name of machines and devices.
Ext. ID	External ID number, as defined in the Virtual CIM Setup.
Pallets	Displays the ID number of the pallets.
Templates	Displays the ID number of the templates.
Parts	Displays the ID number of the parts.

The *File* menu offers the following options:

Open	Loads a new graphic CIM cell. *Do not use.*
Exit	Quits the Graphic Display module.

7

CIM Manager Utility Programs

This chapter describes the CIM Utility Programs which are used for preparing the OpenCIM system for production. These programs are an integral part of the CIM Manager software and can be accessed from the Utility Programs menu of the CIM Manager menu bar. The Utility Programs menu includes the following:

- **Machine and Process Definitions**, introduces the CIM Machine Definition window, enabling you to define the machines and processes in OpenCIM.
- **Part Definition**, introduces the CIM Part Definition window, enabling you to define the parts that OpenCIM can manufacture.
- **Storage Definition**, introduces the CIM Storage Manager, enabling you to track the parts in storage.
- **MRP** (**Material Requirements Planning**), introduces the CIM MRP window, enabling you to create customer lists and product orders.
- **Optimization**, introduces the CIM Optimization Definition window for defining queue algorithms as well as additional optimization methods used in OpenCIM.
- **Performance Analysis**, introduces the CIM Performance Analysis window for viewing and analyzing information generated form the manufacturing cycle.
- **Reports**, introduces the CIM Report Generator enabling you to generate predefined or customized reports for viewing and printing.

You can use these programs to view some existing sample definitions to assist you in making your own.

As you read through this chapter, it is recommended that you perform the "Procedures." These tutorials will help you become familiar with using the OpenCIM software.

The examples shown in this manual are based on the sample project TUTORIAL_SAMPLE which is located in the projects archive list in the project Manager. (For details on transferring a project from Archive to User refer to Chapter 5).

Machine and Process Definitions

When you define a machine, you actually define the specific process a machine will perform. Machine names are usually predefined in the Virtual CIM Setup and only need to be selected from the Machine Name drop-down list.

The process name enables the CIM Manager to determine which machine is capable of performing the specific work required to produce a part (as defined in the Process field in the Part Process Table in the Part Definition form). If two machines that are capable of performing the specific process are available, the CIM Manager tries to optimize the use of these two machines to complete the process (see *Optimization* in this chapter).

The Machine Definition form lets you view any machine that has been defined for the system. You can define new or modify existing processes for the machine to perform. A *machine record* contains the machine name and one or more defined processes (process record). Each field and the control buttons associated with this form are described in detail in this section.

The CIM Machine Definition window displayed below is accessed by from the CIM Manager Main Window, by selecting **Utility Programs** | **Machine Definition** from the Menu bar.

Figure 7-1: Machine and Process Definition Form

Machine Definition Window

Main Menu

Option	Description
File	Contains these file options: Save All, Save Selected Machine, Print Machine Report, Exit (from Machine Definition screen). The first three options also appear as tool buttons in the Toolbar (see below).
Edit	Contains the following options for editing rows: Insert Before, Insert After, Delete Row. The rows inserted are copies of the original row, enabling you to edit the content.
	You can also access these options by right-clicking the process list cell you want to edit.
Help	On-line help.

Toolbar

Option	Description
	Saves all machine records to disk.
	Saves the selected machine record to disk.
	Automatically resizes the columns to accommodate long values (when the window is maximized).
	Print machine report.
JIGXY6	Select a machine from the machine name drop-down list. The names that appear in the list are defined in the Virtual CIM Setup.

Machine Process Table

The data fields above the table are also displayed as columns in the table. Their descriptions are provided below.

Option	Description
Machine Name	A descriptive name which uniquely identifies the machine. You can edit/examine the record for a specific machine by selecting that machine name from the drop-down list in the toolbar. All machines that are defined in the Virtual CIM Setup appear in this list.
Process	The name of a production process that can be performed by this machine. A Process Name can only be used once for a given machine.
	The name should be easily recognizable to CIM users and may contain the characters A–Z, 0–9 and underscore (_), but no spaces.
	This Process Name is assigned to a part in the Part Definition form (in the Process field of the Part Processes Table).
	Assigning a process to a part instead of a machine can have advantages when there are two or more machines capable of performing the same process. Having more than one machine capable of performing a given process allows the CIM Manager to select the machine which can process a part most efficiently and redirect production if one machine fails.
Note	*For different machines that can perform the same process, you should enter the same process name. Likewise, do NOT use the same process name to refer to processes on different machines which do not perform the same operation on the same part.*

The following reserved words cannot be used as Process names:

ALLOC	GET	PACK
ASSEMBLY	GET_FIX	PLACE
BASE	MAKE	PROCESS
CNC	MOVE	QC
DELIVER	NEXT	RENAME
END_ASSEMBLY	NOP	TARGET
FREE	ONFAIL	TRANSFER

Option	Description
File	A file containing the G-code program or other program associated with this process. This file name can include a valid directory path to a file. If no path is specified, the CIM Manager expects to find this file in the current working directory associated with the device driver for this machine. A file can contain one machine control program. Different machines that perform the same process will have their respective control programs stored in different files.
Program	The name of the machine control program associated with the process being defined. This Program Name is used by an ACL controller which is operating a machine.
Fail(%)	Your estimate of the number of rejected parts that will result when this process is run on this machine (0 - 100%). The CIM Manager takes this value into consideration only when simulating a quality control process.
Duration	The number of minutes this process takes to produce one part. The CIM Manager takes this value into consideration when choosing among multiple machines that can run the same process. Format is *hh:mm:ss*
Parameters	This string of arguments is passed to a machine control program associated with this process.
WS	The workstation in which the machine is placed. Automatically displayed by the system (as defined in the Virtual CIM Setup).
Machine Type	The type of machine selected. Automatically displayed by the system (as defined in the Virtual CIM Setup).
Action Type	A label that defines the characteristics associated with a process. One of these Action Types (in the data field above the table) is selected by default:

Action Type	Description
Assembly	A process which involves the assembly of two or more subparts.
QC	A process involving a test that reports a Pass/Fail result to the CIM Manager. If the result is Fail, the rejected part is redone. A quality control process requires an ONFAIL entry in the Part Processes table in the Part Definition form see "Part Definition" below.
CNC	A process which has G-code program(s) associated with it. The CIM Manager downloads the G-code file specified in the File field to the CNC machine (unless this file is already resident in the CNC machine).
Process	A basic machine operation which does not require any special action beforehand or afterwards. Runs the ACL program specified in the Program field.

Option	Description
`Place`	A robot operation used for non-standard operations performed by a robot. The File and Program fields will be blank.
Robot- controlled	Specifies if a robot is needed to perform the process. For example, if a welding action is performed by a robot, specifying YES signals the CIM Manager that the robot is in use and is not free to perform another operation. This option is available only if the machine selected can use a robot to perform a task, and if the Action Type is Process.
Cost per Hour	Estimated hourly cost to run this machine. The CIM Manager uses this as one of the criteria in order to decide on the optimum production method.

List of Preloaded Programs

Tasks are control programs that can be downloaded to a machine (e.g., G-code to a CNC machine). The CIM manager keeps track of which programs currently reside in a machine's memory. If a certain process requires a machine control program that is not resident, the CIM Manager instructs the CNC device driver to download it to the machine.

Option	Description
Max Preloaded Programs	The number of control programs that can reside in a machine's memory at one time. Once this number is exceeded, the CIM Manager begins overwriting programs in the machine's memory when it needs to download a new program.
List of Preloaded Programs	The current status of control programs that are loaded in the machine's memory. This box is for information purposes only; it cannot be used to change the programs residing in a machine.

How to Define a Machine

The procedures presented below refer to an OpenCIM sample application for producing a simple, covered box (product). When defining the process for the covered box, two processes need to be defined: CNC milling and assembly.

These procedures represent a simplified example. When defining more complicated applications, entries to other fields will also be required.

Note

❶
❷
❸
Procedure

A. Defining the Milling Process

1. In the Machine Definition form, select **EXPERTMILL1** from the Machine Name drop-down list; this is the name for the milling machine, as defined in the Virtual CIM Setup. The MILL1 record becomes the current record and appears in green in the Machine Process Table. The record contains the process(es) defined for the machine. The workstation is shown as WS3, the machine type as M-MACHINE, the action type as CNC, and Robot-controlled as No and gray.

2. In the Process column, type in the Process Name `MILL2`. (You will need to make sure—either now or later—that `MILL2` appears in the Process field in the Part Process Table in the Part Definition form).

3. In the File column, type in `GCODE.NC`, for example, as the name of the G-code program file that contains the instructions for this type of process.

4. In the Duration column, type in the amount of time it takes the mill to perform this operation, e.g., thirty seconds.

5. Your screen should now look like this:

Figure 7-2: CIM Machine Definition – Defining the Milling Process

---
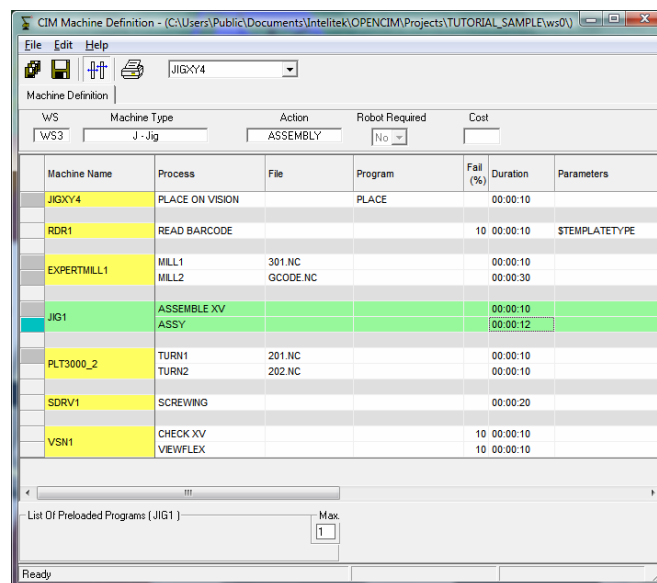
6. Click [icon] to save the currently displayed information to the database. You can now generate and view a Machine or Process Report by using the Report Program.

7. Continue with the procedure *Defining the Assembly* to define the other process which is required to produce the covered box

Note: *The assembly operation is usually performed by a jig device (such as a pneumatic jig) not a machine.*

B. Defining the Assembly Process

1. In the Machine Definition form, select JIG1, from the Machine Name drop-down list; this is the name of the device that performs assembly operations, as defined in the Virtual CIM Setup. The JIG1 record becomes the current record and appears in green in the Machine Process Table. The workstation is shown as WS3, the machine type as J-JIG, the action type as ASSEMBLY, and Robot-controlled as NO and gray.

2. In the Process column, type in the Process Name `ASSY`. (You will need to make sure—either now or later—that ASSY appears in the Process field in the Part Process Table in the Part Definition form.)

3. In the Duration column, type in the amount of time it takes the mill to perform this operation, e.g., 12 seconds.

4. Your screen should now look like this:

Figure 7-3: CIM Machine Definition – Defining the Assembly Process

5. Click [icon] to save the information to the database.

You can now generate and view a Machine or Process Report by using the Report Program.

Part Definition

A product is manufactured from a group of subparts (bill of materials) that are put together according to a specified set of machine processes. Starting with a set of raw materials (supplied parts), you define parts at the intermediate stages of production required to assemble a final product.

The Part Definition screen, or form, allows you to enter the bill of materials and the associated production processes used to produce a part. Using the Part Definition form, you can either:

- Modify/view the production process for an existing product.
- Describe the production process for a new product.

Defining a new product involves the following steps:

- Drawing a part definition tree.
- Setting up all machine processes necessary to produce a product and all its subparts.
- Determining what new template designs are required to handle all the parts involved and assign these designs template ID numbers.
- Determining the types of racks that can hold each subpart.

The Part Definition form for Product (or Phantom) parts lets you create, view, or modify the current part (either a product or its subparts). A *part record* contains all the fields shown on the Part Definition form below. Each field and the control buttons associated with this form are described in detail in this section.

Figure 7-4: Part Definition Form for Product or Phantom Part

If you define the part as Supplied, the Part Process table will be replaced by a section containing data regarding the supplier and supplied material, as shown below:
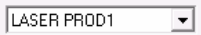
Figure 7-5: Part Definition Form for Supplied Part

Part Definition Form
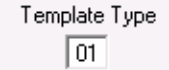
Main Menu

Option	Description
File	Contains these file options: New Part, Save Current Part, Save All, Print, Exit (from Part Definition screen). The first five options also appear as tool buttons in the Toolbar (see below).
Edit	Contains these row and part editing options: Insert Before, Insert After, Copy Part, Paste Part, Delete Row, Delete Part. You can also access these options by right-clicking the process list cell you want to edit.
Help	On-line help.

Toolbar

Option	Description
	Defines new part
	Saves all part records to database.
	Saves the selected part record to database.
	Automatically resizes the columns to accommodate long values (when the window is maximized).
	Prints part report.
LASER PROD1	Selects a predefined part from the part name drop-down list.
3	Selects a predefined part ID from the part ID drop-down list.

Information Bar

Option	Description
Template Type 01	Defines the template type (01-99) whose pin arrangement can accommodate the selected part.
Rack/Feeder	Defines the types of racks/feeders that are capable of accomodating the selected part.
Color	Defines the color of the parts that can be viewed on the conveyer. You can define a different color for each part. The part color will change after the work on the part is completed. Note: The color will not be changed for custom parts which are provided in the OpenCIM installation.
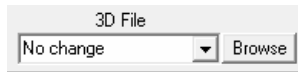
Option	Description
3D File No change ▾ Browse	Defines a part shape from either the list of supplied or previously added shape parts, or by browsing to a new user defined part. When a new 3D file is added by browsing, the file is added to the project directory for example: - In Windows XP: [default system drive (for example C:)]\Documents and Settings\All Users\Documents\Intelitek\OpenCIM\Projects\ TUTORIAL_SAMPLE\CIM_ CUSTOM_parts. - In Windows Vista/7: [default system drive (for example C:)]\Users\Public\Documents\Intelitek\OpenCIM\Projects\ TUTORIAL_SAMPLE\CIM_ CUSTOM_parts.

Designing User Parts and Objects

Users can add other parts and objects over the existing ones that are provided with the installation of the software.

There are two stage to creating new parts and objects.

- Creating Parts and Objects
- Adding Parts and Objects to OpenCIM Projects

Creating Parts and Objects

There are three ways to create new parts and objects:

- Modify an Existing User Part File
- Create a New Part/Object File
- Import a CAD File

Modify an Existing User Part File

Existing user parts that are provided with the installation may be modified to suit the requirements of the user.

To modify an existing user part:

1. Open the CIM_CUSTOM_PARTS folder located in the following directory:

    - Windows XP: [default system drive (for example C:)]\Documents And Settings\All Users\Documents\ Intelitek\OpenCIM\CIM_CUSTOM_parts

    - Windows Vista/7: [default system drive (for example C:)]\Users\Public\Public Documents\Intelitek\OpenCIM\CIM_CUSTOM_parts.

2. Using a text editor that saves files in plain ASCII format, such as Notepad, open one of the existing user part files in the directory. Below is an example of a part file.

```
ModelBegin
TransformBegin
Color 0.0 1.0 1.0
Surface 0.4 0.3 0.2
```

```
Opacity 1.000000
LightSampling Facet
GeometrySampling Solid
TextureModes Lit
Texture NULL
ClumpBegin
Translate 0.0 0.0251 0.0
Tag 1
Block 0.05 0.05 0.05
ClumpBegin
Tag 2001
Color 1 0 0
Opacity 1
Block 0.03 0.02 0.07
ClumpEnd
TransformEnd
ModelEnd
```

3. Edit the file to modify the part. (Refer to the documentation provided in Program files\intelitek\opencim\sources\Graphics\RW\Documentation for details on RWX commands).

See the Adding Parts and Objects to OpenCIM Projects section for details on adding the file to OpenCIM project(s).

Create a New Part/Object File

To create your own user part and user object files:

1. Open a text editor that saves files in plain ASCII text, such as Notepad.
2. Write the program to create your own part. (Refer to the documentation provided in Program files\intelitek\opencim\sources\Graphics\ RW\Documentation for details on RWX commands).

See the Adding Parts and Objects to OpenCIM Projects for details on adding the file to OpenCIM project(s).

Import a CAD File

You can create a 3D object in a CAD program, and import it to CIM Part Definition.

To import a CAD file:

1. Create a file in any CAD program and save it as a *.dxf or *.3dc file.
2. Use the RW3DCONV converter to convert the file to RWX format. This converter is provided with the installation of OpenCIM, and is located in the following directory: program: files\intelitek\opencim\sources\Graphics\RW\ converter.

Ensure that there are no more than 1000 polygons in the CAD file.

Note

See the next section for details on adding the file to OpenCIM an project(s).

For more information on RWX files, refer to the RWX help file which is provided with the OpenCIM installation, and is located in the following directory: Program files\intelitek\opencim\sources\Graphics\RW\Documentation.

Adding Parts and Objects to OpenCIM Projects

Parts and objects can be added to a single project only, or to all projects, such that they will appear in the 3D file are drop-down menu of every project in OpenCIM.

To add the part or object to a single project:

1. Save the file as a .rwx file.
2. Open the Part Definition utility by selecting **Utility Programs | Part Definition** from the OpenCIM Project Manager.
3. Click **Browse** in the 3d File area at the bottom right of the window.
4. Browse to the location of the file that you saved and click **OK.** The part is added into the project, and appears in the *3D File* drop-down menu. A copy of the file is added to the following directory:
   - XP: [default system drive (for example C:)]\Documents And Settings\All Users\Documents\ Intelitek\OpenCIM\ TUTORIAL_SAMPLE\CIM_CUSTOM_parts.
   - Vista/7: [default system drive (for example C:)]\Users\Public\Documents\Intelitek\OpenCIM\Projects\TUTORIAL_SAMPL E\CIM_CUSTOM_parts.

To add the file to all projects, save the file as a .rwx file in the following directory:
   - XP: [default system drive (for example C:)]\Documents And Settings\All Users\Documents\ Intelitek\OpenCIM\CIM_CUSTOM_parts
   - Vista/7: [default system drive (for example C:)]\Users\Public\Public Documents\Intelitek\OpenCIM\CIM_CUSTOM_parts.

Part Table

Option	Description
Part Type	Select one of the following types for this part:

Product	A part that can be ordered from the CIM. The final part at the top of the part definition tree is always defined as a product. Part is the product that is produced by the CIM system. In some industrial software, the term "MAKE" is often used to refer to the product.
Supplied	A part received from an outside source, i.e. a part not produced by the CIM, therefore it does not require a process definition. Supplied parts do not contain any entries in their Part Process tables. A supplied part is found only at the bottom of the Part Definition tree. In some industrial software, the term "BUY" is often used to refer to the supplied part.
Phantom	A part or subpart which has failed QC. This definition allows the CIM Manager to issue instructions on how to handle a rejected part. *Phantom parts cannot be ordered*.

Product Part Data

Option	Description
Part Name	A string which uniquely identifies this part (i.e. two parts cannot have the same name). The name should be easily recognizable to CIM users. The string may contain the characters A–Z, 0–9 and underscore (_), but no spaces.
Part ID	A numeric value (1 – 999) which uniquely identifies this part (i.e. two parts cannot have the same ID). This Part ID can be used with devices which require a numeric part identifier. For example, the ACL controller uses the Part ID to activate the appropriate control program to handle this part.
Subpart	The name of a material used to produce the current part.

Option	Description
	A subpart must be defined in its own Part Definition record. A subpart can either be a raw material (i.e. a Supplied Part) or a part produced by the CIM (i.e. a Phantom Part or a Product).
	Some rows in the Part Process table require a Subpart name while others do not. A Subpart name is required in the following circumstances:

- A Subpart name is required in row 1 of the Subpart column.
- A Subpart name is required for each part that is included in an assembly.
- A Phantom Subpart name is required after each quality control test in order to associate a name with the ONFAIL exception handler.

After the first row, a subpart name is not required if the process being performed operates on the same part that was listed in the previous row. For example, the first row could specify the name of a cube that is to be machined into a box. The second row specifies a process that drills a hole in the box. In this case, the subpart field of the second row would be blank because the drill operates on the same subpart specified in row one.

If you need more than one of a subpart, add a separate row to the Part Process table for each unit.

A circular definition error will result if you enter a subpart name that matches the name of the part being defined (i.e. Subpart = Part). This error will also occur if any of the subparts in turn contain a subpart that matches the Part Name being defined.

Option	Description
Process	Enter the name of a production process that has been defined in the Process field of the Machine Definition screen.
	If this process exists on more than one machine, the CIM Manager selects the machine to use according to its production strategy (e.g. minimize cost, minimize production time, etc.).
Parameters	The Parameters field specifies how to carry out this process when it is performed for the current part.

For quality control devices, the parameter string is used to specify the type of QC test and the range of acceptable values.

For a machine that performs assembly operations, the parameter string specifies where to put the part that is being added to the assembly. If this target location contains compartments, you can add an optional index for the compartment number.

The table below details how parameters are used by several devices:

Device	Example	Description	Note
ROBOT-VISIONpro	1, 4	`type of test, minimum value, [maximum value]`	If maximum value is omitted, the minimum value represents the single acceptable value.

Device	Example	Description	Note
Laser Scan Meter	`1,150, 160`	`type of test, minimum value, [maximum value]`	If maximum value is omitted, the minimum value represents the single acceptable value (with a tolerance of ±5%).
Assembly Machine	`BOX, 2`	`target location, [target index]`	Places subpart assembly `BOX` in location #2.

The following system variables can be used in the Process definition.

Variable	Description
`$PARTID`	Part ID as defined in the Part Definition form.
`$TEMPLATEID`	The Template ID (six digits) defined in the Storage Definition form.
`$TEMPLATETYPE`	The Template Type (two digits) defined in the Storage Definition form.
`$PRIORITY`	The Priority defined in the Manufacturing Order.
`$DURATION`	The Duration defined in the Machine Definition form.

Supplied Part Data

When you select Supplied as the Part Type, the Part Process table is replaced by a form that allows you to define the supplied part.

Option	Description
Part Name	A string which uniquely identifies this part (i.e. two parts cannot have the same name). The name should be easily recognizable to CIM users. The string may contain the characters A–Z, 0–9 and underscore (_), but no spaces.
Part ID	A numeric value (1 – 999) which uniquely identifies this part (i.e. two parts cannot have the same ID). This Part ID can be used with devices that require a numeric part identifier. For example, the ACL controller uses the Part ID to activate the appropriate control program to handle this part.
Supplier Name	The name of the supplier from whom the raw material / supplied part is purchased.
Supplier Catalog Number	The supplier's catalog number for the raw material / supplied part.
Minimum Order	The smallest quantity of this part which can be purchased from the supplier at a time.
Safety Stock	The number of units of this material / part required by the CIM cell to guarantee production without interruption.
Cost	The cost of one unit of the raw material / supplied part.
Supply time (days)	The amount of time it takes the supplier to deliver this material / part to the CIM cell.

Option	Description
Description	A description of the part being defined that explains what it is and where it is to be used.
Template Type	The Template type (01 – 99) whose pin arrangement can accommodate this part.
Rack/ Feeder Types	If this part is to be stored in a feeder, specify which types of feeder are capable of accommodating this part (Feeder Type > 100). Feeder types are defined in the Virtual CIM Setup. You can specify multiple feeder types in this field; each one separated by a comma (e.g. `101, 102, 103`). Selections are made by choosing from a drop-down list or by typing in the entry. Similarly, if this part is to be stored temporarily in a rack during processing, specify which types of racks are capable of accommodating this part. See Rack Type definition above.

How to Define a Part

In order to define a part, it is important that you understand the entire operation. A part is only the starting point. This part (supplied, raw material) moves within the CIM system according to a predefined path, the part is processed (with another part) and the product is then created.

In order to define a part, you need to:

- Define the supplied material(s)
- Define the process that must be performed on the material(s)
- Define how to assemble the parts (processed and supplied materials)

These concepts can be better explained by referring back to the OpenCIM sample application of producing a simple, covered box from a small, solid cube and a matching cover. From this example we can determine that:

Raw Material #1	Box
Raw Material #2	Cover
Product	Covered_Box

❶
❷
❸
Procedure

Defining the Raw
Material(s)

1. In the Part Type form, select Supplied parts.
2. From Main Menu select **FILE | NEW PART** or click the "New" button on the toolbar.
3. In the Part Name field, select BOX (the solid cube from which the box will be milled) from the drop-down list.
4. In the Part ID field, enter a unique ID for this raw material; for example, 1.
5. In the Template Type field, enter an identifying number for the type of template which will be dedicated to carrying this part; e.g., 01. (This data will be read and used by the Storage Definition program.)
6. In the Rack/Feeder Type, select the type of rack that will be used to hold this part at the workstation; e.g., Rack 201. If a rack or a feeder is not used, leave the field blank.

    The remaining fields are not required to enable production; they are used to provide statistical data.
7. Click **Save** to save the part.
8. Click **New** and repeat steps #2 – 6 for the other raw material which will be used in the assembly: COVER. Your screen should now look like this:

Figure 7-6: Part Definition Form for Supplied Parts – Defining a Part

9. Continue on to the procedure "Defining the Product" as described below.

When to Define a Subpart (New Part)

Use the following criteria to help determine when to create a subpart at an intermediate stage of production (by entering a name in the Subpart field):

- Define a new subpart if it is needed in the production of other products.
- Define a new subpart to enable an order to be placed for this part (e.g. for use as a spare part).
- Define a new subpart if it requires a different template type.
- Define a new subpart if it is to be used in the assembly of some other product.

How to Define an Assembly Process

An assembled part (assembly) always has at least two rows. The Subpart name specified in row 1 is referred to as the original material.

The assembly is always performed at or by a machine (such as a pneumatic vise), which is often defined as JIG in the list of Machine Names in the Machine Definition form.

The process for the subpart in the second row must be defined in the Machine Definition form as Action Type **Assembly**.

❶
❷
❸

Procedure

Defining the Product

1. In the Part Type form, select Product parts.

2. From Main Menu select **File | New Part** or click the **New** button on the toolbar.

3. In the Part Name field, enter `COVERED_BOX` (the name of the product).

4. In the Part ID field, enter a unique ID for this product; for example, `3 or accept the default ID number.`

5. Select the Subpart cell. Choose `BOX` from the drop-down list and type MILL2 as the Process.

6. Add a new process row by right-clicking on any cell into the current row and choose **Insert After** from the edit menu. In the second row in subpart field, choose `COVER` from drop down list and type ASSY as the Process.

7. In the Template Type field, enter an identifying number for the type of template which will be dedicated to carrying this part; e.g., `01`. (This data will be read and used by the Storage Definition program.)

8. It is not necessary to specify a Rack/Feeder type for the final product.

Your screen should now look like this:

Figure 7-7: Part Definition Form for Product Parts

9. Click ![save icon] to save .

10. You can now generate and view a Part Definition Report by using the Report Program.

![Note icon]

Note

The above listed procedures represent a simplified example. When defining more complicated parts it may be necessary to:

- *Enter products in the Subpart column of the Part Process Table.*
- *Use only predefined process names in the Process column of the Part Process Table.*
- *Add parameters.*

How to Define Quality Control Processes

Whenever you include a quality control process in a Part Definition, you must make provisions for how to handle the rejected part if it fails the quality control test by defining a special part (*quality control exception handler*). While a rejected part is being processed, the CIM Manager begins producing a replacement part.
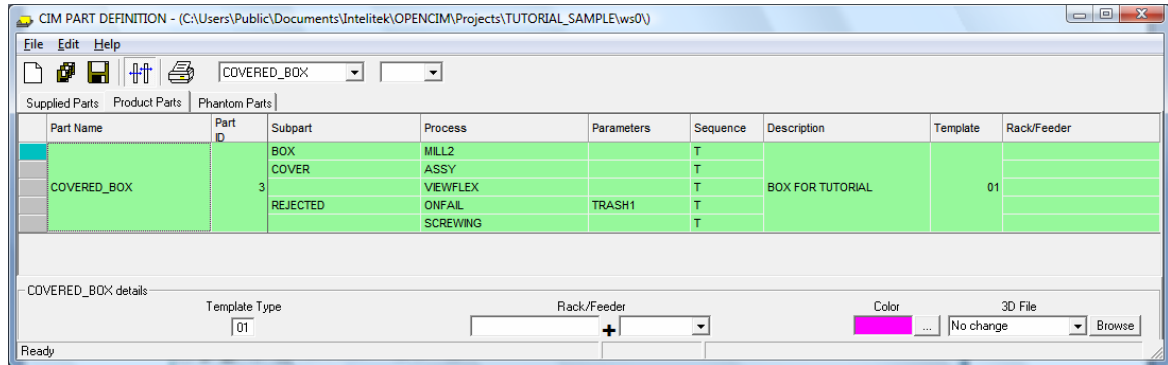
Figure 7-8: Including a Quality Control Check in a Part Definition

Table Entry	Explanation
PROCESS │ VIEWFLEX	Quality Control Test
SUBPART │ REJECTED	Quality Control Test Failed (exception handler)
PROCESS │ SCREWING	Quality Control Test Passed (continue from here)

The following procedure details the steps involved in handling rejected parts:

❶
❷
❸
Procedure

Setting Up the QC Exception Handler

1. In the Process column of the Part Process table in the Part Definition form, enter a quality control test, i.e., a process whose Action Type is defined as QC in the Machine Definition form, for example: **Viewflex**.

2. In the next row of the table, in the Process column, enter **ONFAIL**.

3. In the Subpart column of this row, enter a unique name for the part, e.g. **Rejected**, and make sure it has been defined as a **Phantom** part.

4. Click **Save** when you are finished.

5. For all subsequent rows in the Part Process table, assume that the part has passed the quality control test. Continue defining the normal production steps for this part.

Storage Definition

The CIM Manager must keep track of which parts are in storage and which templates are available to move these parts from station to station on the conveyer. You can use the Storage Definition form to:

- Update the contents (part and/or template) of storage locations.
- Create/modify template codes.
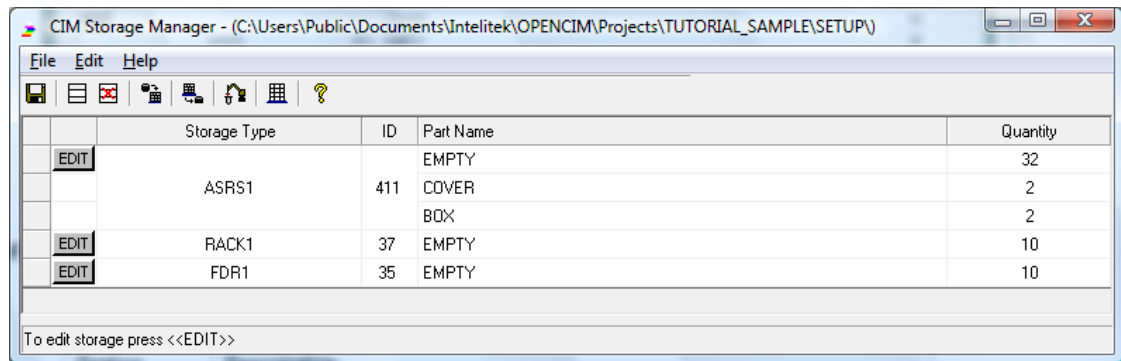
Figure 7-9: CIM Storage Manager Window

Storage Manager Form

The Storage Manager administrates all types of materials used in an OpenCIM cell. There are three types of Storage: ASRS (automated storage and retrieval system), Rack and Feeder.

Option	Description
ASRS	The ASRS is the main storage device in an OpenCIM cell. It serves as a warehouse for parts in various stages of production. ASRS cells contain templates, either empty or loaded with parts.
Rack	This type of storage can contain parts in any stage of production. Templates cannot be stored in racks.
Feeder	Contains raw material only.

Main Menu

Option	Description
File	Contains these file options: Save to DataBase, Reset Storage, Create Default Storage, Clear Temporary Storage, Initialize Storage, Exit (from Storage Manager screen). These options also appear as tool buttons in the Toolbar (see below).
Edit	Contains these part editing options: Add Part to Storage, Delete Part from Storage.
Help	On-line help.

Toolbar

Option	Description
	Saves the current Storage configuration to the database.

⊟	Adds a new row in the ASRS block. This is used when the location of the part within the ASRS is not important.
✖	Deletes a row from the ASRS block.
▦	Resets storage or default storage. Restores a predefined configuration of the storage from the backup database file.
▦	Creates default storage. Creates a backup database file of the current storage configuration.
⚙	Clears temporary storage. Removes any part or template from Temporary Storage devices.
▦	Initializes storage. Removes all parts from all devices leaving them empty of parts. **Deletes ALL storage data from database**.

When you activate Initialize Storage, you must close the Manager window and reopen it again in order to update the storage database.

Note

All devices that do not contain a part or a template at the beginning or at the end of a complete production round are considered temporary storage devices. (e.g. robot, buffer, machine, conveyor pallets, etc.).

Note

Storage Data Table

The storage devices that appear in the Storage Type column are defined in the Virtual CIM Setup. For further information, see "Edit Menu: New Object" in Chapter 7.

ASRS Definition

Click EDIT in the leftmost column of the ASRS row to display the ASRS Storage Definition form:
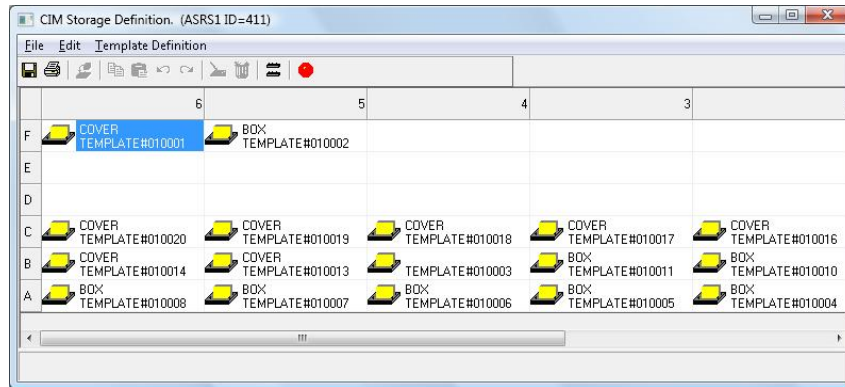
Figure 7-10: ASRS Definition Form

Use the following edit options to configure the ASRS:

Option	Description
Edit Cell	Edits the selected cell. You can fill the cell with a defined part on a template or fill the cell with an empty template. You also can edit a cell by double-clicking on any cell to open the Cell Edit form.
Delete Part	Deletes the part placed in the template leaving it empty.
Clear Cell	Clears the contents of the current cell (i.e. erases both template and part from this cell).
Template Definition	Opens the Templates form where you can define a new template type or delete an existing one.

The standard Windows options (copy, paste, undo, redo, etc.) are also available.

How to Modify the Contents of an ASRS Storage Cell

Whenever you add or remove a part or a template from a storage cell, use the Storage Definition form to register the change. The following three procedures explain how to:

- Add a part to storage cell.
- Add a blank template to a cell.
- Clear the contents of a cell

❶
❷
❸
Procedure

Inserting a Part in a Storage Cell

1. Move the cursor to the desired cell and double-click it.
2. Select the part to add from the Part name list box.
3. Click [save icon] to save this change.

❶ ❷ ❸ Procedure Inserting a Blank Template in a Storage Cell	1. Move the cursor to the desired cell and double-click it.
	2. Select the Template number from the Template list box. If the template is not defined, proceed to define it.
	3. Click [floppy disk icon] to save the change.

❶ ❷ ❸ Procedure Clearing a Storage Cell	1. Move the cursor to the desired cell.
	2. Click the **Clear Cell** [trash icon] button.
	3. Click the **Save** [floppy disk icon] button to save the change.

How to Define a Template

Templates are special trays that can be customized with an array of pins to hold different parts. Each part has a unique template type that can hold it. A specific type of template can hold various types of parts that fit into its pin arrangement.

Each template has a specific six-digit ID number (standard series starting with 010001 to 010040 and ending with 090001 to 090040) that identifies it. This number appears on an optional sticker that is affixed to the side of the template that faces the barcode or RFID Reader.

The first (leftmost) two of the six digits represent the template type number. All six digits are used as an ID number for the specific template.

RFID tags must be assigned a template ID in the RFID device driver. For more information, refer to the subsection How to Assign a Template ID to an RFID Tag below.

❶ ❷ ❸ Procedure Adding a Template Code	1. Enter the Template Definition in the ASRS form by clicking the **Template Definition** [icon] button.
	2. Click the **Add New Template** [icon] button.
	3. Type the template's two-digit number.
	4. Click the **Save** [floppy disk icon] button to save the change.

How to Assign a Template ID to an RFID Tag

RFID tags each have a unique tag serial number. In order to use RFID tags in a CIM system, they must be assigned a template ID.

To assign a template ID to an RFID tag:

1. Launch the RFID device driver as follows:

   a) Locate and launch the relevant workstation launcher as described in OpenCIM Loader: DDLoader.EXE, on page 12-2. The CIM DDLoader is displayed.
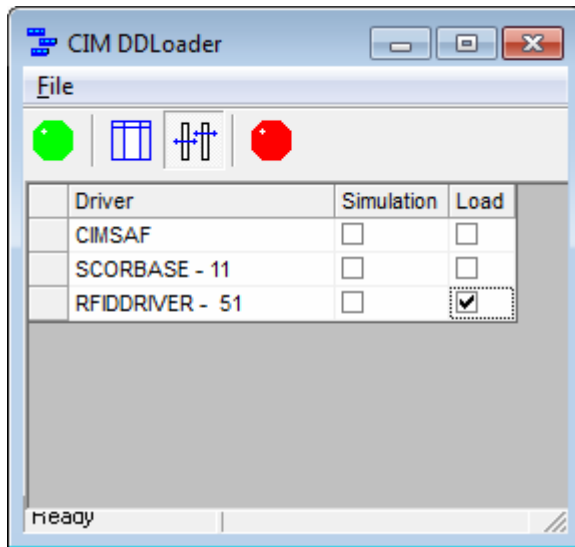
Figure 7-11: CIM DDLoader

b) Check the Load column of the RFIDDRIVER driver, and click the Load Selected

Drovers icon. The RFID Device Driver window is displayed.
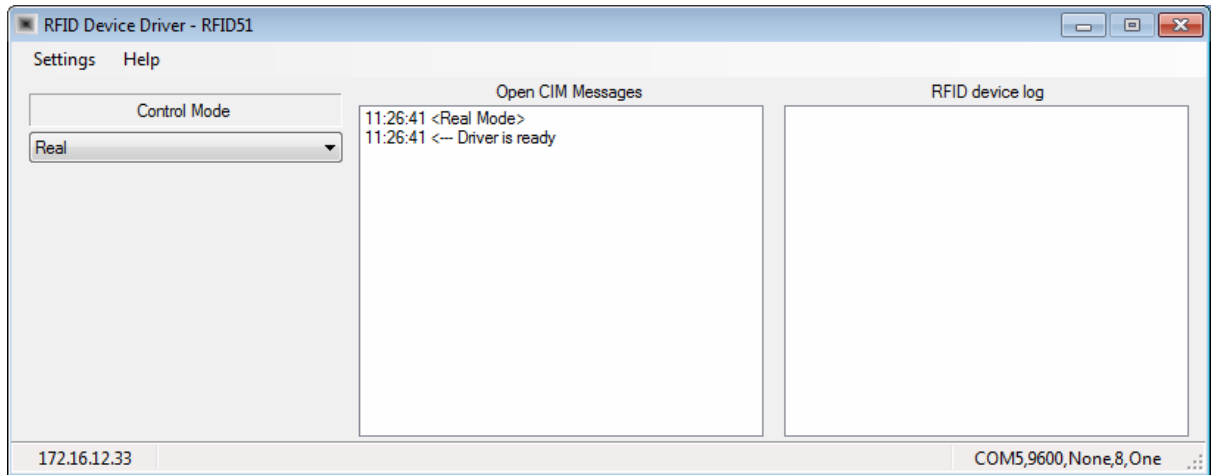
Figure 7-12: RFID Device Driver

c) From the Control Mode drop down list, select **Standalone** to switch to Standalone Mode.

d) Select **Settings | Template Lookup Table**. The Template Lookup Table is displayed.
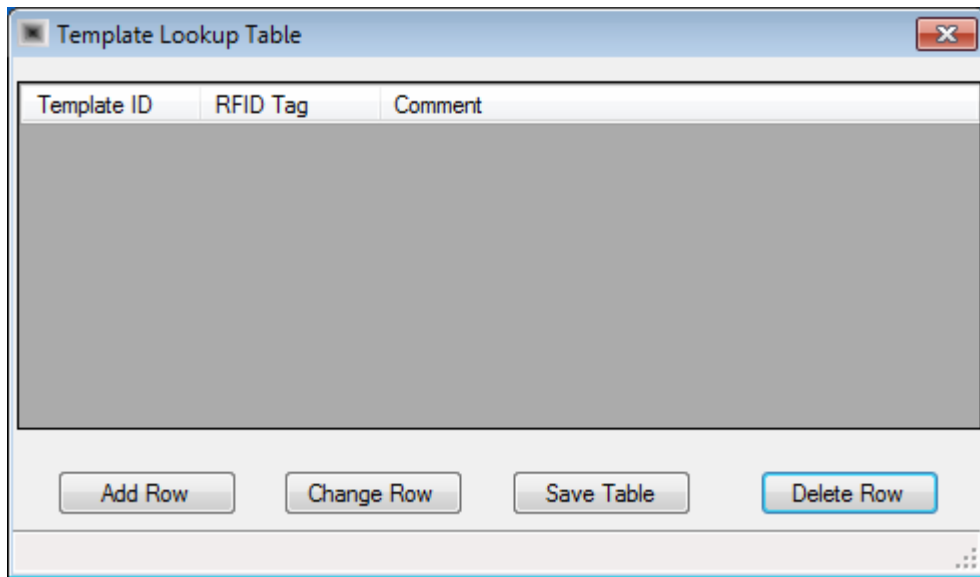
Figure 7-13: Template Lookup Table

e) Click **Add Row**. The Add Template Information window is displayed.

Figure 7-14: Add Template Information Window

f) Pass an RFID tag in front of the RFID reader. The tag's unique serial number is displayed in the RFID tag field.

Figure 7-15: RFID Tag in Add Template Information Window

g) Insert a template ID in the Template ID field, and optionally, a description in the Comment field.

The RFID tag comment only appears in the RFID Device Driver Lookup Table, and is intended for identification of the RFID tag (for example, Template 6).

Note

h) Click **Save**. An entry for the RFID tag is added to the lookup table.
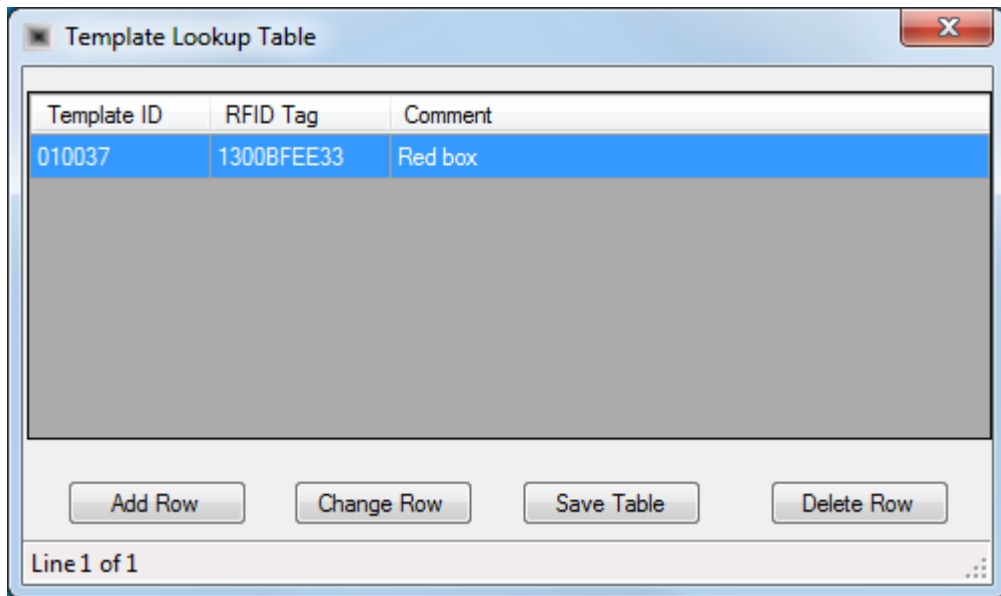
Figure 7-16: Template Lookup Table with Entry

i) To change the Template ID or Comment for any RFID tag, select the tag in the Template Lookup Table and click **Change Row**.

j) Click **Save Table** to save the new entry and close the Template Lookup Table.

k) Pass the RFID tag in front of the RFID reader and verify that the Template ID is included in the RFID tag entry in the RFID device log.
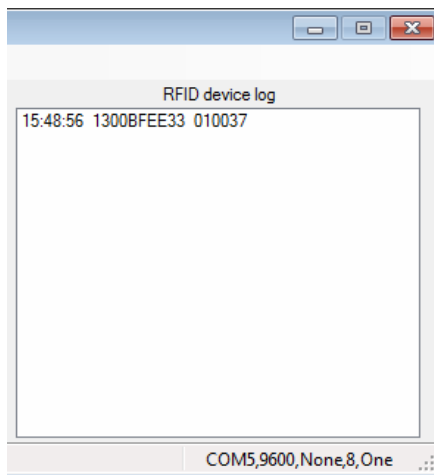
Figure 7-17: RFID Device Log Entry in RFID Device Driver

l) From the Control Mode drop down list, select **Real** to switch to Real Mode which is the normal operational mode.

Feeder Definition

Click **EDIT** in the leftmost column of the FDR row to display the Feeder Definition form:

Figure 7-18: Feeder Definition Form

During OpenCIM production operations, the message `Part not currently available` may be displayed. This indicates that the part feeder, or the ASRS, has run out of the required part.

Option	Description
Part Name	The names of all the parts that have been associated with the specified type of feeder, as defined in the Part Definition form.
Type	The number that identifies a certain type of feeder, as defined in the Virtual CIM Setup.
Capacity	The number of units of this type of part / material which can be placed into the feeder, as defined in the Virtual CIM Setup.
Quantity	The number of units currently loaded in the feeder. You must manually update the value of this field whenever you add parts to or remove parts from the feeder. The CIM Manager automatically updates this field during production.
💾	Exits and saves any changes.
🔴	Exits without saving any changes.

Rack Definition

Click **EDIT** in the leftmost column of the RACK row to display the Rack Definition form:
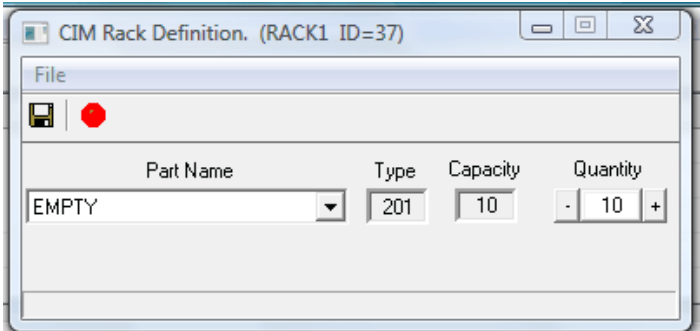
Figure 7-19: Rack Definition Form

Option	Description
Part Name	This list contains the names of all the parts that are associated with the specified type of feeder, as defined in the Part Definition form.
Type	The number that identifies a certain type of rack, as defined in the Virtual CIM Setup.
Capacity	The number of units of this type of part / material which can be placed in the rack, as defined in the Virtual CIM Setup.
Quantity	The number of units currently loaded in the rack. You must manually update the value of this field whenever you add parts to or remove parts from the rack. The CIM Manager automatically updates this field during production.
	Exits and saves any changes.
	Exits without saving any changes.

☞ *When editing the storage parameters you must click the create default storage* *icon*
Note *for the parts to be added.*

MRP

About MRP

Material Requirements Planning (MRP) enables manufacturers to calculate the material requirements from a list of items they intend to sell. MRP provides a tool for floor control, master production scheduling and capacity planning. Manufacturing Resource Planning (MRP II) coordinates and integrates manufacturing resources together with engineering, marketing and financial resources.

About OpenCIM MRP

The OpenCIM MRP program is used to create and define three types of orders:

- Customer Orders: products ordered
- Manufacturing Orders: items to be produced
- Purchase Orders: items to be purchased from suppliers

In general, OpenCIM MRP allows you to create a list of customers and define the products ordered by each customer. Once Customer Orders are created, the MRP program automatically creates a Manufacturing Order and a Purchase Order. You can view and modify or simply accept the Manufacturing Order, or define a completely new one. When the Manufacturing Order is submitted, the MRP creates an A-Plan file, or production work order. (For further details, see "A-Plan Report" later in this chapter.) In addition, the MRP creates a Purchase Order for items that must be supplied to the CIM. The OpenCIM Report Generator can be used to display and print the Purchase Order.

The following figure is a flow chart of the MRP program.
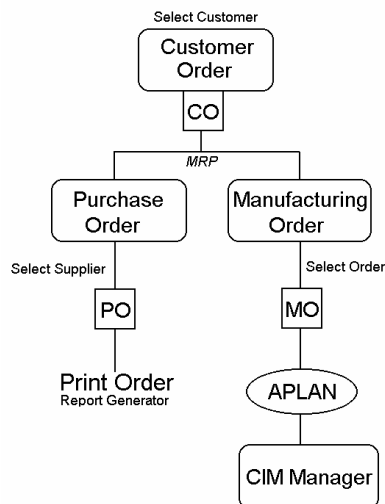
Figure 7-20: MRP Flow Chart

Customer Order Form

When you first activate the MRP, the Customer Order form appears. You can switch to the other order forms (Manufacturing and Purchase) by clicking the appropriate tab.

A customer order is a list of the parts (products) ordered by a customer. The Customer Order form shown below lets you create, view and modify a list of customers and their orders.

Parts must be defined in the Part Definition form before they can be ordered by customers.

Figure 7-21: Customer Order Screen

Toolbar

The following buttons apply only to the Customer Order table. Any changes you make using these buttons will not be stored in the database until you click **Save**.

Option	Description
	Adds a blank row to the Order table.
	Saves the selected Customer Order to the database.
	Automatically resizes the columns to accommodate long values (when the window is maximized).
	Adds a new customer to the customer list.
	Edits the information of the selected customer.
	Creates the selected customer order and saves it to the database.

✍ **Note:** *You can only order parts that have been defined as type Product or Supplied. Phantom parts cannot be ordered.*

Order List

Option	Description
Required Quantity	The number of units required by the customer.
Priority	The priority of this order (1–9). A priority of 1 is most urgent, 9 is least urgent. The CIM manager program uses this priority value to determine the sequence in which to produce orders. Different parts may have the same priority.
Due Date	The shipping deadline for the part. MRP will generate a Manufacturing Order and Purchase Order that will ensure completion of the part by the end of the day preceding the deadline.

> ✎ Note *In a commercial CIM environment, this deadline is normally expressed as a specific date and time. In an educational CIM environment, the Due Date is relative to the time an order is submitted. A relative time allows the same order to be resubmitted day after day without the need to edit the date field.*

How to Define a Customer

When you define a Customer, you are defining the name of the customer for whom a finished product or products will be made by the CIM cell.

❶
❷
❸
Procedure

Defining a New Customer

1. In the Customer Order form, click **New Customer**. The Customer Data box opens.

2. In the Customer Data box, fill in the name of the customer, and other customer information (e.g., address or phone number). Click **Save**. The box closes and this customer's name is added to the Customer List.

3. If you want to make any changes to specific customer information, select the customer from the list, click **Edit Customer** and make the desired changes. Note that the customer name cannot be changed in this option.

How to Define a Customer Order

When you define a customer order, you are defining the type and quantity of finished products for a specific customer.

❶
❷
❸
Procedure
Defining a Customer Order

1. Add a new customer order to the list by clicking **New Order**.
2. Right-click in the Customer column to open the customer list. Select the desired customer.
3. Right-click in the Part name column to open the part list. Select the desired part, for example: COVERED_BOX.
4. In the Required Quantity field, enter the number of units ordered by the customer, in this case 3.
5. In the Due Date field enter a value, in this case 2 (i.e., the part is to be completed in two days). This causes the MRP to instruct the CIM to begin production immediately.
6. Click Save to save the order for this customer.
7. To order more parts for the same customer in the same order, insert a new line by right-clicking on the last cell in the order and choosing **Insert After**. Repeat steps 2-6.
8. When you have finished filling in the order for a specific customer, create the order by clicking on **MRP**.

Manufacturing Order Form

A manufacturing order specifies the type and quantity of parts to be produced by the CIM cell on a specific day.

The Manufacturing Order form shown can be generated by the MRP program according to the customer orders currently in the system. You can view and modify or simply accept the Manufacturing Order, or define a completely new one.

You can define an order at any time, but you must finish defining all machine processes and subparts used in the order before you submit the order for production.

Each row in the Manufacturing Order table represents a total quantity of a particular part which needs to be manufactured on the specified date, so that all customer orders are filled.

Figure 7-22: Manufacturing Order Screen

Tool Bar

You can switch to the other order forms by clicking the appropriate tab.

The following buttons apply only to the Manufacturing Order table. Any changes you make using these buttons will not be stored on disk until you click **Save**.

Option	Description
▯	Adds a blank row to the Order table.
💾	Saves the selected Customer Order to the database.
⊨⊨	Auto-resizes the part list columns.
🖨	Prints the Manufacturing report.
🖨	Prints the A-Plan report for the last manufacturing order created. (For further details, see *A-Plan Report* section in this chapter).
M⃝	Creates the selected manufacturing order and its A-Plan, and saves them to the database.

The fields described below compose the Manufacturing Order table:

Option	Description
Part Name	The name of the products to be manufactured. This field corresponds to the Part field on the Part Definition form. Right-click in the Part Name column to open the product list.
	Note You can only order parts that have been defined as type Product. Products that have been defined as type Phantom or Supplied cannot be ordered.
Total Quantity	The total number of units ordered which must be manufactured on the specified day.
Initial Quantity	The number of parts to be extracted from the ASRS when production begins. The initial quantity is a number that can range from 1 (one) up to the value of the total quantity. Usually the value is 1 or 2. This field allows you to optimize the manufacturing process. (Refer to *Optimizing the Scheduling in OpenCIM* in Chapter 8, CIM Setup, for more details).
Priority	The priority of this order (1-9). A priority of 1 is most urgent, 9 is least urgent. The CIM Manager uses this priority value to determine the sequence in which to produce orders.
Due Date	The shipping deadline for the part, as generated by the MRP.
	Note In a commercial CIM environment, this deadline is normally expressed as a specific date and time. In an educational CIM environment, the Due Date is relative to the time an order is submitted. A relative time allows the same order to be resubmitted day after day without the need to edit the date field.

How to Create or Modify a Manufacturing Order

A Manufacturing Order is created automatically by clicking the MRP button in the Customer Order form.

The following procedure explains how to edit or create a Manufacturing Order.

❶
❷
❸
Procedure

Editing and Submitting a Manufacturing Order

1. Select the order date (number) from the Order List.

2. Click the desired row in the Part Name column and open the product list. Select the product required by the customer.

3. In the Total Quantity field, enter the number of items that need to be produced, in this case 5.

4. In the Initial Quantity field, enter the number of parts to be extracted from the ASRS when production begins, normally 1 or 2.

5. In the Priority field, enter 1 (indicating the highest priority).

   In the Due Date field enter a value which is one greater than the value in the Order List, in this case 2, indicating that the product will be manufactured today and ready for shipment tomorrow.

ᏰᏪ Note *To order other products at this time, add a new line by right-clicking the last cell in the order, choosing **Insert After**, and repeating steps 2-6.*

6. Click Save to save the entries without changing the last submitted production plan (A-Plan).

7. Click the MO icon to submit this order and create a new production plan.

You can now operate the CIM Manager and start production.

How to Submit an Order

Before you click MO and submit the manufacturing order, make sure that the following CIM definitions are up to date:

- Machines and Processes
- Parts
- Storage

After setting up these CIM elements, you can initiate production. You will receive an error message if you try to submit an order when any one of the following conditions exists:

- An undefined part is referenced on the Manufacturing Order form.
- An undefined subpart is referenced in a Part Process table.
- An undefined machine process is referenced in a Part Process table.

Purchase Order Form

A purchase order is a list of the parts that need to be supplied to the CIM cell so that it can complete the Customer Order.

The Purchase Order form shown below can be generated by the MRP program according to the customer orders currently in the system. You can view and modify or simply accept the Purchase Order, or define a completely new one.

The Purchase Order form lets you create, view and modify a list of suppliers.

Parts must be defined in the Part Definition form before they can be ordered from suppliers.

Each row in the Purchase Order table represents a total quantity of a particular part which needs to be purchased by a specified date, so that all customer orders are filled.

Figure 7-23: Purchase Order Screen

Toolbar

You can switch to the other order forms by clicking the appropriate tab above the table.

The following buttons apply only to the Purchase Order table. Any changes you make using these buttons will not be stored on disk until you click **Save**.

Option	Description
	Adds a blank row to the Purchase Order table.
	Saves the selected purchase order in database.
	Auto-resizes the part list columns.
	Allows you to add a new supplier to the Supplier list.
	Allows you to edit the information of the selected supplier.
	Creates the selected purchase order and saves it in the database.

The following fields compose the Purchase Order table.

Option	Description
Part Name	The name of the part you want supplied. This field corresponds to the Part field on the Part Definition screen.
	Note: You can only order parts that have been defined as type Supplied.
Quantity	The number of units you want to receive from the supplier.
Cost	The cost per unit, as defined in the Part Definition form.
Due Date	The date on which the part must be received from the supplier.
	Note In a commercial CIM environment, this deadline is normally expressed as a specific date and time. In an educational CIM environment, it is an advantage to have the Due Time relative to the time an order was submitted. A relative time allows the same order to be resubmitted day after day without the need to edit this field each time.
Send Date	The deadline for sending the Purchase Order to the supplier. Calculated by subtracting the time required by the Supplier (as defined in the Part Definition form) from the Due Date.

How to Define a Supplier

When you define a Supplier, you are defining the name of the supplier who will provide parts for the CIM cell.

❶
❷
❸
Procedure

Defining a New
Supplier

1. In the Purchase Order form, click **New Supplier** to open the Supplier Data box.

2. In the Supplier Data box, fill in the Name of the Supplier, and other Supplier information (e.g., address or phone number). Click **Save**. The box closes and this supplier's name is added to the Supplier List.

3. If you want to make any changes to specific supplier information, select the Supplier from the list, click **Edit Supplier** and make the desired changes. Note that you cannot change your name when editing supplier details.

How to Create or Modify a Purchase Order

The following procedure explains how to edit or create a Purchase Order.

❶
❷
❸
Procedure

Defining a Customer Order

1. Add a new purchase order to the list by clicking **New Order**.

2. Click the desired row in the Supplier column and open the supplier list. Select the name of the desired Supplier.

3. Click the desired row in the Part name column and open the part list. Select the desired part, for example: BOX.

4. In the Required Quantity field, enter the number of items ordered by the Supplier, in this case 3.

5. In the Due Date field enter a value, in this case 2 (i.e., the part is to be completed in two days). This causes the MRP to instruct the CIM to begin production immediately.

6. Click Save to save the order for this Supplier.

↪ Note *To order other products at this time, add a new line by right-clicking on the last cell in the order, choosing **Insert After**, and repeating steps 2-6.*

7. If you want to order more parts in the same order, add a new line by right clicking on the last cell in the order, choose **Insert After** and repeat steps 2-6.

8. After you have finished filling in the order for a specific customer, create the order by clicking **MRP**.

9. Click the PO button to activate the Report Generator, which will display or print the Purchase Order.

How to Send a Purchase Order

When you click PO, the MRP program prompts you to enter the date or dates for which you to print out Purchase Orders.

Since you do not want to send orders too far in advance (which will commit the purchase), and since you do not want to send a number of orders day after day, the MRP allows you to consolidate your orders for a period of time, say a week.

Once the dates are entered, the OpenCIM Report Generator menu appears on the screen. Refer to "Purchase Order Report" later in this chapter.

Optimization

The order of operation (timing), performed by CIM is controlled using the CIM optimizing mechanisms, which run concurrently and make decisions based on real-time situations in the work cell. You can manipulate the behavior of CIM by changing any one mechanism, or a combination of any of these optimizing mechanisms. This section describes the CIM Optimization Definition in CIM as well as the additional optimizing mechanisms, described in *Additional Optimization Methods in OpenCIM*, at the end of this section.

When activating OpenCIM, parts are dispatched from storage and placed in the queues to the various machines for processing. In certain cases some parts need to be processed in a number of different machines. The CIM Manager sorts these parts by creating a virtual queue of parts that are waiting to be processed in each machine, and the machine in turn always processes the first part in the queue. Optimization is performed using different methods for sorting the machine queue. For a general overview of optimization refer to *What is Optimization?,* as described in *Chapter 2, System Overview*.

Note — Even though each queue is managed separately for each machine, the performance is tested for the overall system.

CIM Optimization Definition

The CIM Optimization Definition enables users to select machine queue algorithms and define their weight. Users can then observe the effect of the different algorithm combinations on the overall system performance.

The results generated from the CIM Optimization Definition are displayed in the CIM Performance Analysis window as described in *Performance Analysis*.

The CIM Optimization Definition window is displayed by selecting **Utility Programs | Optimization Definition** from the OpenCIM Manager main window, and appears as follows:
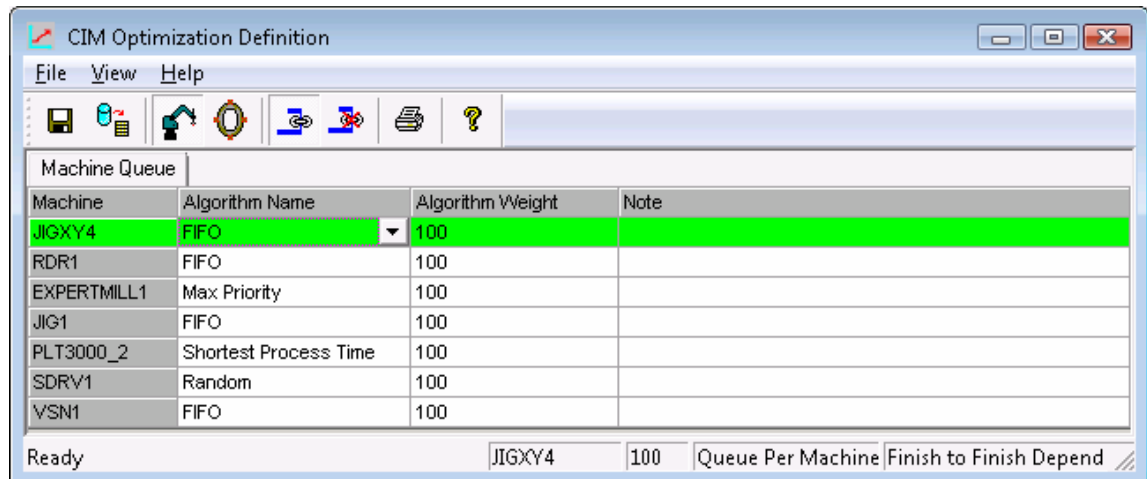
Figure 7-24: CIM Optimization Definition

CIM Optimization Definition Window

Main Menu

Option	Description
File	Contains the following file options: Save and Refresh (described in the toolbar options below). The Print, Preview and Print Setup options will be defined later on.
View	Contains the following toggle view options: Toolbar, Status Bar
Help	Displays the online help.

Toolbar

Option	Description
	Save: Saves the Machine Queue form to the database .
	Refresh: Refreshes the Machine Queue form to the entries that were defined before the last save.
	Queue Per Machine: Displays the list of machines and their alogirithms, enabling you to select a different algoritm and weight for parts in the queue for each machine.
	Queue Per System: Displays the system algorithm, enabling you to select the same alogorthim for parts in the queue for all the machines in the system.
	Enable Dependancy: Enables finish-to-finish dependency. Meaning, the algorithms will plan the manufacturing process, to enable all tasks to finish together (for the overall system). This is performed, for example, by starting the manufacturing process with the part containing the longest process time and ending with the part containing the shortest process time.
	Disable Dependancy: Disables finish-to-finish dependancy.
	Print Optimization: Displays an optimization report.

Machine Queue Form

Figure 7-25: Machine Queue Form Tab in CIM Optimization Definition Window

The Machine Queue Form contains the following fields:

Column	Description
Machine	Contains a predefined list of all the machines defined in the CIM Setup.
Algorithm Name	The name of the algorithm defined for the parts that are in the queue to the selected machine. You can select the required algorithm from the dropdown list, as follows:

- **FIFO (First in First Out):** Parts are processed according to first in first out. Meaning, the parts that arrive first in the queue are processed first.
- **Maximum Priority**: Parts are processed according to their priorities (**1** through **10**) that were defined in the CIM MRP window. Meaning, the parts with the highest priority (such as, **1**) will be processed first.
- **Random**: Parts are processed based on a random selection basis.
- **Shortest Process Time**: Parts are processed according to their process time period. In this case, the parts with the shortest process time will be processed first.

Column	Description
Algorithm Weight	Enables you to enter the weight of the selected algorithm (the total weight of all algorithms must be 100).
Note	Enables you to enter a text comment for reference purposes.

Status Bar

The status bar, shown above, contains the following information:

Option	Description
JIGXY4	The current machine for which the queue is defined.
100	The total algorithm weight.
Queue Per Machine	Defines the current queue characterization, as follows:

Option	Description
	• **Queue Per Machine**: Displayed when the button is selected from the toolbar. The queue is defined specifically for each machine.
	• **Queue Per System:** Displayed when the button is selected from the toolbar. The queue is defined for the whole system.
Finish to Finish Dependency	Defines whether or not finish-to-finish dependancy is activated, as follows:
	• **Finish to Finish Dependency:** Enables process dependency, displayed when the button is selected. Meaning, the algorithms will plan the manufacturing process, to enable all tasks to finish together (for the overall system).
	• **No Finish to Finish Dependency:** Disables part dependency, displayed when the button is selected.

Optimization Right-Click Menu

PLT3000_2 / FIFO

Insert Row Before
Insert Row After
Delete Row
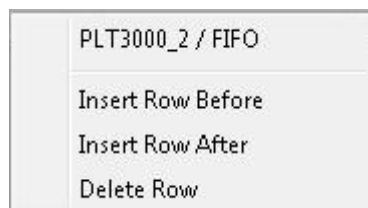
The status bar, shown above, contains the following information:

Option	Description
Insert Row Before	Enables you to insert a new alogrithm before the selected algorthim.
Insert Row After	Enables you to insert a new alogrithm after the selected algorthim.
Delete Row	Enables you to delete an existing algorithm.
	↪ **Note:** A minimum of one algorithm exists per machine. When only one algorithm exists, the system does not enable you to delete it.

Defining Algorithms

The following procedure explains how to define an algorithm in the Optimization Manager and then run the CIM cell to view the virtual queues of parts that are waiting to be processed in each machine.

❶
❷
❸
Procedure
Defining Algorithms

1. From the CIM Project Manager main window, select a project for which you want to define algorithms.
   For example, **CIM-04-MQ**.

2. From the Project Manager toolbar click the 🔵 CIM Manager icon. The CIM Manager main window is displayed.

3. Select **Utility Programs | Optimization Definition**. The CIM Optimization Definition window is displayed.
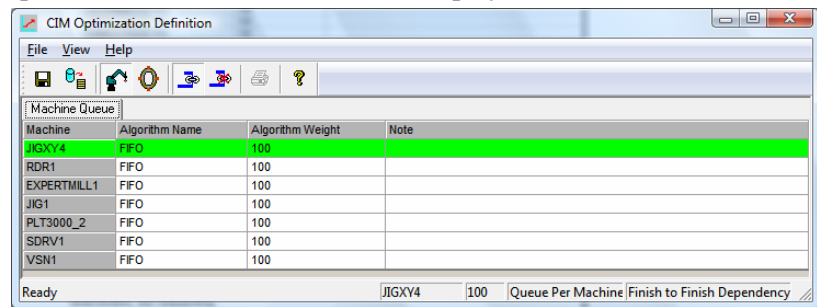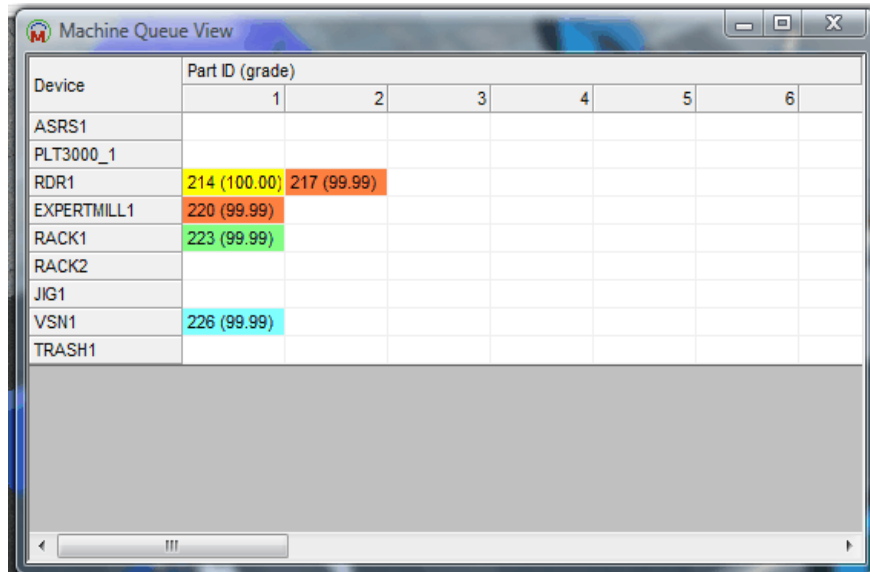
Figure 7-26: Defining Algorithms in the CIM Optimization Definition Window

4. Ensure the 🔧 Queue per Machine icon is selected (as shown above) displaying the queue per machine.

5. In the Machine Queue tab select the required algorithm for each machine from the **Algorithm Name** dropdown list.

6. Enter the required weight in the **Algorithm Weight** field.

7. If required you can add additional algorithms for a specific machine, as follows:

   • Right click the algorithm. A right-click menu is displayed.

   • Select **Insert Row Before** or **Insert Row After** to add an algorithm, and then select the required algorithm from the **Algorithm Name** dropdown list.

   • In the **Algorithm Weight** column, enter the weight for each algorithm (ensuring the total algorithm weight per machine is 100.)

   ☞ **Note:** The curent sum of the algorithms is displayed in the status bar.

❶
❷
❸
Procedure

Defining Algorithms

8. Click **Save** and then select **File | Exit** to close the Optimization Manager.

9. From the CIM Manager window, select **Utility | Part Definition**. The part definition window is displayed. For each part select a different color. This will enable you to observe the flow of parts according to their color in the manufacturing cycle. Close the Part definiton window. Refer to the *Part Definition* section in this chapter for further details

10. Select **Utility | MRP**. The CIM MRP window is displayed. In the **Customer Order** tab, select the parts, define their quantities and then select ![MRP icon]. Click the **Manufacturing Order** tab, select the order to process and click ![MO icon] and then click **OK**. Refer to the *MRP* section in this chapter for further details. Close the CIM MRP window.

11. From the CIM Manager window click the **Start** ![power button] button. Click **OK** at the confirmation message and then click ![play button] and observe the CIM manufacturing cycle.

12. Select **Window | Machine Queue**. The Machine Queue View window is displayed.

Figure 7-27: Machine Queue View

You can now view the parts which are placed in the machine queues based on the algorithms that you selected in the Customization Manager window in step 3.

Additional Optimization Methods in OpenCIM

Further to the Optimization Manager, described in the previous section, additional optimization mechanisms (and their examples) that exist in OpenCIM are described below:

- In each manufacturing order line (in the order definition), for each type of part you can decide how many parts of this type will be released from storage at the beginning of the manufacturing process in order to fill the buffers. This number is referred to as "Initial Quantity".

- The release time of each additional part from the ASRS is not set in terms of time, but rather in terms of the work (part) progress. For example, an additional similar part is fed to the system only when the previous part has reached a certain stage in its production plan. This stage is marked by adding the command NEXT in the definition of the part production process. The default used for a production plan is that the system will begin to feed the next part after the last manufacturing process defined for each part. The default can be changed by adding the command NEXT in the Process column in the Part Definition form.

Example

This example demonstrates how to use the NEXT command to set the release time of the parts from storage.

We will examine part P1, whose manufacturing process requires the use of three machines, M1, M2, and M3, and the process time of each machine is 2, 7 and 3 minutes respectively.

The "schedule" for the part is represented as follows (neglecting the transfer times between the machines):
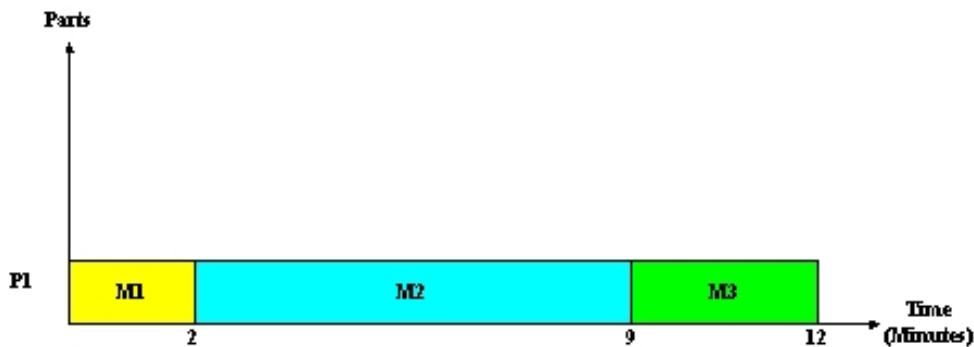
Figure 7-28: Schedule for Part

If a new part is released every time a part finishes its process at M1, there will be an accumulation of parts in front of M2, due to its longer process time. Alternatively, if a new part is released every time a part finishes its process at M2, no queues will be created at any place in the system, since the process at M2 is the longest. Also, releasing a new part only after the end of the last process defined for the part will prevent an accumulation of parts.

The problem in this case is that machine M1 will remain idle until the end of the part process in M2, and even worse, M2, which creates the bottleneck in this example, will remain idle in between parts. The solution is to release more than one part the first time, in order to fill the buffers for each of the machines.

In our example, releasing two parts at the beginning of production, and releasing one additional part each time a part finishes its process at M2, will give maximal throughput of the production line (since machine M2, which forms the bottleneck, will always be busy).This also minimizes the in-process stock and leaves a free time-slot for machines M1 and M3 to work on other parts with a lower priority from the same production process. It can then be seen that the location of the NEXT command immediately after the longest process results in maximal utilization of the system. On the other hand, locating the NEXT command at the end of the production process will result in a system which uses more parts in the buffer, thus continuing to deliver good throughput even in cases of failure, inaccurate data or a combination of simultaneous production of different types of parts having different priority levels.

The timing for parts of different types (it is possible that they share the same machines for part of their production process) uses the same mechanism for each part, and in addition uses the machine queue mechanism, in order to decide which part will be processed first in a certain machine. As a simple example, the machine queue mechanism will choose a part having a higher priority level.

In the following figure, a further example of the process scheduling is given (the same example but with a different level of detail). Part P1 is the part described above, and part P2 is processed by machines M1 (3 minutes process time) and M4 (2 minutes process time). The order included four parts of each type and the initial quantity of both parts is two (2). Part P1 is defined with a higher priority than part P2. The user did not use the NEXT command for either part, so the system is using the default of putting the NEXT command after the last process defined for each part.
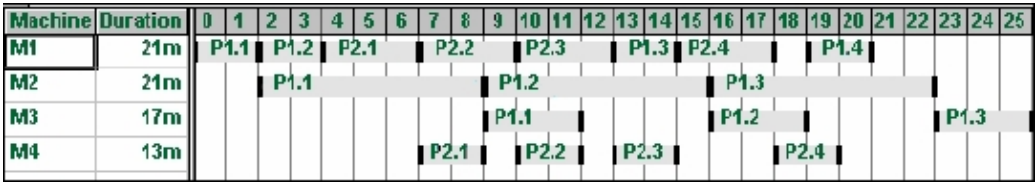
Figure 7-29: Process Scheduling

> P1.1 is the first part of type P1.
> P1.2 is the second part of type P1.
> P2.1 is the first part of type P2.
> P2.2 is the second part of type P2.

Time	Machine	Description
0	M1	CIM Manager invoked. 2 parts of type P1 and 2 parts of type P2 were sent from storage. M1 selects 1 of the 4 parts based on priority level (P1.1) to begin processing.
2	M1	Finished processing P1.1 and selects one of the 3 parts remaining in the buffer (based on the highest priority level) to begin processing (P1.2).
	M2	P1.1 is sent from M1, for processing.
4	M1	Finished processing P1.2 and selects from the queue, based on priority level one of the two remaining parts (P2.1) to begin processing.
	M2	P1.2 is sent from M1 to wait in the buffer of M2.
7	M1	Finished processing P2.1. Part P2.2 is selected from the queue to begin processing.
	M4	P2.1 is sent from M1 to begin processing.
9	M2	Finished processing P1.1 and begins to process P1.2.
	M3	P1.1 is sent from M2 to begin processing.
	M4	Finished processing P2.1 (the last process) so, the NEXT command is performed and P2.3, is waiting on the buffer of M1
10	M1	Finished processing P2.2 and P2.3 is selected from the queue to begin processing.
	M4	P2.2 is sent from M1 to begin processing.
12	M3	Finished processing P1.1.
	M4	Finished processing P2.2. The NEXT command is activated and part P1.3 is waiting on the buffer of M1.
13	M1	Finished processing P2.3 and P1.3 is selected from the queue to begin processing.
	M4	P2.3 is sent from M1 to begin processing.
14	M1	The NEXT command is performed, so P2.4, is waiting on the buffer.
15	M1	Finished processing P1.3 and P2.4 is selected from the queue to begin processing.
	M2	P1.3 is sent from M1 to wait in the buffer of M2.
	M4	Finished processing P2.3. The NEXT command is activated and P2.4 is waiting on the buffer of M1.
16	M2	Finished processing P1.2 and P1.3 is selected from the queue to begin processing.
	M3	P1.2 is sent from M2 to begin processing.
18	M1	Finished processing P2.4. The machine remains idle because there are no parts waiting in its queue.
	M4	P2.4 is sent from M1 to begin processing.

Time	Machine	Description
19	M1	Receives P1.4 which was released from the ASRS as a result of the NEXT command and begins processing.
	M3	Finished processing P1.2. The NEXT command is performed, so P1.4, is released from the ASRS and sent to M1 for processing. You can eliminate the inactive time period (Time 18) by increasing the initial quantity for part P1.
20	M4	Finished processing P2.4. The NEXT command is not performed because all 4 parts of type P2 are ready.
21	M1	Finished processing P1.4.
23	M2	Finished processing P1.3 and P1.4 is selected from the queue to begin processing.
	M3	P1.3 is sent from M2 to begin processing.
26	M3	Finished processing P1.3. The NEXT command is not performed because all four parts of type P1 are being, or have been processed.

Note *The system works in parallel on orders with both high and low priority, taking into consideration that orders with high priority are treated first.*

Benefits of the Optimization Approach

The Optimization Approach offers the following benefits:

- The system continues to follow the priority you define for each part even though the system is required to work on many parts, from different priority levels.

- The Optimization Approach handles incomplete or incorrect predicted process time in a competent manner (i.e. the NEXT command is actually executed when the machine finishes processing the part and not according to some pre-calculated time).

- The Optimization Approach implemented in the CIM environment can handle different combinations of parts, in different quantities and priority levels that need to be produced in an efficient manner.

- In the example illustrated above the transmission time was neglected. However, the OpenCIM system still takes the transmission time into account through the use of its optimization mechanisms. This can become a significant point when CIM systems have short process times and the transmission time cannot be overlooked.

- Machines, robots, storage locations and even conveyors have their own priority queue which you can control in order to increase the performance of the production schedule (e.g. in the example given above, we assumed that all four parts were sent to machine 1 simultaneously because we ignored the transmission time. However, in the real CIM, the optimization mechanisms ensure that machine 1 will work on the part with the highest priority level because the ASRS will know to first select and send the part with the highest priority level from the four parts that it was ordered to release by the CIM Manager).

The Optimization Approach is completely distributed. Each machine can continue to work independently as long as there are parts in its queue. Computational overload does not occur on any of the station PCs+, even with very large CIM cells.

Performance Analysis

The Performance Analysis utility in OpenCIM enables users to analyze the impact of different algorithm combinations on the system performance. You can use this utility to view, print and analyze the manufacturing cycle data to improve system performance, such as shorten the production time and as a result improve efficiency and lower the production costs.

The data in this utility is generated according to the definitions in the CIM Optimization Definition, as described in *Optimization.*

*When activating the CIM Performance Analysis utility, ensure that the simulation speed has been set to **1** in the Modes window.*

Note

CIM Performance Analysis

The CIM Performance Analysis enables you to view information that was generated from the last manufacturing cycle in the system and then save it for comparison and backup purposes. You can then view a summary of data comparing the different previously saved manufacturing cycles. In addition, you can print the currently displayed performance report and the corresponding optimization report (as displayed in the CIM Optimization Definition).

The CIM Performance Analysis window is displayed by selecting **Utility Programs | Performance Analysis** from the OpenCIM Manager main window, and appears as follows:

Run ID	70					
Total Run Time	00:36:40					
Note	Last Run Results					
Machine	Total Process Time	% Efficiency	Max Queue Length	Production Cost	# Setups (CNC Only)	% Failures (QC only)
JIGXY4	00:00:00	0.00	0	0.00	0	0.00
RDR1	00:00:00	0.00	0	0.00	0	0.00
EXPERTMILL1	00:00:55	2.50	1	0.00	1	0.00
JIG1	00:00:10	0.45	2	0.00	0	0.00
PLT3000_2	00:00:00	0.00	0	0.00	0	0.00
SDRV1	00:00:00	0.00	0	0.00	0	0.00
VSN1	00:00:55	2.50	1	0.00	0	0.00
System Summary	00:02:00	1.82	2	0.00	1	0.00

Figure 7-30: CIM Performance Analysis Window

CIM Performance Analysis Window

Main Menu
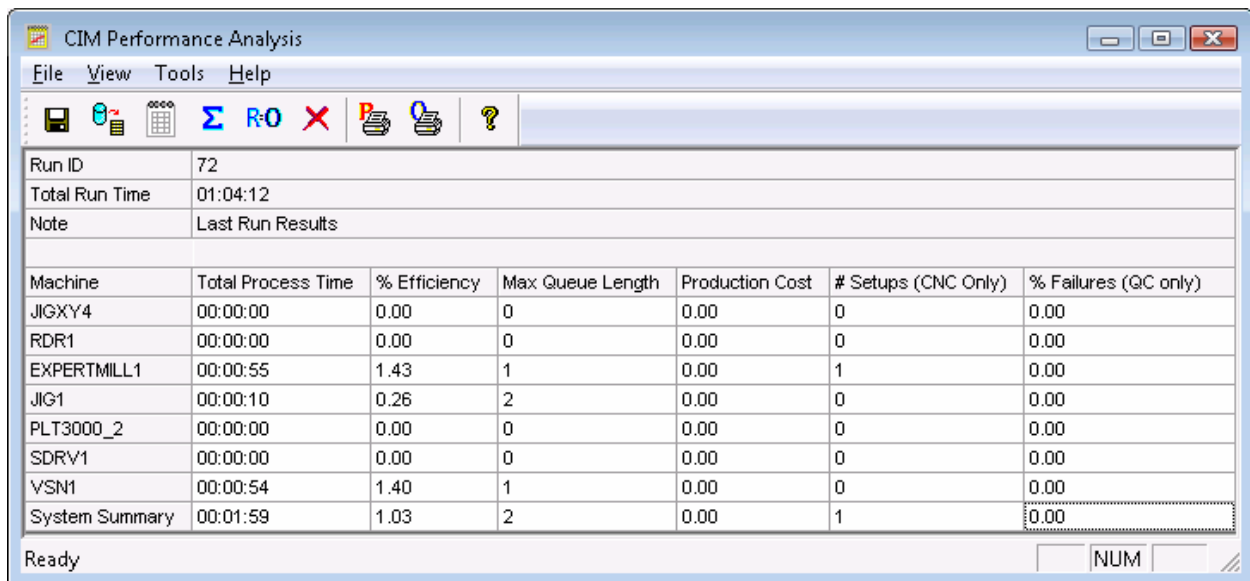
Option	Description
File	Contains the following file options: Save, Refresh, Delete All, Print Performance, Print Optimization, Exit. Each of these options are described in the *Toolbar* section below.
View	Contains the Toolbar toggle view option.
Tools	Show Single, Show Summary, Reset Run ID. Each of these options are described in the *Toolbar* section below.
Help	Displays the online help.

Toolbar

Option	Description
	Save: Saves the current manufacturing cycle run performance report to the database, using the Optimization settings defined in the CIM Optimization Definition. The Save Report window is displayed, enabling you to add the cycle's description and save it to the **PERFORMANCE_X.DBF** file. (The system also saves the corresponding optimization file.)
	Refresh: Refreshes the view to display the results from the last manufacturing cycle run.
	Show Single: Displays the Choose Performance File window, enabling you to select and view a previously saved **PERFORMANCE_X.DBF** file. Information per machine includes, total process time, efficiency, Max Queue Length, Production cost and more. For further details, refer to the *Single Manufacturing Cycle Performance Table*, described in the next section.
	Show Summary: Displays a summary of all the previously saved manufacturing cycles. Information per cycle includes, Total Process Time, Efficiency, Max Queue Length, Production Cost and more. For further details, refer to the *Summary of Manufacturing Cycle Performance Table*, described in the next section.
	Reset Run ID: Resets the run ID number to 0. This option is used for example, when a new Manufacturing Order is created.
	Delete All: Deletes all previously saved reports from the database.
	Print Performance: Prints the currently displayed performance report.
	Print Optimization: Prints the corresponding optimization report as displayed in the CIM Optimization Definition.

Manufacturing Cycle Performance Table

The CIM Performance Analysis enables you to view the results of the last manufacturing cycle and save it for future reference. The results include the process time, the efficiency per machine and per system the number of failures that were detected and so on.

Figure 7-31: Manufacturing Cycle Performance Table

The Manufacturing Cycle Performance Table contains the following information:

Column	Description
Machine	Contains the list of machines that were defined in the CIM Setup.
Run ID	Contains nutsthe ID number of the manufacturing cycle.
Total Run Time	The time period of the manufacturing cycle.
Note	The manufactuirng cycle's description.
Total Process Time	The total process time performed on a specific machine, as well as the system summary which is the total process time of all the machines in the cycle .
% Efficiency	The efficiency of each machine in the cycle, as well as the system summary which is the efficiency of all the machines together. Machine efficiency is defined as the process time divided by the total manufacturing time of the machine.
Max Queue Length	The maximum number of parts that existed in the machine queue during the manufacturing cycle.
Production Costs	The production costs per machine and per system. Production costs per machine is defined as the process time multiplied by the cost per hour (defined in *Machine and Process Definitions*).
# Setups (CNC Only)	The number of setups that exist in the CNC machine per manufacturing cycle. A new setup is created each time the CIM Manager instructs the CNC machine to load a new program. A new program is required when a new type of part is required for proccessing in the CNC machine.
% Failures (QC Only)	The number of part failu res that were detected in the QC device.

Summary of Manufacturing Cycle Performance Table

The CIM Performance Analysis enables you to view a summary of the different manufacturing cycles that were saved in the system. This enables you to compare the results of the cycles, such as the process time, the efficiency, the amount of failures and so on.

Figure 7-32: Summary of Manufacturing Cycle Performance Table

The Summary of Manufacturing Cycle Performance Table contains all the information fields that were described in the previous section (total process time, efficiency, max queue length, production cost and more).

Creating Manufacturing Cycle Performance Reports

The following procedure explains how to create manufacturing cycle performance reports for backup and retrieval purposes.

❶
❷
❸
Procedure

Creating Manufacturing
Cycle Performance
Reports

1. From the CIM Project Manager main window, select a project for which you want to define algorithms.
   For example, **CIM-04-MQ**.

2. From the Project Manager toolbar click the CIM Manager icon. The CIM Manager main window is displayed.

3. Select **Utility Programs | Optimization Definition**. The CIM Optimization Definition window is displayed, as shown in the *CIM Optimization Manager*.

4. In the Machine Queue tab select the required algorithm for each machine from the **Algorithm Name** dropdown list, as described in *Defining Algorithms.*

5. After defining the algorithms for the manufactuirng cycle, the next step is to create a manufacturing order, as follows:

   - Select **Utility Programs | MRP**. The CIM MRP window is displayed. In the **Customer Order** tab, select the parts, define their quantities and then select .

   - Click the **Manufacturing Order** tab, select the order to process and click and then click **OK**.
     Refer to the *MRP* section in this chapter for further details.

   - Close the CIM MRP window.

6. After creating the manufactuirng order, the next step is to run the manufacturing cycle, as follows:

   - From the CIM Manager window click **Start** , and click **OK** at the confirmation message.

   - Click **Run** and observe the CIM manufacturing cycle.

   - When the manufacturing order is complete click **Stop** .

7. The next step is to view the performance data.. Select **Utility Programs | Performance Analysis**. The CIM Performance Analysis is displayed showing the performance data from the last manufacturing cycle.

❶
❷
❸
Procedure

Creating Manufacturing
Cycle Performance
Reports

8. Click **Save**. The Save Report window is displayed.
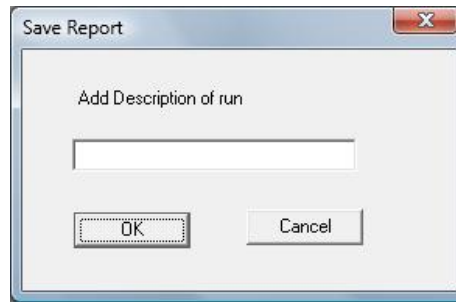
Figure 7-33: Save Report Window

9. Enter a description of the manufacturing cycle and click **OK**. A message is displayed informing you of the performance file name, such as **Performance_5.DBF** for example.

10. Click **OK** to save the file to the database for retrieving later.

&✎ **Note:** You can click **Show Summary** Σ to display the Summary of Manufacturing Cycle Performance Table, as shown in the previous section.
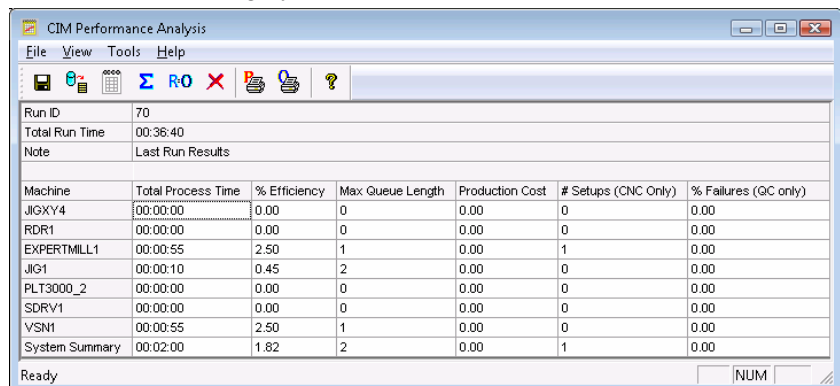
Viewing Manufacturing Cycle Performance Reports

The following procedure explains how to view predefined manufacturing cycle performance reports. You can view a single manufacturing cycle report or a summary of all the manufacturing cycle reports that exist in the system for comparison purposes.

❶
❷
❸
Procedure

Viewing the
Manufacturing Cycle
Performance Table

1. From the CIM Project Manager main window, select a project for which you want to define algorithms. For example, **CIM-04-MQ**.

2. From the Project Manager toolbar click the CIM Manager icon. The CIM Manager main window is displayed.

3. Select **Utility Programs** | **Performance Analysis**. The CIM Performance Analysis window is displayed containing the results from the last manufacturing cycle run.

Figure 7-34: CIM Performance Analysis

4. Click **Show Single** . The Choose Performance File window is displayed.

Figure 7-35: Choose Performance File Window

5. Select the required performance file from the list and click **Open**. The Manufacturing Cycle Performance Table of the selected file is displayed in the CIM Performance Analysis window.

If required, you can print the required performance and optimization reports as follows:

- Click **Print Performance** to print the currently displayed performance report.

- Click **Print Optimization** to print the corresponding optimization report as displayed in the CIM Optimization Definition.

Reports

The snapshots shown below are all taken from the CIM-08-A-DEMO project provided in OpenCIM installation.

OpenCIM provides a powerful, yet flexible report generator. This utility program allows you to view and print information from the various OpenCIM databases. You can access ten types of predefined reports, or you can create your own user-defined reports. The predefined reports that can be generated are shown in the following figure:
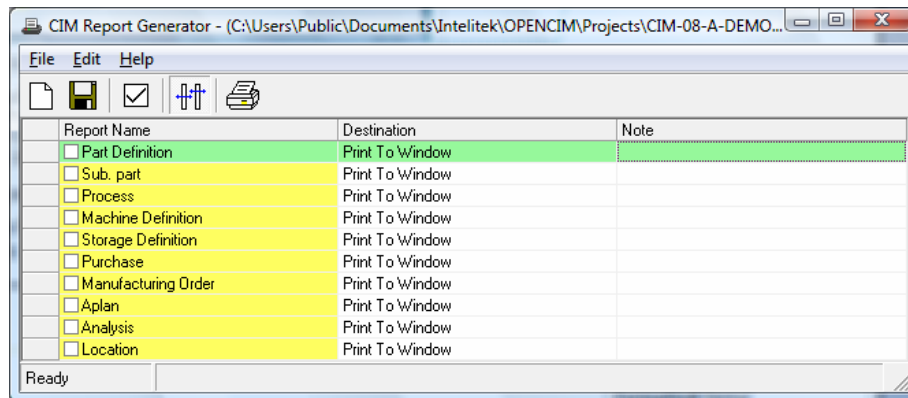
Figure 7-36: OpenCIM Report Generator Dialog Box

The following procedure details the steps involved in generating a report.

❶
❷
❸
Procedure

Generating a Report

1. Select Report Generator from Utility Program in CIM Manager Main Menu; the OpenCIM Report Generator dialog box appears.

2. Select the requested report.

3. Click the Print button. The desired report will appear on the screen. If you want to print it on a printer, click the printer button on the report screen.

Part Definition Report

The Part Definition Report is generated from information that was entered in the Part Definition form. It shows the names and description of all parts used by the CIM cell. The following is an example of a Part Definition Report.

Each of the columns in the Part Report relates to a specific field in the Part Definition form, as follows:

Part Report	Part Definition Form (Field)
#	Part # as listed in sequential order.
Part Name	Part Name
Type	Part Type: supplied, product or phantom.
Part ID	Part ID
Template ID	Template Type
Part Description	Description

Subpart Report

The Subpart Report is generated from information that was entered in the Part Process Table in the Part Definition form. The Subpart Report is a Bill of Material. It shows all the subparts which comprise the finished product. The following is an example of a Subpart Report.

Each of the columns in the Subpart Report relates to a specific field in the Part Definition form.

Subpart Report	Part Process Table (Field)
Part Name	Part Name.
Sub-Part Name	The column **Subpart** in the Part Process Table.
Manufacturing Process Name	The column **Process** in the Part Process Table.
Manufacturing Parameters	The column **Parameters** in the Part Process Table for each corresponding process for a particular subpart.

Manufacturing Order Report

The Manufacturing Order Report displays all production orders for a particular date.

The report is generated from the information that was entered in the Manufacturing Order form. The following is an example of a Manufacturing Order Report.

Each of the column headings in the Order Report relates to a specific field in the Manufacturing Order form, as follows:

Manufacturing Order Report	Manufacturing Order Form
Part Name	The column "Part". There can be more than one part listed.
Total Number of Parts Ordered	The column "Total Qty". Each Total Qty listed corresponds to a specific part ordered.
Initial Number to Produce	The column "Initial Qty". Each Initial Qty listed corresponds to a specific part ordered.
Priority	The column "Priority". The Priority level (from 1 - 9) listed corresponds to a specific part ordered.
Final Storage Location (if not ASRS)	Refers to the final storage location listed in the column "Note" (for a specific part).

Machine Report

The Machine Report lists the names of all machines in the OpenCIM cell. This report is generated from the information that was entered in the Machine Definition form. The following is an example of a Machine Report.

Each of the columns in the Machine Report relates to a specific field in the Machine Definition form, as follows:

Machine Report	Machine Definition Form (Field)
#	The sequential number of the machine as listed.
Machine Name	Machine Name
Cost Per Hour	Cost Per Hour, in the Machine Process table.
Maximum Number of Preloaded Programs	Max Preloaded Programs
Program 1 Program 2 Program 3	List of Preloaded Programs

Process Report

The Process Report shows the user-defined name (Process Name field) of each machine in the OpenCIM and the processes performed by the machine. This report is generated from information that was created from the Machine Process Table in the Machine Definition form. The following is an example of a Process Report.

Each of the columns in the Process Report relates to a specific field in the Machine Definition form, as follows:

Process Report	Machine Process Table (Field)
#	The sequential number of the machine as listed.
Machine Name	Machine Name.
Process Name	The column "Process" in the Machine Process Table.
Process Type	The column "Action Type" in the Machine Process Table.
Program File Name	The column "File" in the Machine Process Table.

ASRS Report

The ASRS Report shows the contents of the ASRS. It is generated from information that was entered in the Storage Definition form. The following is an example of an ASRS Report.

Each of the column headings in the ASRS Report relates to a specific field in the Storage Definition form, as follows:

ASRS Report	Storage Definition Form (Field)
Name	Name of storage location
Index	The number displayed in parentheses below the ASRS grid; e.g., ASRS (15). This is an internal index used in communication between the CIM Manager and the ASRS robot controller (and not the Index of the graphically displayed ASRS cell.)
Part Name	The name of the part residing in the current storage cell as defined in the Part Definition form (refer to cell in grid).
Part ID	Part ID, as defined in the Part Definition form.
Status	Status of the storage cell (Empty, Empty Template or Part on Template).
Template Number	The six-digit template number.

Analysis Report

The Analysis Report is detailed information on the status of the entire system and is geared for the more experienced user. The report contains a Log file summary of each action. See below for an example of an Analysis Report.

The following is a description of each of the columns in the Analysis Report:

Heading	Description
Part Name	The name of the part as defined in the Part Definition form or in the Virtual CIM Setup.
Device	The name of the robot, machine or conveyor that performs the operation, as defined in the Machine Definition form.
Action	Robot: (pick-and-place)
	Assembly: (base and pack)
	Conveyor: (deliver) or machine (the name of process as defined in the Machine Process Table).
Subpart Name	Robot: the number of a template or the name of a part.
	Machine: the name of a part or material.

Target	Where the process should be performed.
Index	Indicates the exact location on a device which has more than one location for a part.
Status	The status of the action .
Time	The time the action started or the time the action was completed.

Each line in the Analysis Report will have one of the following status reports:

- **Activate**: CIM Manager has determined that a command can be sent to a machine (e.g., a robot or CNC machine).

- **In Process**: command has been sent to the machine; for example: GET part from device A (source) and PUT part in device B (target).

- **Start**: Operation has started. Source device may now receive its next command (i.e., another "Activate"). Used, for example, to notify CIM Manager that a pallet can be released from a conveyor station.

- **Finish**: Operation has been completed. Target device may now receive its next command (i.e., another "Activate").

- **Done** (End): The machine is now ready to receive its next command.
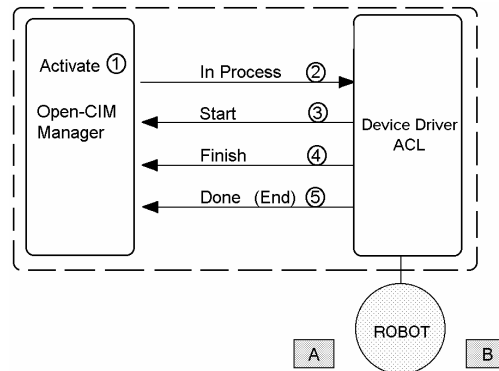
Figure 7-37:   Analysis Report Flow Chart - Example for ACL Device Driver

Location Status Report

The Location Status Report is a detailed listing of every location defined in the CIM system. This report is used by the system administrator or the more experienced user for debugging the system.

There is more than one type of location.

- Locations are defined in the Virtual CIM Setup. Every ASRS compartment, every station on the conveyor, places on a buffer and places inside the machine.

- Every template, a part of an assembly (e.g. one part is on top of another part), a gripper on a robot, every part in the system.

The Location Status Report enables you to know exactly what and where something is in the system at a given time. The following is an example of a Location Status Report.

The following is a description of each of the columns in the Location Status Report :

Heading	Description
Location	The name of the location.
ID	The location ID (numeric) as defined in the Virtual CIM Setup.
Index	Indicates the exact location on a device which has more than one location for a part..
Part Name	The name of the part in this location as defined in the Part Definition form.

	Status	Status of the specified location (Empty, Part on Template, Empty Template or Part).
	Template Number	The number of the template at this location.
	Type	The type of device as defined in the Virtual CIM Setup.

A-Plan Report

The A-Plan Report is a detailed description of the manufacturing process to be performed by the CIM system. The A-Plan is created when you click MO in a completed Manufacturing Order form. The A-Plan is a table of sequential instructions which the CIM Manager executes in order to produce the products being ordered.

This report is intended for the more experienced user. See Experimenting with Production Strategies Using the A-Plan in Chapter 11, OpenCIM Programming, for detailed information on how the part will be produced.

The following is an example of an A-Plan Report.

Purchase Order Report

As explained earlier in this chapter, clicking on the **PO** icon in the **MRP** program activates the Report Generator, which will display or print the Purchase Order.

Figure 7-38: Purchase Order Report

User-Defined Report

The User-Defined Report allows you to design and customize a report to the exact specifications of your CIM system. You can create an unlimited number of User-Defined Reports.

The Report Generator program employs Seagate (Microsoft) Crystal Report (also available in Visual Basic) to generate the OpenCIM reports. The Crystal Report program necessary to create your own user-defined report is not included with the OpenCIM software and must be purchased separately. Refer to the documentation provided by the Microsoft Visual Basic Crystal Report for instructions on how to create a report.

8

Virtual CIM Setup

The Virtual CIM Setup is an interactive graphic module that allows you to create a simulated CIM cell. The CIM cell may contain the actual elements and connections of a real CIM installation, or it may define a theoretical (meaning, virtual) CIM cell. This chapter includes the following sections:

- **CIM Setup Overview**, introduces the OpenCIM Setup application.
- **Accessing CIM Setup**, describes how to access the virtual CIM Setup application.
- **CIM Setup Main Window**, describes the components of the CIM Setup interface.
- **Working with the CIM Setup**, provides guidelines for setting up and working in your CIM cell.
- **Tutorial**, describes step by step instructions for designing and operating a CIM cell.

CIM Setup Overview

The first part of this chapter presents the menus and screen elements of the Virtual CIM Setup module. You are then encouraged to perform the Tutorial, which will enable you to practice using this module and create a Virtual CIM.

Click the Virtual CIM Setup icon to open the Setup screen. A number of menus are available, but many menu items will not be available until a setup has been created or loaded.

Virtual CIM Setup Limitations

The Virtual CIM Setup allows you to construct all sorts of CIM cells, although some configurations would be difficult or even impossible, to actually operate. To ensure your success in operating a Virtual CIM cell, create your cell in accordance with the following guidelines:

- Use only one conveyor in the cell.
- Use no more than eight stations around the conveyor.

Accessing CIM Setup

The CIM Setup application is accessed from the Project Manager main window, enabling the user to create and modify the Virtual CIM setup of the selected CIM Cell.

To access the CIM Setup application:

From the Project Manager Main window, shown in Chapter 5, Project Manager, click CIM Setup on the toolbar. The Virtual CIM Setup Main window is displayed, as shown in Figure 8-1.

CIM Setup Main Window

The Virtual CIM Setup Main Window appears, as follows:
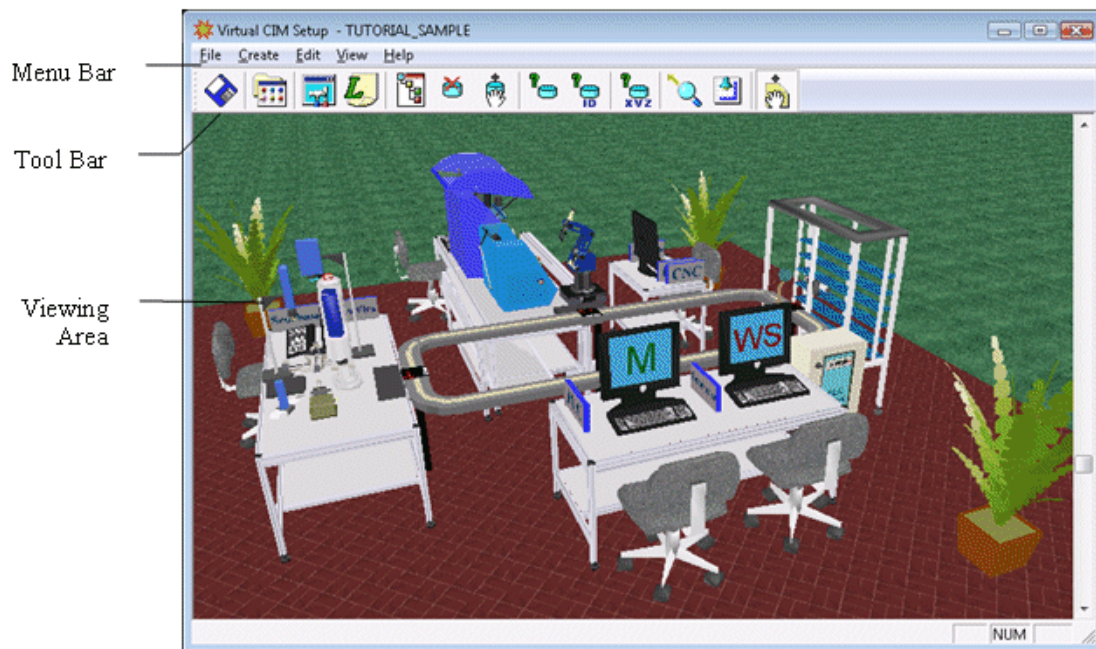
Figure 8-1: Virtual CIM Setup Main Window

The Virtual CIM Setup window, shown above, contains the following elements, each of which is described in the sections that follow:

- CIM Setup Menu Bar
- CIM Setup Toolbar

Virtual CIM Setup Menu Bar

The Virtual CIM Setup menu bar contains five menus, each of which is described in detail in the sections that follow.

File Menu

The following table contains a brief description of each option in the File menu:

Option	Description
Save	Saves the current placement of all objects in the CIM cell.
Exit	Exits the Virtual CIM Setup module. You will be prompted to save the current placement of objects.

Create Menu

The following table contains a brief description of each option in the Create menu:

Option	Description
Group	The system automatically creates shortcuts to loaders for drivers of all the workstations in the selected project. This group contains shortcuts to loaders that are required in order to operate the CIM cell defined by the Virtual CIM Setup.
	The group can be accessed from the following directory path:
	Windows XP: [default system drive (for example C:)]\\Documents and Settings\All Users\ Documents\Intelitek\OpenCIM\Projects\<name of project>\OpenCIM<name of project>
	Windows Vista/7: [default system drive (for example C:)]\Users\Public\Documents\Intelitek\OpenCIM\Projects\<name of project>\OpenCIM<name of project>
Setup File	Prompts you to confirm the overwrite of the SETUP.CIM file. For full details, refer to *Chapter 12, Inside OpenCIM*
Loader Section	Creates a file WS*n*.INI for each station (e.g. WS2.INI).
ACL DMC File	Prompts you to confirm the overwrite of the DEVICE.DMC file. For full details, refer to *Chapter 9, OpenCIM Device Drivers*, and *Chapter 12, Inside OpenCIM*.

Edit Menu

The following table contains a brief description of each option in the Edit menu:

Option	Description
New Object	Displays the New Objects dialog box, enabling you to add new elements to your new CIM cell, as described, for example in *Adding a Conveyor*. The New Objects dialog box contains all the elements which can be included in the Virtual CIM Setup, as follows: • Double click on the [+] button of a category in order to expand its list of elements and select an object. • Move the cursor into the graphic scene. The cursor changes to a cube. Point and click on the location where you want to place the object. You may need to wait a moment for it to appear; *do not double-click*.
Delete Object	Activates the delete mode in order to delete an object from the cell. Using the cursor, point and click on the object you want to delete. Click **Yes** to confirm the deletion. If you click **No**, the delete mode remains in effect, enabling you to select and delete another object.
Drag Object	Enables you to pick an object and drag it through the cell
Setup IDs	Displays the Edit IDs dialog box, containing a list of all the objects and ID numbers in the cell, enabling you to change the ID number of a device. Double-click on the highlighted line, and enter a new number at the prompt.
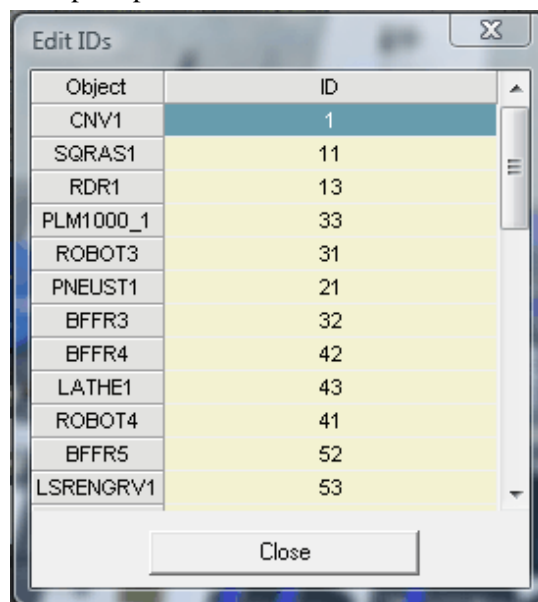
Figure 8-2: Edit Setup IDs Menu

Once you have changed an ID, be sure to select **Create | Setup File** and **Create | ACL DMC File** in order for the change to take effect.

Option	Description
Floor Size	Opens the Set Floor Size dialog box that enables you to set the X and Y dimensionsof the cell. Click **OK** to confirm the new size or **Cancel** to close the dialog box.
Set Textures	Opens a sub-menu listing cell textures:floor, land and background enabling you to select the required *.bmp file for the specified texture.
Building	Displays the walls of the building in the CIM Cell.

View Menu

The following table contains a brief description of each option in the View menu:

Option	Description
Show Names	Displays the names of all devices in the scene.
Show IDs	Displays the device ID numbers.
Show Positions	Displays the position coordinates of all devices.
Redirect Camera	Allows you to select a different focal point in the scene.
Drag 3D Frame	Allows you to drag the cell.
Camera Top	Displays the overhead view of the scene.
Scene Origin	Displays the scene origin, e.g. coordinate 0,0 of the room.When selected, a red cross is displayed in the center of the shop floor.
Toolbar	Displays or hides the toolbar.
Status Bar	Displays or hides the status bar.

Note

The options in this menu are similar to those in the View menu of the Graphic Display and Tracking module. For full details, refer to Chapter 6, Operating CIM Manager

Virtual CIM Setup Toolbar

The toolbar icons provide quick access to the most commonly used functions in the Virtual CIM Setup window and appear as follows:

The following table contains a brief description of each option on the CIM Manager toolbar:

Option	Description
	Save: Saves the current placement of all objects in the CIM cell.
	Groups: The system automatically creates a group of icons in your windows **Start** menu of the selected project.
	Setup: Prompts you to confirm the overwrite of the SETUP.CIM file. For full details, refer to *Chapter 12, Inside OpenCIM* .
	Loader: Creates a file WS*n*.INI for each station (e.g. WS2.INI).
	New Object: Displays the New Objects dialog box, enabling you to add new elements to your new CIM cell as descibed, for example, in *Adding a Conveyor*.
	Delete Object: Deletes currently selected object.
	Drag Object: Allows dragging of object from one location to another.
	Show Name: Shows name of the object.
	Show ID: Shows ID of the object.
	Show Position: Shows coordinates (from bottom left corner).
	Redirect: Defines the position that will be in the center of the image.
	Camera Top: Places the camera on top of the cell at the center of the image,
	Drag 3D Frame: Enables you to pan the CIM cell left, right, up and down.

Configuration Parameters Popup Menus

The Virtual CIM cell contains configuration parameters popup menus that are displayed when you double-click on an object in the Virtual CIM. The following figure shows some of the variations of the Configuration Parameters menu.
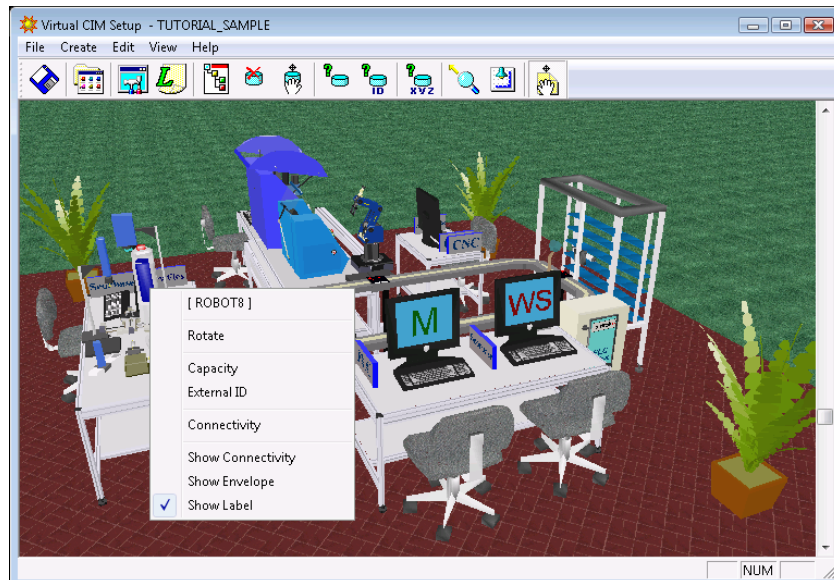
Figure 8-3: Parameter Configuration Popup Menus

Once defined and saved, these parameters are written to OpenCIM INI files, as described in Chapter 12, Inside OpenCIM.

The following table contains a list as well as the description of all parameters that can be defined. The parameters for many objects do not normally require manipulation. The Tutorial at the end of this chapter demonstrates the recommended sequence and actions for setting object parameters.

Option	Description
Capacity	Defines the number of items that the device can hold.
	You must define a value for feeders, racks and trash bin. For most other objects you can accept the system-defined value.
	When selecting new objects, select **Buffer 2** for a buffer whose capacity is two items, and select **Buffer 1** for a buffer that can contain only one item. The capacity parameter is thus already defined as 2 and 1, respectively.
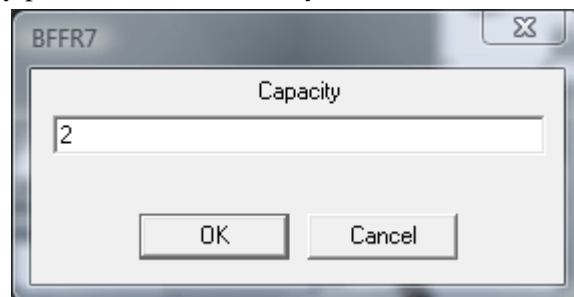
Figure 8-4: Defining Capacity

Option	Description
CommPort	For ACL device drivers, the settings must be as follows:

```
BaudRate = 9600
Parity   = None
DataBits = 8
StopBits = 1
XonXoff  = No
```

These are the standard RS232 settings for communicating with ACL controllers (both Controller-A and Controller-B). These settings should not be changed since they match the fixed settings in the controller.

For the PLC device driver, the settings will be one of the following:

OMROM PLC	Allen-Bradley PLC	LG PLC
BaudRate = 9600	BaudRate = 9600	BaudRate = 9600
Parity = Even	Parity = None	Parity = None
DataBits = 7	DataBits = 8	DataBits = 8
StopBits = 2	StopBits = 1	StopBits = 1
XonXoff = No	XonXoff = No	XonXoff = No

These are the RS232 parameters used by the PLC device driver when it communicates with the PLC. You must set these parameters to match the RS232 settings on the PLC.

Connectivity — Connects all objects and their associated device drivers.

Also connects robots to all objects which are physically within their reach and to all objects which can be connected by means of an RS232 cable.

When you click on Connectivity, the Connections menu opens.
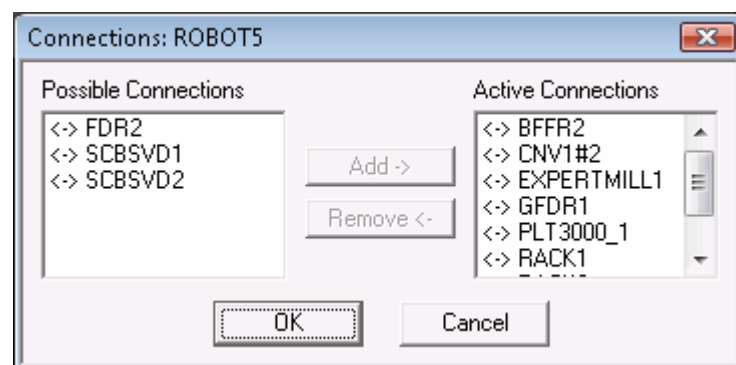
Figure 8-5: Defining Connections

Option	Description
	One or more connections can be added or deleted at a time.

- To make a connection, highlight the device driver name(s) for the selected object, click **Add**, and then **OK**.
- To remove a connection, highlight the device driver name(s) for the selected object, click **Remove**, and then **OK**.

Active Connections: A list of all connections that have already been established between this and other objects.

Possible Connections: A list of all objects to which this object can be connected. For robots, this list will also display all devices that are in physical reach of the robot (i.e., that are in the Work Volume of the robot), and all objects that can be connected by means of an RS232 cable.

Possible Connections for Robots will also display all the device drivers for this robot within the CIM cell, since robot controllers can communicate with any PC in the CIM cell. Normally, you should select only one robot device driver per robot. Do not select device drivers that are not intended for this robot.

If you want to connect an object (not a device driver) to the robot, and the object is not within reach of a robot, reposition it on the screen until you see the object's name appear in the list of possible connections.

Conversely, if you move an object too far away from a robot, the connection will be severed. A warning will prompt you to reposition the objects and reestablish the connections, if desired.
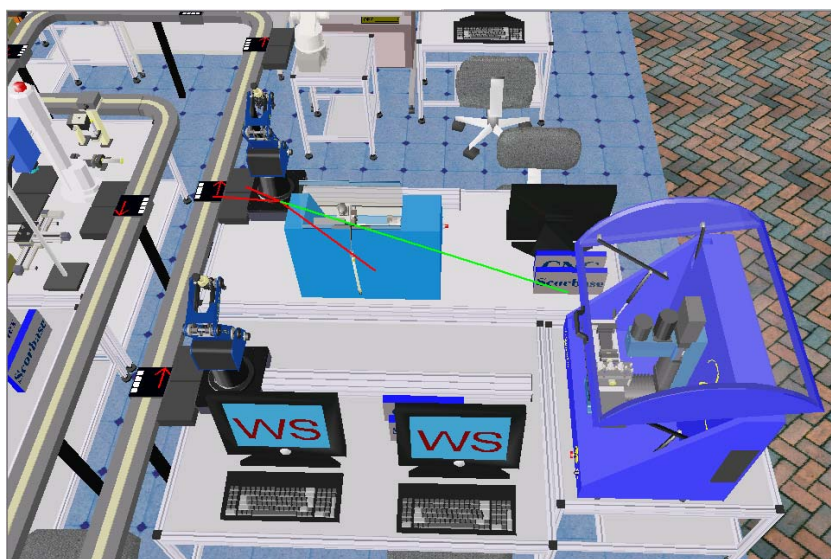
Figure 8-6: Robot Connections

✍Tip*: To see which device driver belongs to an object, click on the object. From the object's Parameter Configuration menu, select Connectivity/OK. A line connecting the object and its associated device driver appears on the screen . (Alternatively, click on a device driver to find its associated device.)*

Green lines show the connections between device drivers and their associated objects. **Red** lines show the physical connections between robots and other objects; i.e., devices that are within the robot's reach.

Option	Description
	Green lines indicate which software is running on a PC. Click on a PC. To see the green lines, select Show Connection from the Parameter Configuration menu.

Figure 8-7: Workstation Connections

External ID	Unique numerical Device ID that identifies each device in the system. The software defines IDs automatically in sequence as the devices are created in the Virtual CIM cell. You can, however, modify this ID in order to structure the numbering of the devices, for example, in relation to the station they are located in (see example in the tutorial).
Location	Identifies the location of a feeder when it is placed and used within an ASRS unit. (In an actual CIM cell, a feeder can only be placed on an ASRS carousel, not in the $ASRS^2$, ASRS-36u, or ASRS-36x2.)
Properties	Defines the Workstation (WS*n*) at which the object's device driver is connected and running.
	Also defines the INI file associated with the object's device driver. This INI file contains the definitions and parameters for the object, and is invoked by the device driver's Loader command line.
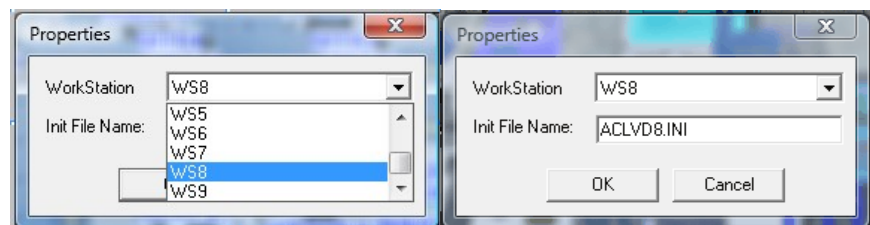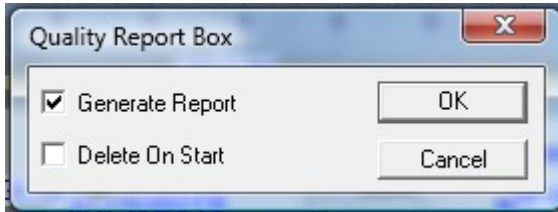
Figure 8-8: Defining Workstation for Device Driver

Once Properties have been defined for a device driver, other options in the Parameter Configuration menu become available.

Option	Description
QC Report	The Quality Control device driver allows you to write the results of a QC test to an ASCII text file that can be input to a spreadsheet program.

- Delete on Start. This switch controls whether a new quality control report file will be created each time this device driver is activated. If checked (Yes), the previous file is deleted; if not checked (No), results are appended to the existing report file.

Figure 8-9: Defining Quality Control Report

Option	Description
Rotate	Rotates a displayed object to any degree. Useful, for example, to properly attach buffers to the conveyor, or to logically orient a PC in the Virtual CIM display.
Sub Type	System-defined parameter. Normally, you do not need to manipulate this parameter.
Variables	The default values for the variables used by the CNC script.
Write Load	Creates the WS*n*.INI file for the particular workstation.

Working with the CIM Setup

This section describes how to set up your CIM Cell and provides guidelines for creating conveyors (user defined, rectangular and L-shaped), as well as adding stations, tables and robots and so on. It includes the following:

- Creating a User Defined Conveyor
- Creating a Rectangular
- Creating an L-Shaped Conveyor
- Adding Tables
- Adding Robots
- Manipulating the Graphic Display
- Changing the Focus of the Graphic Display

Creating a User Defined Conveyor

Since the Virtual CIM conveyor is the most difficult of the elements to place in the scene, this section contains detailed, procedural instructions for creating a conveyor.

If you are replicating an actual CIM cell, you will need to know the exact dimensions of the conveyor.

To add a user defined conveyor:

2. Select **Edit | New Object**. *The* **New Objects** dialog box is displayed.
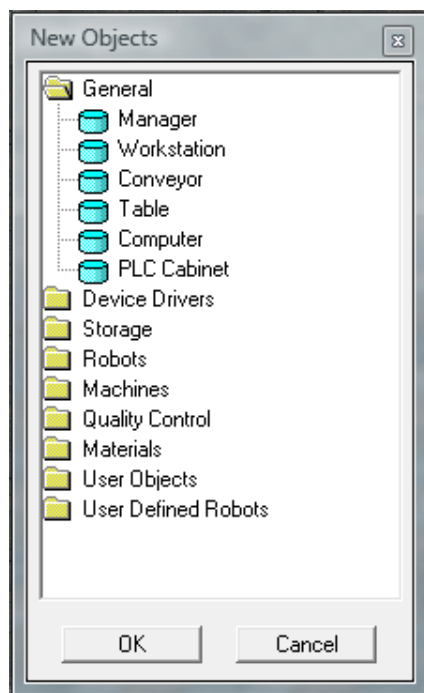
Figure 8-10: New Objects Dialog Box

3. From the **General** tree select **Conveyor**. The Conveyor dialog box is displayed.
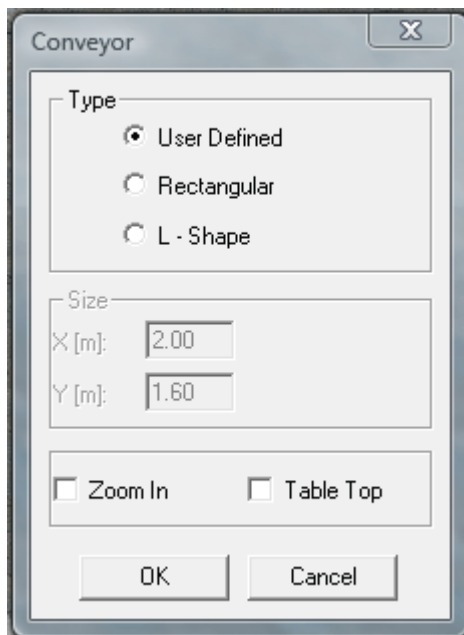
Figure 8-11: Conveyor Dialog Box

4. **From** the Type area, select **User Defined**.
5. Select the required option, as follows:
6. Select Zoom In to place a typical 3m X 4m OpenCIM conveyor (with straight segments of 1.40m). Each grid represents about20 cm. When Zoom In is not selected, the grid resolution is increased, which enables you to place a larger conveyor (8 m x 10 m) into the CIM cell
7. Select Table Top to place the conveyor on a table.
8. Click **OK**. A grid will appear on the screen. The set of numbers displayed on the left are the screen pixel coordinates of the cursor, while the numbers on the right are metric coordinates (in meters).

Note

When creating a conveyor, use the mouse buttons as follows:

*   Click with the right mouse button and drag to select menu options
*   Point and left-click to place objects on the grid.

To select the starting point of the conveyor, do the following:

1. Place the cursor on the grid and right-click to open this icon menu:
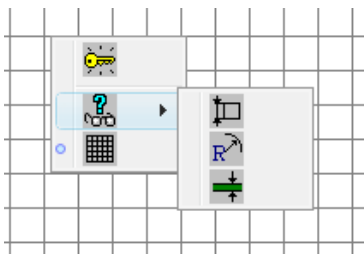
The **key** icon is used to mark the starting point of the conveyor.

The **spectacles** icon opens another icon menu. *Do not attempt to manipulate the displayed values. These settings are for technical support personnel only.*

The **grid** icon toggles the grid display on and off.

2. Drag and select the key icon. A wand-pointer appears on the screen. Use this cursor to click on a starting point for the conveyor. *It should be placed on the right side of the grid*, as shown in Figure 8-12: Adding Conveyor Segments. A red dot appears on the grid.

Since the conveyor movement is normally counterclockwise, the conveyor segments are added in the counterclockwise direction.

To create the conveyor, do the following:

1. Again click on the right mouse button. An icon menu will open.
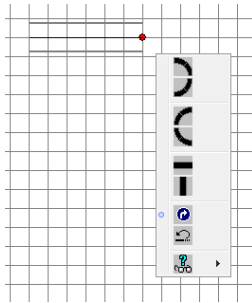
Segment icons: The first six icons represent segments of the conveyor. Segments are added and connected in consecutive order.

To insert a straight segment, select either the horizontal or vertical bar. Using the left mouse button, place the pointer on the square whose upper left hand corner marks the end of the segment you want to add. Carefully position the cursor in order to align the segments of the conveyor

Reverse: The white curved arrow in the blue circle is used to reverse the direction of conveyor movement. When this icon is selected, conveyor movement is clockwise, and the segments will be added accordingly.

Undo. The undo icon cancels the last segment(s) added to the conveyor. Undo is available after the first segment has been placed on the grid.

OK. This button will appear when you have completely connected all segments of the conveyor.

Spectacles: The spectacles icon opens another popup menu. *Do not attempt to manipulate the displayed values. These settings are for technical support personnel only.*

2. Using the icon menu, select and connect each segment of the conveyor.
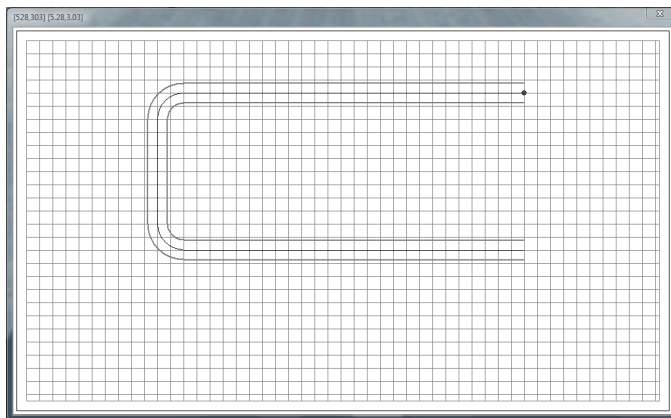
Figure 8-12: Adding Conveyor Segments

Tips

Avoid using too many UNDOs while drawing the conveyor; a faulty display may result.

When drawing the conveyor, do not extend the straight segments too close to the edge. Allow enough room for the corner segments.

3. When you have completed drawing the conveyor, click on the 🔲 button. A chain of dots will appear in the conveyor.

Creating a Rectangular Conveyor

As an alternative to creating your own user defined conveyor (described in Creating a User Defined Conveyor), the Virtual CIM Setup also enables you to select a predefined rectangular conveyor. This includes, defining the conveyor dimensions and zoom options and so on.

To add a rectangular conveyor:

1. Select Edit | New Object. The **New Objects** dialog box is displayed as shown in Figure 8-10: New Objects Dialog Box.

2. From the General tree, select Conveyor and then click OK. The Conveyor dialog box is displayed, as shown in Figure 8-11: Conveyor Dialog Box.

3. Select Rectangular and define the conveyor dimensions in the Size area.

4. Select the required option, as follows:

   1. Select **Zoom In** to place a typical 3m X 4m OpenCIM conveyor (with straight segments of 1.40m). Each grid represents about 20 cm. When **Zoom In** is not selected, the grid resolution is increased, which enables you to place a larger conveyor (8 m x 10 m) into the CIM cell

   2. Select **Table Top** to place the conveyor on a table.

5. Click **OK**. A Rectangular conveyor is displayed, as shown in the following example:
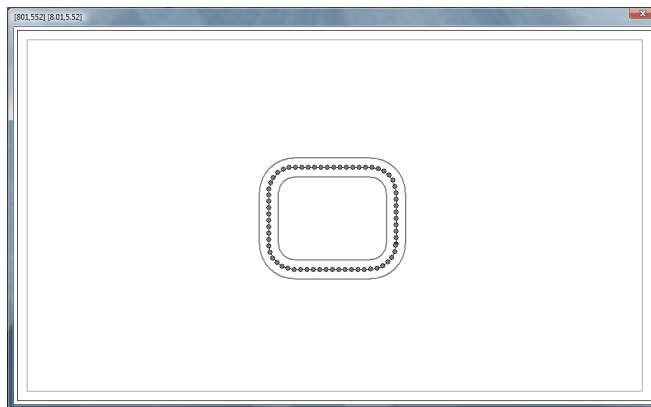
Figure 8-13: Rectangular Conveyor

Creating an L-Shaped Conveyor

As an alternative to creating your own user defined conveyor,(described in Creating a User Defined Conveyor), the Virtual CIM Setup also enables you to select a predefined L-Shaped conveyor. This includes, defining the conveyor dimensions and zoom options and so on.

To add an L-Shaped conveyor:

1. Select **Edit | New Object**. The **New Objects** dialog box is displayed as shown in Figure 8-10: New Objects Dialog Box

2. From the **General** tree, select **Conveyor** and then click **OK**. The Conveyor dialog box is displayed, as shown in Figure 8-11: Conveyor Dialog Box.

3. Select **L-Shaped** and define the conveyor dimensions in the Size area.

4. Select the required option, as follows:

    1. Select **Zoom In** to place a typical 3m X 4m OpenCIM conveyor (with straight segments of 1.40m). Each grid represents about20 cm. When **Zoom In** is not selected, the grid resolution is increased, which enables you to place a larger conveyor (8 m x 10 m) into the CIM cell

    2. Select **Table Top** to place the conveyor on a table.

5. Click **OK**. An L-Shaped conveyor is displayed, as shown in the following example:
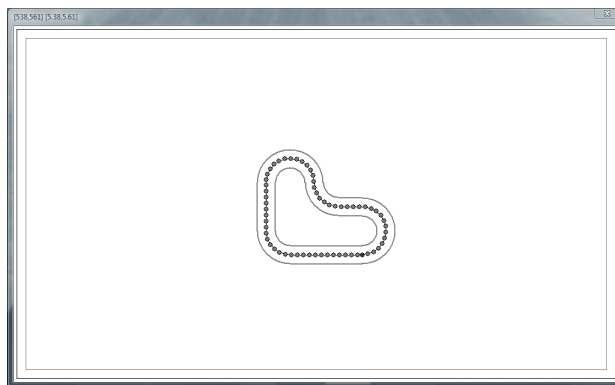
Figure 8-14: L-Shaped Conveyor Segments

Adding Stations

After defining the conveyor you must place the required stations around the conveyor. This includes defining the station location and number and so on.

To add stations around the conveyor:

1. Click on the right mouse button. A popup menu is displayed.

2. Select the **Station** ⬚ icon. Then point and click on one of the dots in the conveyor. The Conveyor Stations dialog box is displayed.
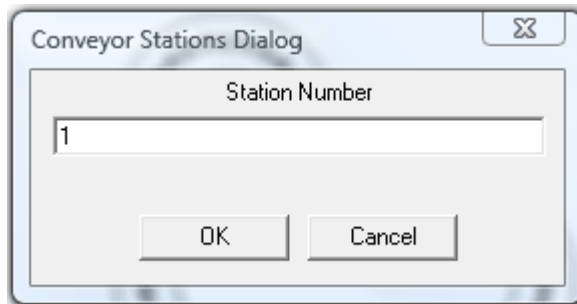
Figure 8-15: Conveyor Stations Dialog Box

3. Enter the station number in the field provided and click OK.

☞**Tip** *The first station should immediately follow the starting point of the conveyor, in accordance with the counterclockwise or clockwise movement of the conveyor. Additional stations should be placed consecutively in the same direction until you reach the starting point again. It is recommended that you plan your conveyor in advance to determine the correct location for each station.*
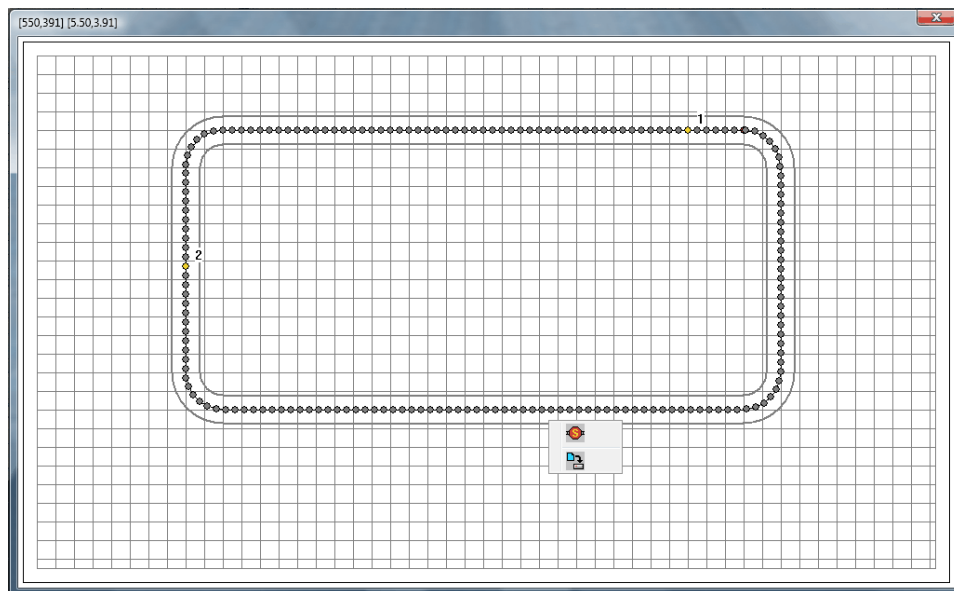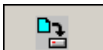
Figure 8-16: Adding Conveyor Stations

☞
Tip *Place stations on straight segments of the conveyor. They cannot be placed on the curves.*

4. Repeat steps 2 through 3 for each conveyor station.

5. Select the **File** ⬚ icon to save the conveyor definitions.

Once you have saved the conveyor setup, the conveyor will be drawn on the CIM scene window.

Do not save the conveyor until you have placed ALL stations around the conveyor. Once a conveyor has been saved, it cannot be altered. To change a conveyor configuration, you will need to repeat the entire procedure.

Note

Adding Tables

Tables should be placed at every station around the conveyor, either after the conveyor is created, or after all elements have been placed in the scene.

When you place objects in the Virtual CIM, you do not need to define a height coordinate, since the system assumes all objects are at their proper heights. All objects will be displayed at the correct height, even if they are not sitting on tables.

You can select size and color of any table you want to create, but only before you create it. Once the table is made it cannot be altered. To make any changes in tables properties you will need to repeat the entire procedure.

It is recommended that you place tables in the Virtual CIM so that objects at the stations will not appear to float in space.

Notes

To place a table on the Virtual CIM, do the following:

1. Select **Edit | New Object** The New Objects dialog box is displayed, as shown in Figure 8-10: New Objects Dialog Box.

2. From the **General** tree select **Table** and then click **OK**. The Table dialog box is displayed.
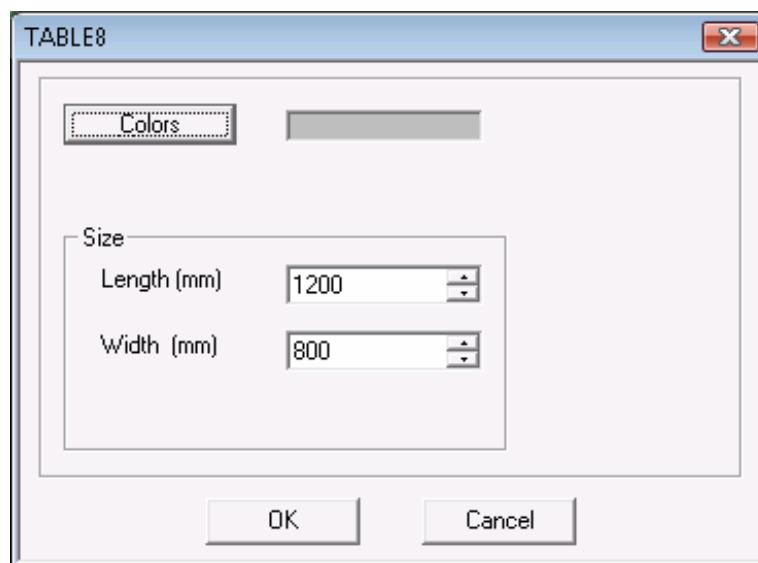
Figure 8-17: Table Dialog Box

3. Click the Colors button to change the table color.
4. In the Size area, define the dimensions of the table and click OK.
5. Point and click on a spot near the first conveyor station to place the first table.
6. Repeat steps **2** through **5** to place tables at all the stations around the conveyor.

7. Click and drag on the cursor to adjust the location of each table in the scene. The coordinates that appear on the table (and any other object that you manipulate) mark the center point of the table or object relative to the cross at the center of the shop floor, which is displayed by selecting Scene Origin from the view menu.
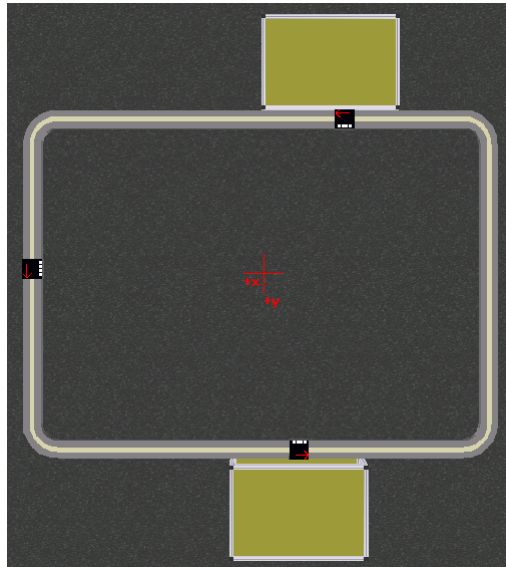
Figure 8-18: Adding Tables

Adding Robots

As indicated at the beginning of this chapter, after the conveyor is created, you should add the ASRS, robots, machines and other devices.

By default the system automatically assigns numbers to the robots (and all other objects) in the order of creation. You should therefore place your first robot (Storage robot) at Station 1, the second robot at Station 2, and so on.

To add robots:

1. Select **Edit | New Object** The New Objects dialog box is displayed.

2. From the **Robots** tree click the Robot to add (**ER 4u**, for example) and then click **OK**. The Optional LSB box is displayed. .
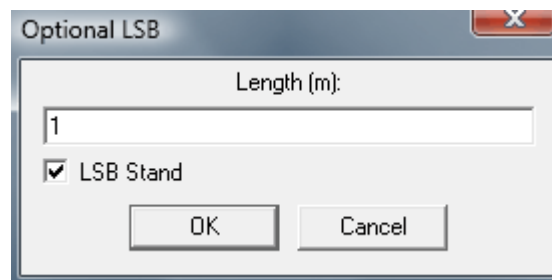
Figure 8-19: Optional LSB Dialog Box

3. If the robot is to be placed on a slidebase, enter the slidebase size in meters and click **OK**. Otherwise, click **Cancel**.

If the slidebase requires a stand, check LSB Stand in the dialog box and click OK.

4. Point and click on a spot at the first station to place the first robot. Repeat this step for each station around the conveyor.

5. Double-click on a robot to open its Configuration Parameters popup menu.

The Robot Configuration Parameters menu differs from the Table Configuration Parameters menu. Each object in the Virtual CIM has a particular Configuration Parameters menu, as described in the *Configuration Parameters Popup Menus* section

Manipulating the Graphic Display

The Virtual CIM Setup module uses the same method as the Graphic Display module for manipulating the view of the CIM cell.

❶
❷
❸

Procedure

Manipulating the Graphic Display

1. To change the angle of the overhead scene, place the cursor on the vertical scroll bar and drag it up and down.

2. To rotate the scene, place the cursor anywhere on the screen and:

   - Click the right mouse button and drag to the right to rotate the display counterclockwise.
   - Click the right mouse button and drag to the left to rotate the display clockwise.

3. To zoom the scene, place the cursor anywhere on the screen and:

   - Click the right mouse button and drag up to zoom in.
   - Click the right mouse button and drag down to zoom out.

Changing the Focus of the Graphic Display

❶
❷
❸

Procedure

Changing the Focus of the Graphic Display

1. Click on **View|Redirect Camera**.

2. Click any object in the scene. It now becomes the center point for the display manipulation. The view changes to an overhead scene (if it is not already), which you can now manipulate, as described in Manipulating the Graphic Display.

Tutorial

In this tutorial you will learn to create and run an OpenCIM cell which contains three stations. It includes the following:

- **Stage 1: Designing the CIM Cell**: Describes how to create a graphic CIM cell using the Virtual CIM module.
- **Stage 2: Operating the CIM Cell:** Describes how to define and operate the CIM cell using the CIM Definition and CIM Operation modules. You will also use the Graphic Display module to view your CIM cell in operation.

Stage 1: Designing the CIM Cell

The following procedure is the recommended sequence for setting up the Virtual CIM Cell.

❶
❷
❸
Procedure

Setting up the Virtual CIM Cell

1. Plan and document your CIM system before starting.
2. Define the conveyor and the location of the workstations.
3. Place tables at the stations.
4. Place an ASRS device at a workstation (station 1 is recommended).
5. Place robots at the stations.
6. Place CNC and any other machines at the stations.
7. Place buffers at the workstations.
8. Place PCs at the workstations.
9. Define the device drivers for the workstations.
10. Define all connections and properties for each device driver.
11. Create the Setup, Map and Icon Group for this CIM cell.

This section will guide you through a complete procedure for creating an OpenCIM cell using the Virtual CIM Setup module.

To design the CIM Cell:

1. From the Project Manager main window, click **New** on the toolbar. The New Project dialog box is displayed.
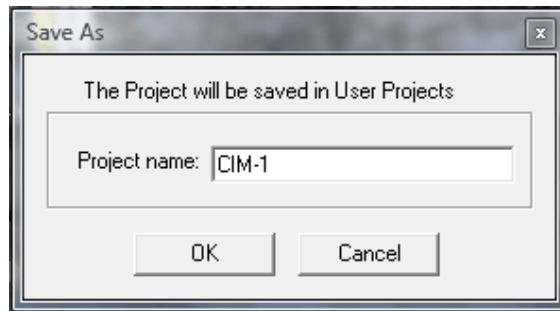
Figure 8-20: New Project Dialog Box

2. In the Project name field, enter **CIM-1** and click **OK**. The new project is displayed in the User Project tab in the Project Manager main window (for further details refer to *Chapter 5, Project Manager*).

3. Select the **CIM-1** project and click the **Virtual CIM Setup** icon on the toolbar. The Virtual CIM Setup main window is displayed, as shown in *Chapter 8, OpenCIM Setup*.

4. Select **Edit | New Object**. In the New Objects window, double-click **General** to expand the tree, and then select **Conveyor**. The Conveyor dialog box is displayed.
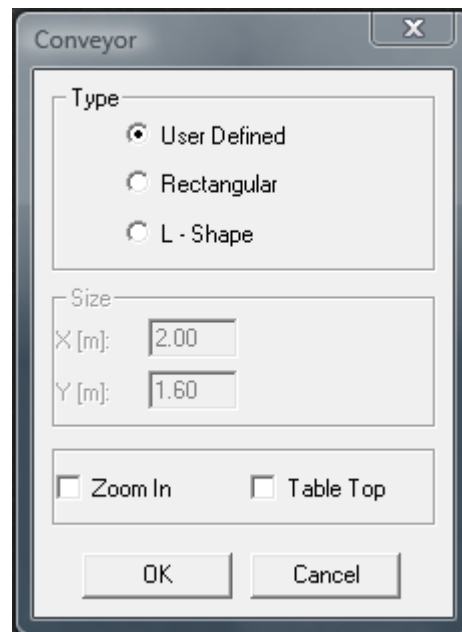
Figure 8-21: Conveyor Dialog Box

5. From the Type area, select User Defined and click OK. The Conveyor grid appears.

    Click the right mouse button. Drag the mouse and then select the key icon .

6. A wand appears. Bring the wand into the lower right side of the grid and click the left button. A red dot appears; this is the conveyor starting point.

7. Click the right mouse button. Drag and select the vertical segment. Point and click the cursor on a spot above the starting point.

8. Select an arc-up-and-left segment. (The conveyor is created in the direction of movement; normally counter-clockwise.)

9. Continue selecting and adding segments to the conveyor until it is complete. Use the ⟲ curved arrow to undo any mistakes. (Avoid using too many undos.)

10. When the conveyor is complete, click the right mouse button and then click the 𝗢𝗞 icon.

11. Click the right mouse button and select the 🅢 **Station** icon from the popup menu, to create a station. Point to a location just above the starting point of the conveyor (not on the curve). Click the left mouse button. Accept the prompt for Station number **1**.

12. Add the stations in accordance with the counterclockwise or clockwise movement of the conveyor.

13. Repeat the previous step for two more stations. Add the stations in consecutive order around the conveyor.

14. When all three stations are marked around the conveyor, click the right mouse button and select the **File** 🗂 icon from the pop-up menu to save the conveyor and click **OK** to save the conveyor setup.

15. The grid closes, and a cube cursor appears on the screen Bring the cursor to the center of the screen and click. (To display a red cross marking the center of the shop floor, select **Scene Origin** from the view menu). The conveyor appears in the Scene Window. Click on the conveyor and drag with the left mouse button to center the conveyor on the floor.

16. Select **File | Save** to save your work.

17. Select Edit | New Object | Storage | ASRS36.

18. Click the left mouse button near Station 1. The ASRS36 appears on the screen. (*This object is the robot that has a storage rack with 36 shelves.*)

19. Click on the **ASRS36**. The object's parameter configuration submenu opens. Select Rotate. Enter **–90**. The lower small frame should be outside the conveyor. Adjust the orientation and position of the ASRS36.

20. Add Tables for the workstations.

21. Add objects to Stations 2 and 3, as follows:

    - At Station 2, select Robots | ER 9, and enter 1.0 (meters) at the prompt for LSB (robot will be mounted on a linear slidebase).

    - At Station 3, select Robots | ER 14, and then click Cancel at the prompt for LSB.

During the selection and placement of new objects, wait for the cursor to settle in place. Avoid double-clicking.

Your screen should now look like this:
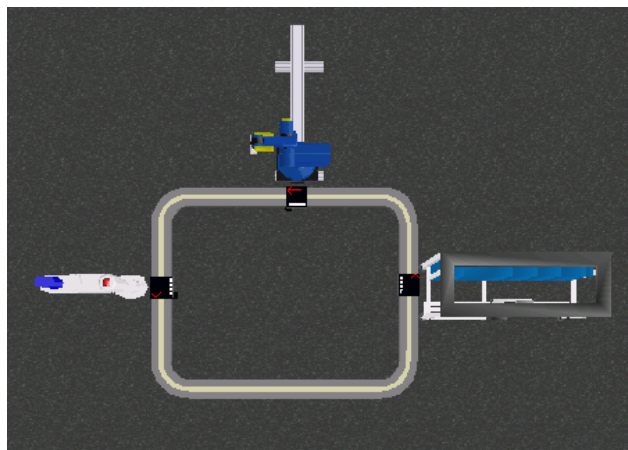
Figure 8-22: Conveyor Stations and Objects

22. Using the same procedure you used for placing robots at stations, add the following objects to the CIM cell:

- At Station 2: EXPERTMILL VMC-600.
- At Station 3: Jig-XY, Screwdriver, and Vision Camera.

As you work, rotate and reposition the objects on the shop floor.

Use Redirect View and Zoom to help you with the placement of objects.

Save your work regularly when creating the CIM cell.

23. Add station buffers (Buffer 2) around the conveyor, in the following order:

- Station 1
- Station 2
- Station 3

24. Select **Edit | Delete Object** and click on the buffer you placed at Station 1. The buffer will be erased. (The ASRS36 does not require a buffer, but a temporary buffer was created at Station 1 causing the buffers at the other stations to be named BFFR2 and BFFR3, respectively.)

25. Rotate the buffer at Station 3 90°. Adjust the location of other buffers, so that they are "attached" to the conveyor and within reach of the robot at the station.

26. At station 3 add a Rack, Feeder and Trash. The following figure will guide you in properly placing all the objects at this station.
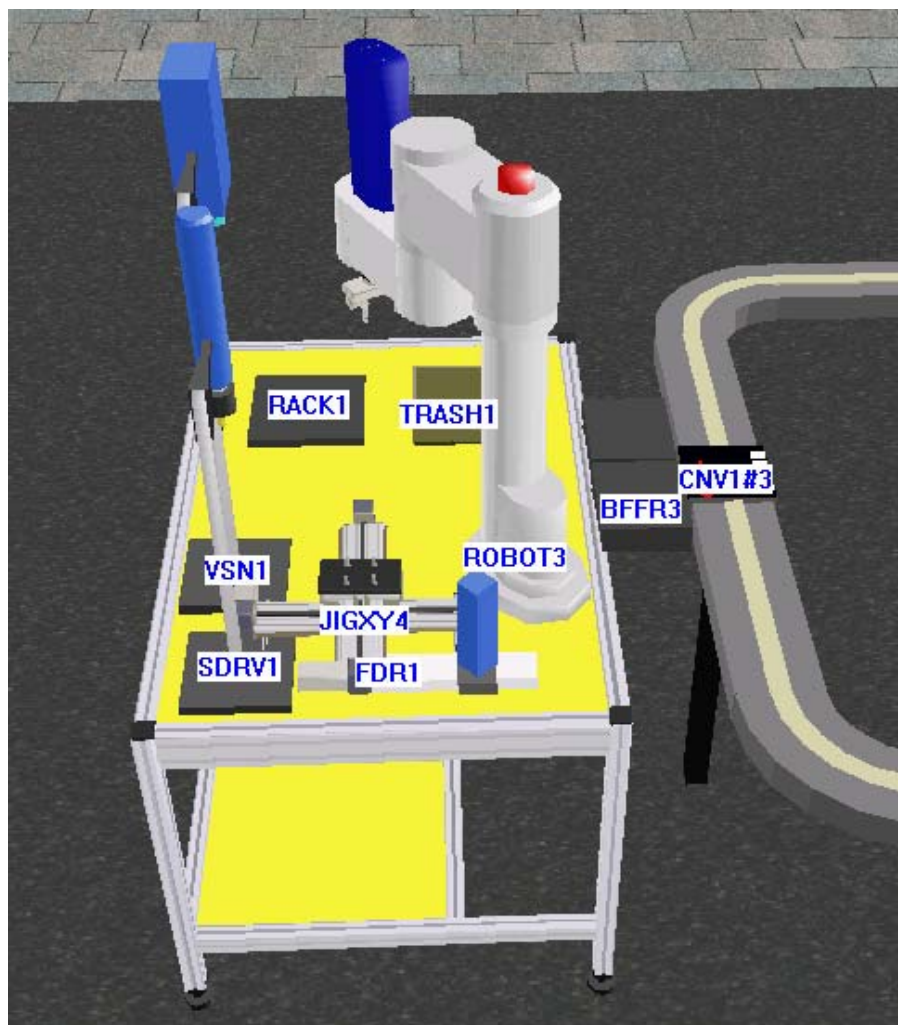
Figure 8-23: Placing Station Objects

27. Save your work at this point.
28. Add workstations (PCs) for each station. Start with the Manager PC and then continue with the Station PC's for Station 1, 2 and 3.
29. Scale Tables for the workstations and arrange them as shown below:
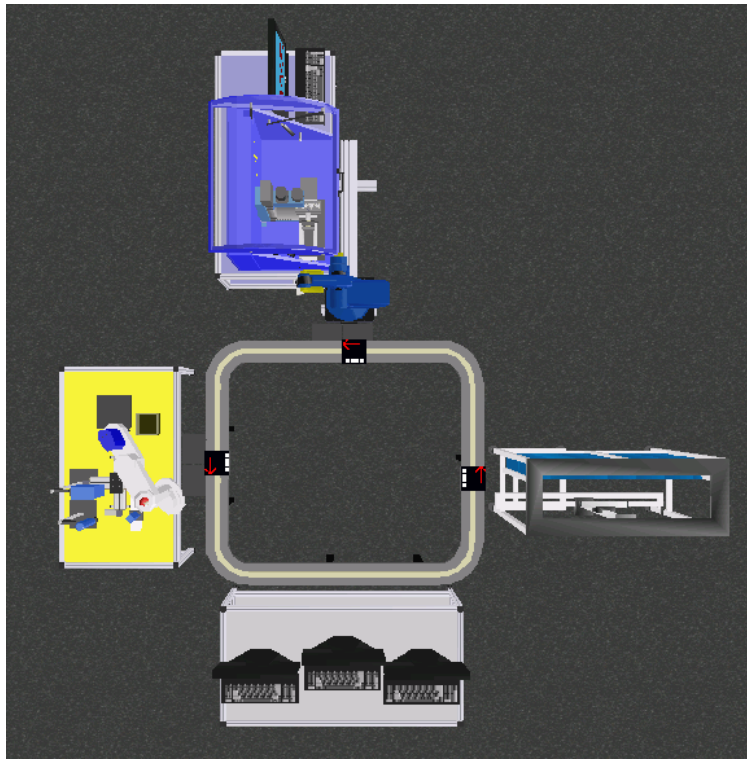
Figure 8-24: Placing Station Tables

30. Add device drivers near the PC at each station:
    - For each robot at the station (including the ASRS36): add Scorbase device driver.
    - For a CNC machine (lathe/mill): add a CNC device driver.
    - For the Vision Camera: add a ViewFlex device driver.
    - For the Conveyor: add a PLC device driver. Place it at Station 1.

    The Jig-XY and Screwdriver are controlled by the Scorbase controller (and its device driver), and do not require device drivers of their own.

31. For each device driver, you will need to set its properties and connections. Click on each object in order to open the corresponding parameter configuration menu, and do the following:
    - Click on Properties. Select the number of the workstation at which the device driver is running (e.g., WS1 for ASRS36 and PLC).
    - Click on the device driver and enter the commport settings
    - Click on Connectivity:
        - Connect the PLC device driver (PLCVD1) to the conveyor (CNV1).
        - Connect each Scorbase device driver (A SCBSVDn) to its corresponding robot (ROBOTn).
        - Connect each CNC device driver (CNCVDn) to its corresponding CNC machine (EXPERTMILLn or PLT3000).

32. For each robot, select Connectivity and make the connections to all objects which are physically within its reach. Connect each robot to the conveyor station (CNV1#*n*), the station buffer (BFFR*n*) and all machines and devices at its station. Make sure the appropriate device driver is connected to the robot. If you are unable to make a connection, move the robot and device closer to each other, and try again.

33. Connect the Screwdriver and the Vision Camera to the Jig-XY only.

34. For the Feeder:

- **SubType** : Set to 101.
- **Capacity:** Set to 10.
- For the Rack:
- **SubType:** Set to 201.
- **Capacity**: Set to 9.
- From the Create menu, perform the following:
- Select Loader Section.
- Select **Setup File** and then click **OK** at the prompt to overwrite the SETUP.CIM file.
- Select **Group**. This will create a directory which contains all the shortcuts needed to prepare and operate the CIM-1 project. You can display this folder by browsing to [projects directory]\CIM1\OPENCIM CIM1 in Windows explorer.
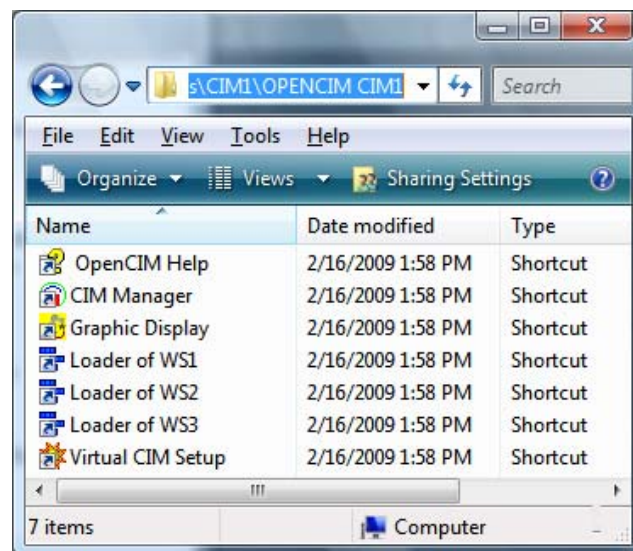
35. Save and exit the Virtual CIM Setup module.

Stage 2: Operating the CIM Cell

This part of the tutorial will give you practical experience in using the CIM operation modules. The following steps all relate to the CIM-1 program group which you have created in the Virtual CIM Setup.

To operate the CIM Cell:

1. From the Project Manager main window, select the User Projects tab and then select the CIM-1 project.

2. From the Project Manager toolbar, select CIM Manager [icon]. The CIM Manager main window is displayed for the CIM-1 project.

3. Select Utility Programs | Machine Definition. The CIM Machine Definition window is displayed.

4. In the CIM Machine Definition window perform the following:

   - From the Machine Name list, select **EXPERTMILL1**, and enter the following information:

     Process Name: PROG_BOX1

     File Name: 1.GC

     Program: leave blank

     Duration: 00:00:25

5. Click **Save**.

6. Close the CIM Machine Definition window.

7. Select **Utility Programs** | **Part Definition**. The CIM Part Definition window is displayed.

8. In the CIM Part Definition window perform the following:

   - Select the Supplied Parts tab.

   - Select File | New Part to define a new part, and enter the following information:

     Part Name: CUBE

     Part ID: 77

     Template Type: 01

9. Click **Save**, and check the Errors box. The message **Save done** indicates there are no errors.

10. Select the **Product Parts** tab.

11. Select **File | New Part** to define a new product part, and enter the following information:

    - Part Name: BOX.

    - Part ID: 75

    - Sub part: CUBE

    - **Process:** PROG_BOX1

    - Template Type: 01

12. Click **Save**, and check the Errors box. The message **Save done** indicates there are no errors.

13. Close the CIM Part Definition window.

14. Select **Utility Programs** | **Storage Manager**. The CIM Storage Manager window is displayed.

---

15. In the CIM Storage Manager window perform the following:
    1. Click the **Edit** button of the **ASRS** storage type. The CIM Storage Definition window is displayed.
    2. Double click any cell in the grid. The Cell Edit dialog box is displayed.
    3. Select **CUBE** from the **Part** drop-down list and click **Save** to close the Cell Edit dialog box.
    4. Repeat these steps three more times for other cells.
    5. Close the Storage Definition window to automatically update the storage database.

16. Click the **Create Default Storage** icon to save this storage definition as the default and close the CIM Storage Manager window to return to the CIM Manager main window.

17. Select **Utility Programs | MRP**. The CIM MRP window is displayed.

18. From the CIM MRP window perform the following:
    1. Select **Customer | New Customer**. The New Customer window is displayed.
    2. In the **Name** field, enter **CUST-A**, and if required enter additional information in the relevant fields.
    3. Create two orders for this customer, each one for two parts, but for different supply dates. (To see a list of parts which can be ordered, open the drop-down list when the cursor is on the Part Name field. Click to select a part.) For example:

```
Part Name:    BOX                          Part Name:    BOX
Required:     2                            Required:     2
Priority:                        1         Priority:                        1
Due Date:     2                            Due Date:     4
```

19. Save the order by clicking the **MRP** icon on the toolbar. This runs the MRP program, which creates a Manufacturing Order.

20. Select the **Manufacturing Order** tab and select a Manufacturing Order (from the list of numbers), and click **MO** to submit the manufacturing order. This creates an A-Plan (production work order) for the order.

21. Close the CIM MRP window.

22. Select **Utility Programs | Report Generator**. The CIM Report Generator window is displayed.

23. From the CIM Report Generator window perform the following:

1. From the reports list select the **Part Definition** report and click the **Print Report** icon to display the report on the screen. You may also select Subparts, Process, Analysis and A-Plan to view other reports.
2. Close the Reports Generator window.

24. From the CIM Manager toolbar click the CIM Modes icon. From the displayed Modes dialog box, perform the following:

   1. In the CIM Mode area, select **Simulation mode** and enter the simulation speed. For example **x5**.
   2. In the Remote Graphic Display area, select **No**. This option refers to the external graphic display only; the internal graphic display of the CIM Manager is always active.
   3. Click **Save** to close the Modes dialog box and return to the CIM Manager main window.
   4. Click the **Start** button to execute (meaning, load the Manufacturing Order).
   5. Click the **Run** button to activate (meaning, run the production cycle.)
   6. Congratulations. Your CIM cell is in operation! Look at the Order View, Device View, Program View and Pallets View and follow the progress of the production.
   7. Acknowledge the messages "Part has been Finished." and "Order Finished."
   8. Close the production by clicking the red **Stop** button.

25. You will now repeat the production cycle, and view it through the Graphic Display module.

26. From the CIM Manager toolbar click the **CIM Modes** icon. The Modes dialog box is displayed.

27. In the Remote Graphic Display area, select **Yes** and then click **Save**. Make sure that you activate the external Graphic Display (see the last step of this procedure) otherwise the CIM will run slowly.

28. Click the **Reset Storage** icon on the CIM Manager toolbar.

29. Click the **Graphic Display** icon from **CIM-1** Group (displayed from your windows **Start** menu). The CIM Simulation screen will appear. You will see the CIM cell you created by means of the Virtual CIM Setup in three different 3D views.

30. Return to the CIM Manager screen and perform the following:

   1. Click the blue **Start** button to execute (meaning, load the Manufacturing Order).
   2. Click the blue **Run** button to activate (meaning, run the production cycle).
   3. You also can see your CIM cell in operation by means of the Graphic Display.

9

OpenCIM Device Drivers

☞ *This chapter is not applicable for OpenCIM Offline.*
Note

This chapter describes the OpenCIM device drivers. Device drivers are interface programs that translate and transmit messages between the CIM Manager and the various machines and controllers at CIM stations. It includes the following sections:

- **Overview of Device Drivers**, provides a brief overview of the OpenCIM device drivers and the devices they control.
- **Device Driver Control Panel**, provides a brief overview of the device driver control panel that is used for sending commands to the device.
- **CNC Device Driver**, describes the CNC device driver that controls various types of CIM machines (lathe, mill and more).
- **Robotic Device Drivers,** describes the robotic device driver that controls the devices attached to robotic controller (such as, robots, barcode scanners and more).
- **Quality Control Device Drivers**, describes the quality control device drivers that control QC devices (such as the laser scan meter)
- **ViewFlex Device Driver**, describes the ViewFlex device driver that controls the Vision Machine System.
- **ULS Device Driver**, describes the ULS device driver that controls all operations of the Laser Engraver.
- **BCR Device Driver**, describes the BCR device driver that controls all operations of the Bar Code Reader.
- **RFID Device Driver**, describes the RFID device driver that controls all operations of the RFID reader.
- **PLC Device Driver**, describes the PLC device driver that controls the operation of the conveyor.
- **Hydraulic Device Driver**, describes the hydraulic device driver that controls the hydraulic station.
- **Pneumatic Device Driver**, describes the pneumatic device driver that controls the pneumatic station.
- **Process Device Driver**, describes the process device driver that controls the process station.

Overview of Device Drivers

Device drivers are the link between the CIM Manager and the devices in the CIM cell. The device drivers are used to perform the following functions:

- Relay command and status messages during production between a device and the OpenCIM network.
- Simulate a device.
- Test a device.

OpenCIM device drivers can run on both Station Manager PCs and the CIM Manager PC, depending on the configuration of the CIM system.

Device Driver	Devices Controlled
CNC	A single CNC machine.
Robotic Device Driver (ACL or Scorbase)	Devices connected to and controlled by a robot controller; e.g., robot, automatic screwdriver, barcode reader.
Quality Control Device Drivers	ROBOTVISIONpro, Laser Scan Meter, ViewFlex, Barcode reader, RFID reader.
ULS	Laser Engraver.
PLC	The CIM conveyor.
Hydraulic	The hydraulic station
Pneumatic	The pneumatic station
Process	The process station

Device Driver Control Panel

Device drivers are loaded automatically by the virtual loader, DDLoader.EXE.

This program loads all device drivers from command lines found in the [Loading] section of the device driver's INI file. Refer to Chapter 12 for more details on the loader program.

All device drivers have the following features:

- A **Control Panel** for manually sending commands to the device and for viewing status information.
- A **Virtual Device Driver Status** window for displaying status information, error messages, and responses from the device when appropriate.

When a device driver is loaded its Virtual Device Driver Status window and its Control Panel appear on the screen. For example:
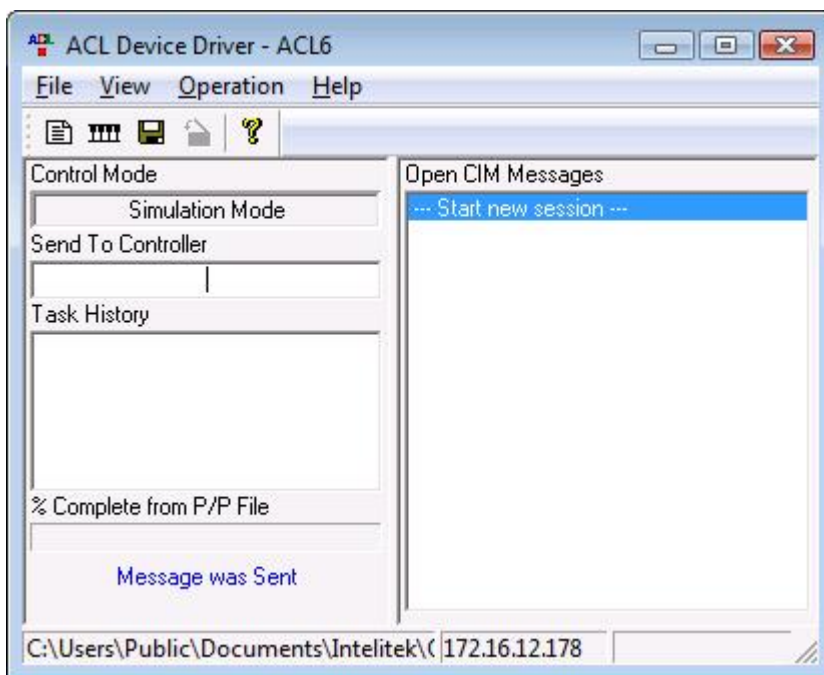
Figure 9-1: Device Driver Control Panel (ACL)

You must close the status window in order to close the device driver.

☞ *The device driver Control Panel is discussed in detail in the section on the ACL device*
Note *driver. The discussion there is applicable to all other device drivers.*

Modes of Operation

Like the CIM Manager, the device drivers can operate in either Simulation Mode or Real Mode. In addition, Manual Mode allows you to interact with the software and hardware.

In order to operate a device driver (DD) in Real Mode, check the Load column next to the desired driver(s). Similarly, to operate a device driver in simulation mode, check the Simulation column next to the desired driver(s).
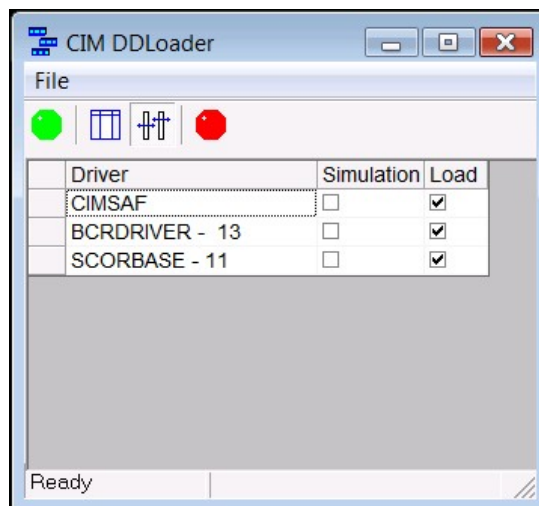
Figure 9-2: DD Loader

The following table describes device driver modes:

Option	Description
Real Mode	Normal operating mode. The device driver is ready to communicate with both the CIM Manager and the physical device (or its controller).
	The message **Connected OK** is displayed after the device driver successfully receives the first message from the device (either on a serial port or a PC I/O card).
	In Real Mode, all communications between the device and other CIM entities occur automatically. However, it is also possible to manually send commands to the device using the Control Panel.
Simulation Mode	In Simulation mode, the device driver receives commands as usual from the CIM Manager and emulates a device by automatically responding with the appropriate status messages. The device driver does not actually communicate with the physical device.
	The quality control device drivers randomly return either a successful or unsuccessful status message based on the parameter FailPercent. All other device drivers always return a successful status message in Simulation mode.
Manual Mode	In Manual mode, the device driver receives commands as usual from the CIM Manager while you interactively emulate the device using the device driver's Control Panel. The device driver does not generate status messages automatically; they are generated only when you manually make selections from the Control Panel.
	In Manual mode the device driver does not actually communicate with the physical device. (See the specific section about each device driver for details on how to use the Control Panel to send responses back to the

Option	Description
	CIM Manager.)
Standalone Mode	The Standalone mode enables you to input manager specific commands and execute them directly to the controller without any intervention from other system components.

Device Driver Loading Options

Normally device drivers are started from the Loader program that reads device driver command lines from an INI file. For example:

```
Load2=..\BIN\ACLDriver.EXE ACLVD1.INI 21 /COM:1 /C
Load3=..\BIN\CNCDriver.EXE CNCVD1.INI 23 /COM:2 /C
```

If you are activating a device driver in standalone mode, cancel the TCP/IP Network Messaging between the device driver and the manager. To do so:

From the Device Driver Control Panel, select OPERATION MENU. Select TCP/IP status and click UNABLE.

Once a device driver has been loaded, you cannot change its control mode. You must exit the device driver and restart it in the desired mode.

Option	Description
Real Mode	This mode is the default if no special mode switches are specified on the command line that invokes the device driver.
Simulation Mode	Use the `/SIMULATION` switch on command line that invokes the device driver.
Manual Mode	Use the `/Com:0` switch on command line that invokes the device driver.
Standalone Mode	The Standalone mode enables you to input manager specific commands and execute them directly to the controller without any intervention from other system components.

If a device driver is unable to open an RS232 port in order to communicate with its device, it displays the following message in the Control Mode box:

```
Cannot Open Com:n
```

This error message indicates that the device driver could not open the serial port on the Station Manager PC. Possible causes include:

- The port is in use by another application.
- The port number is invalid.
- One of the serial port parameters is invalid.

CNC Device Driver

The CNC Device Driver interfaces between the OpenCIM system and various types of machines such as a lathe or milling machine.

It receives command messages from the CIM Manager, adjacent robots, and other CIM devices. In response, it runs the corresponding CNC script program which operates the machine. The device driver responds with status messages about the CNC machine to CIM elements which have registered for these messages.

The CNC Device Driver performs the following functions:

Device Driver Function	By Using
Sends commands to a CNC machine	⇒ the CNC Script Interpreter
Tests and debugs Command Interpreter programs	⇒ the CNC Control Panel
Sends CNC status messages to other CIM elements	⇒ the OpenCIM Network
Loads G-code programs into a CNC machine	⇒ an RS232 connection

The CNC Device Driver controls machines connected in either of the following ways:

- A machine connected to an internal I/O controller board in the Station Manager PC
- A machine connected to an ACL controller

An internal I/O board maps 16 CNC control lines to two output ports on the PC. It also maps 16 CNC status lines to two input ports on the PC. These I/O port addresses are stored in the file CNCVD1.INI.

A CNC machine that receives commands via an RS232 interface can be connected to a serial port on an ACL controller. You can write an ACL program to control the machine and activate this program by sending commands to the ACL device driver using the CNC script language.

Running the CNC Device Driver

This section describes how to run the device driver. It includes the following sections:

- Loading the CNC Device Driver
- CNC Device Driver Status Window
- Generating a CNC Log File

Loading the CNC Device Driver

CNC device drivers are loaded automatically by the DD Loader, DDLoader.EXE. This program loads all device drivers from command lines found in the [Loading] section of the local INI file. To manually start a CNC device driver from the Program Manager (e.g. to run the CNC Control Panel for troubleshooting), select the icon for the appropriate CNC machine.

The following examples assume that the CNC device driver is being invoked from the [Loading] section of the CNCVD1.INI parameter file.

For example, to run the CNC Device Driver for CNC machine # 23, use:

Load4=..\BIN\CNCDriver.EXE CNCVD1.INI 23 COM:3

CNC Device Driver Status Window

The Status window appears while a device driver is running. It displays status and error messages, debug information, and output from the following sources:

 CNC script programs

 ACL programs

 Scorbase programs

 Quality control devices

 PLC programs

Generating a CNC Log File

The CNC log file facilitates debugging and troubleshooting. It captures the results of each operation performed by the device driver that are displayed in the Status window. This information is written to a file called CNC_`DeviceID`.PRT where `DeviceID` is the 3-digit device ID number of the CNC machine (e.g. CNC_023.PRT).

```
17:21:48.72 PULSBIT( 0x500, 0x0, "00001000", 500 )
17:21:49.33 --- OPEN DOOR ---
17:21:49.44
17:21:49.44 --- DOOR IS OPENED ---
17:21:49.55 WAITBIT( 0x500, "00000010", 10000 )
17:21:49.66 --- Condition is true ---
17:21:56.52 PULSBIT( 0x500, 0x0, "00010000", 500 )
17:21:57.13 --- CLOSE DOOR ---
17:21:57.23
17:21:57.23 --- DOOR IS CLOSED ---
17:21:57.34 WAITBIT( 0x500, "00000100", 10000 )
17:21:57.45 --- Condition is true ---
```

Figure 9-3: Sample CNC Log File

The log information includes each CNC script command executed, display messages, OpenCIM network messages received and sent, and error messages. Even messages that have scrolled off the screen are recorded. Each entry in the log file is time stamped to the nearest 1/100 of a sec.

The log file is placed in the same directory as the CNCVD1.INI file after the device driver has been closed.

The log file is saved as ASCII text. You can use any text editor to examine and print its contents.

&⁀ Note *The log file is overwritten each time you save it. If you want to preserve the previous contents, rename the file CNC_`DeviceID`.PRT first.*

Downloading G-Code

A CNC device driver downloads a G-code file to a machine in response to a command from the CIM Manager (in preparation for running a CNC process).

The device driver uses one of the following download mechanisms depending on which you specify:

- A download utility that you supply called from a specified batch file (recommended)
- The built-in downloader of the CNC device driver

This section discusses how to use a utility program that you supply to download G-code. This method is usually preferable to using the device driver's internal downloader because a utility program provided by the CNC manufacturer can take advantage of all of a machine's features (e.g. providing error correction during the download).

The commands required to invoke a machine's downloader are inserted into a DOS batch file. The last command in this batch file creates a flag file which signals that the download is complete. The CNC device driver automatically deletes this flag file each time it invokes the batch file.

To build this batch file and direct the device driver to use it, do the following:

❶
❷
❸
Procedure

Creating a Utility for Downloading G-code

1. Write a batch file named `CNC_L.BAT` which calls the utility downloader as shown in the following example:
   ```
   DLOADG.EXE %1 %2
   ECHO Task Loaded > <%project
   directory%>\WS3\TASK.CNC
   ```

2. When the CNC device driver calls this batch file, it specifies the following two parameters (which appear on the first line of the batch file above):

   ⇒ The G-code file to download (which includes the full DOS path)

   ⇒ The memory region within the CNC's RAM that stores this G-code program.

3. Define the following parameter entries in the `[CNCDriverDefinitions]` section of the INI file for this CNC device driver:

   ```
   Loader          = <%project
   directory%>\WS3\CNC_L.BAT
   TaskLoadedMark  = <%project
   directory%>\WS3\TASK.CNC
   ```

   When these parameters are defined, they tell the device driver to use the specified download utility program instead of its internal downloader.

The CNC Control Panel

The CNC Control Panel is a feature of the CNC Device Driver that allows you to perform the following functions:

- Run CNC programs interactively.
- Control CNC operations by setting bits on the Output Panel.
- Read the status of the CNC machine by examining bits on the input panel.
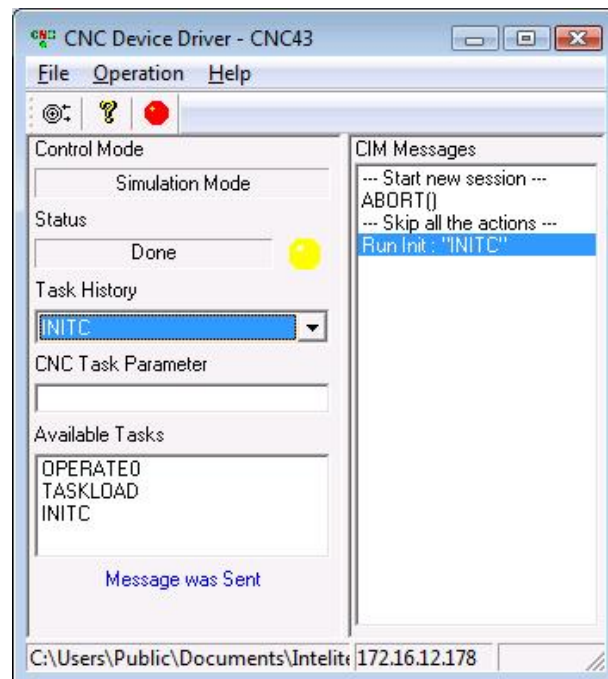
Figure 9-4: CNC Device Driver Control Panel

This section includes the following sections:

- Running CNC Programs Interactively
- PC-Input/Output Panel
- Output Panel
- Input Panel
- Program History List
- Closing the Control Panel

Running CNC Programs Interactively

The CNC Control Panel allows you to run CNC programs created with the CNC Script Interpreter and view the results.

To run a Command Interpreter program using the Control Panel, do the following:

❶
❷
❸
Procedure

Running a Command
Interpreter Program

1. Determine the parameter(s), if any, that are to be passed into the program. Type these value(s) in the box labeled CNC Command Parameters (e.g. `500` for a 500 millisecond delay)

2. Use the mouse to scroll through the CNC Command List in order to find a program. Double-click on a program name to run that program.

If the CNC machine is connected to the I/O board in the PC, you can observe the effects of a program by observing the line indicators in the PC-Inputs and PC-Outputs panels in the Control Panel. In addition, status messages and instructions generated by the program appear in the Status window.

PC-Input/Output Panel

To display the PC-Input/Output Panel, select from the main menu Operation | Input/Output.

Output Panel

The Output panel allows you to observe and set the state (On or Off) of 16 CNC control lines. Control lines may be hard-wired to specific CNC functions. Alternatively, the CNC machine may be configured to activate a certain G-code program associated with a control line. Check the documentation of your CNC machine for specifics on the use of each control line.

The PC-Output panel allows you to set CNC control lines on and off by clicking bits in the appropriate output port. The layout of the panel reflects the way the control lines are mapped to bits in two designated PC output ports. By using a mouse to click on the bits in the panel, you can:

- Toggle the state of a control line.
- Pulse a control line by clicking its bit, waiting the desired interval, and clicking it again to restore it to its original state.

Input Panel

The Input panel allows you to observe the state (On or Off) of 16 CNC status lines. Status lines may be hard-wired to specific CNC components. Alternatively, the CNC machine may use a G-code program to set the state of a status line. Check the documentation of your CNC machine for specifics on the meaning of each status line. The layout of the panel reflects the way the status lines are mapped to bits in two designated PC input ports.

The PC-Input panel displays the state of CNC status lines. It cannot be used to change the value of a status line. Only the CNC machine can change the value of a status line.

Program History List

You can view a list of programs that were run by clicking on the drop-down list box labeled Task History.

Closing the Control Panel

It is a good idea to close the Control Panel in order to prevent others from tampering with it while the CNC machine is active. Use one of the following standard Windows methods for closing a window:

1. Double-click the control bar in the X button of the Control Panel window.

2. Click the control bar and select **Close**.

3. Press **[Alt + F4]** while in the Control Panel window.

Warning!
You should always close the Control Panel if you are going to leave the CIM unattended. Otherwise someone might cause damage if they inadvertently activate a machine (such as a CNC machine) by making selections on the Control Panel (e.g. by clicking on the Output panel or by selecting a task which starts the machine).

Robotic Device Drivers

The robotic device driver relays messages between the OpenCIM network and the devices attached to a robotic controller such as:

- An Intelitek robot
- An automatic screwdriver
- A barcode scanner
- An X-Y table
- ASRS-36
- ASRS-36U

This device driver receives command messages from the CIM Manager and adjacent CNC machines (for ACL only). In response, it runs the corresponding robotic program. The robotic device driver communicates with the controller using an RS232 port on the Station Manager PC.

The robotic device driver performs the following functions:

- Activates robotic programs.
- Receives status messages from robotic programs and relays them to a CIM entity.
- Allows you to interactively operate robots and peripheral devices attached to a robotic controller.
- Allows you to test and debug robotic programs by sending commands from the Control Panel.
- Emulates a robot in Simulation mode.

The primary robotic command is to run a set of pick-and-place programs which direct a robot to move a part from one location to another at a station. Robotic ACL or Scorbase programs can also direct a robot to perform other tasks such as assembly operations or they can control peripheral devices such as the Rotary Table.

Robotic Device Drivers User Interface

The robotic device drivers user interface enables you to perform the following functions:

- Simulate a robot.
- Issue robotic commands interactively.
- Debug pick-and-place programs by running them interactively.
- Test a robot and its positions and programs by issuing a series of pick-and-place commands stored in file.
- Create a file of pick-and-place commands.

Robot Operation

The ACL and Scorbase device drivers contain different interfaces for operating the robot, each of which is described in the following sections.

- Scorbase Device Driver
- ACL Device Driver

Scorbase Device Driver

Since the Scorbase device driver is part of the Scorbase software, you can use Scorbase software functionality in order to operate the robot. The Scorbase device driver window (for OpenCIM) is displayed, as follows:
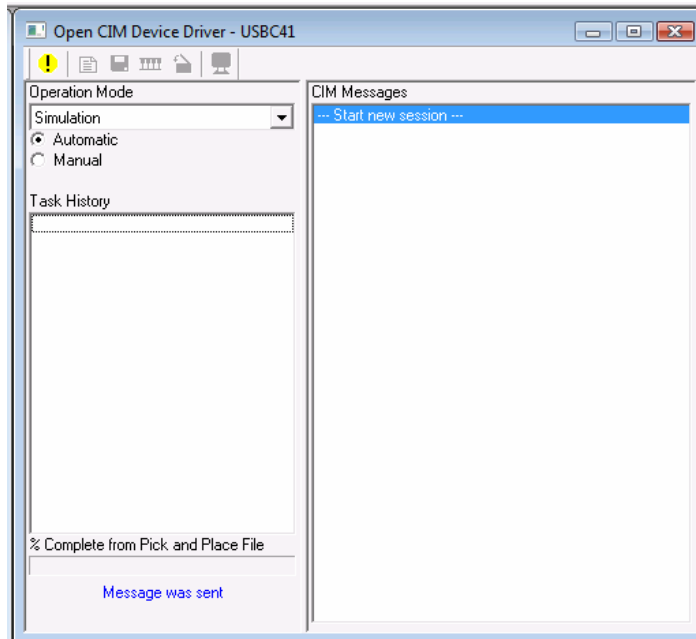
Figure 9-5: Scorbase Device Driver (for OpenCIM)

You can select one of the following modes: Online Mode, Simulation Mode (Automatic or Manual) and Standalone Mode. For further details refer to Control Modes for the Robotic Device Driver in the following section.

Reloading Last Project at Startup

When loading the Scorbase device driver for the first time, perform the following procedure enabling you to relaod the last project the next time you activate the Scorbase device driver.

To reload the last project at startup:

1. From your windows start menu, click the **Loader** icon of your workstation
    . The CIM DDLoader window is displayed.
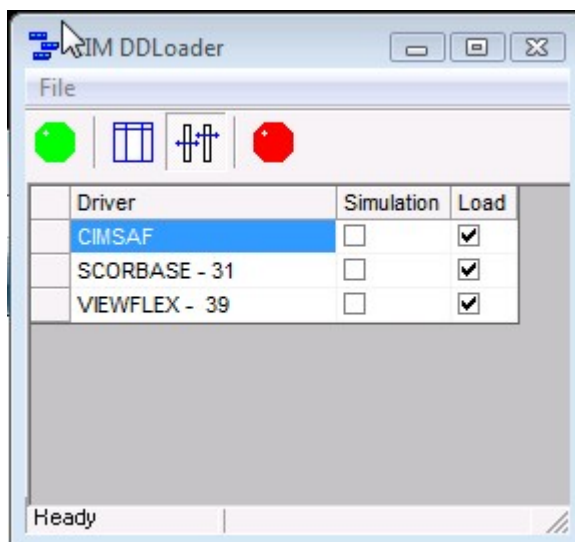
Figure 9-6: CIM DDLoader

2. Ensure the **Load** column is selected ☑ for the Scorbase device driver and click the

   Load Selected Drivers ⬤ icon. The Scorbase device driver window is displayed.
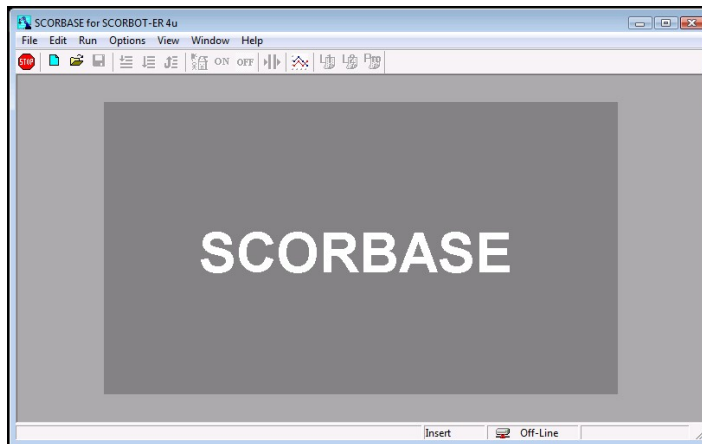
Figure 9-7: Main Scorbase Device Driver Window

3. From the **File** menu select **Open Project**. The Load Project window is displayed.

4. Select the required project in accordance with your OpenCIM workstation configuration and click **OK**. The project is displayed in the main window.

From the **Options** menu, select **Reload Last Project at Startup**. The next time you access the **Scorbase Device Driver**, this project will automatically be displayed.

ACL Device Driver

You can use the ACL device driver, as a limited terminal to send commands to a controller. Commands that you type in the Send to Controller field are sent out the PC's serial port when you press Enter. Responses from the controller are displayed in the status window of the device driver. This capability is useful for testing and debugging individual ACL programs. The ACL device driver window is displayed, as follows:
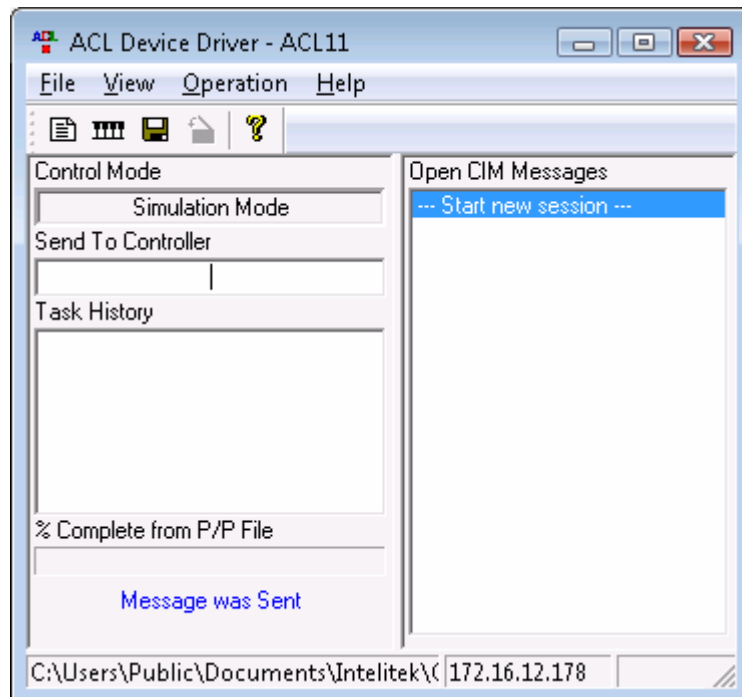
Figure 9-8: ACL Device Driver

Control Modes for the Robotic Device Driver

The robotic device driver, like all other device drivers, can pass actual messages or can generate simulated messages. In Real Mode, the device driver relays messages between the OpenCIM system and a robotic controller. In one of the simulation modes, the robotic device driver can be used to emulate a robot or to test a robot.

The following list shows the messages that can appear in the Control Mode box on the Control Panel. In each mode, the device driver treats command messages the same whether they originate from the CIM Manager or from selections you make from the device driver's Control Panel.

The activation examples show command lines from the Loader's INI file used to start the device driver in the designated mode. Bold command line switches highlight the specific switch used to invoke that mode.

Option	Description
Real Mode (Online)	Normal operating mode. The device driver relays command messages to the robotic controller from the CIM Manager, CNC machines, etc. In turn, it broadcasts status messages from the controller to the OpenCIM network.
	Command line for ACL: `ACLDriver.EXE ACLVD3.INI 31 /COM:1`
	Command line for Scorbase: `/O /I=ACLVD1.INI /N=11 /CIMDD_ONLINE`
Real Mode: Connected OK	The robotic device driver shows that it has received the first message from the robotic controller on the serial port.
Cannot Open Com:n	The robotic device driver could not open its serial port on the Station Manager PC.
Simulation Mode (Automatic)	The robotic device driver receives commands as usual but emulates a robot and a barcode reader by generating status messages automatically.
	In this mode, the device driver does not actually communicate with the robotic controller; only with the CIM Manager (and other devices that send it commands).
	Command line for ACL: `ACLDriver.EXE ACLVD3.INI 31 /COM:1 /SIMULATION`
	Command line for Scorbase: `/I=ACLVD1.INI /N=11 /CIMDD_SIMUL_AUTO`
Simulation Mode (Manual)	The robotic device driver receives commands as usual but only generates status messages when you double click on a line in the Task History box.
	In this mode, the device driver does not actually communicate with the robotic controller; only with the CIM Manager (and other devices that send it commands).
	Command line for ACL: `ACLDriver.EXE ACLVD3.INI 31 /COM:0`
Standalone Mode	The Standalone mode enables you to input manager specific commands and execute them directly to the controller without any intervention from other system components.

⚠ Warning!

*The ACL and Scorbase device drivers enable controlling the robot from both the OpenCIM Manager and Station PC simulataneously. This is useful for development and testing procedures of the robotic programs. However,it can be dangerous when operating the CIM system in online mode. Therefore you should always close the Control Panel to prevent others from tampering with it when the robot is truned on. Otherwise someone might cause damage if they inadvertently activate the robot by making selections from the robotic device driver window (for example, by clicking on the **Play From P/P File** button) .*

Task History List

The Task History box shows the last several commands sent to the controller. You can scroll the background to see commands that have scrolled off the screen.

Testing Pick-and-Place Commands

The pick-and-place buttons of the Control Bar allow you to send commands to the robot to move parts around the station. (These options are available in stand alone mode only when using the Scorbase device driver.)

A brief description of the each these pick-and-place buttons is described in the following table:

Option	Description
Enter P/P Command	Allows you to manually send a pick-and-place command to the robot instead of the command being sent by the CIM Manager. Selecting this button presents you with the Run 'Pick-and-Place' dialog box .This dialog box requests the following six parameters:
	Part ID - Number of the part/template to be moved (template = 0)
	Source ID - Device ID of source location where the robot is to pick up the part/template. Located next to the Source ID field is a drop-down list that allows you to select the Source ID by name and not by index.
	Source Index - Compartment number if source location is divided into cells. Located to the right of the Source Index field is another field that displays the number of the compartment.
	Target ID - Device ID of target location where the robot is to place the part/template. Located next to the Target ID field is a drop-down list that allows you to select the Target ID by name and not by index.
	Target Index - Compartment number if target location is divided into cells. Located to the right of the Target Index field is another field that displays the number of the compartment.
	N**ote** - Can be used to send special instructions to assembly programs or user-developed programs.

Option	Description

⌒ Note

The names and the range that appear in the Pick-and-Place dialog box are taken from the INI file used by this device driver.
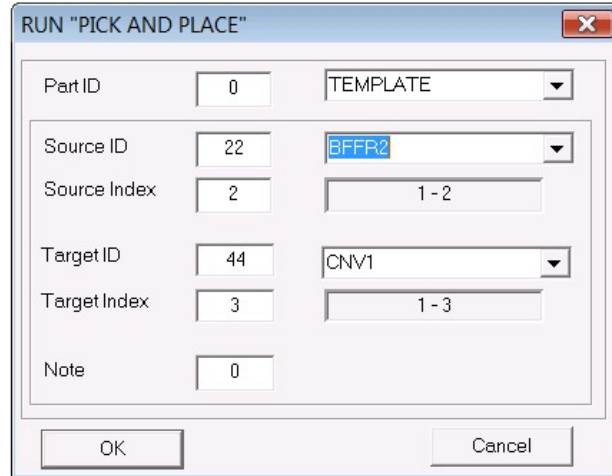
Figure 9-9: Run 'Pick-and-Place' Dialog Box

Play from P/P File

▥

Begins executing a series of pick-and-place commands stored in a special text file designated for this robot, ACL_*xxx*.PNP. The *xxx* is the device ID of this robot (found in the title bar of the Control Panel). These commands are sent one at a time.

This file can be used to thoroughly test a robot's ability to retrieve and deliver parts from every device at a station. Pick-and-place commands involving every device can be stored in the pick-and-place file. Playing this file would then allow you to observe if the robot was able to properly access every device.

% Complete from P/P File

This display activates when you select the Play button. It shows what percentage of the pick-and-place file has already been executed.

Save P/P Cmds

💾

Saves a pick-and-place text file containing all pick-and-place commands found in the Task History box. This file is saved in the current working directory under the name ACL_*xxx*.PNP.

Close P/P File

Terminates the playback of a pick-and-place file.

Quality Control Device Drivers

A quality control device driver interfaces between the OpenCIM network and a quality control device such as the ROBOTVISIONpro Laser scan meter.

A device driver communicates with a QC device using an RS232 connection.
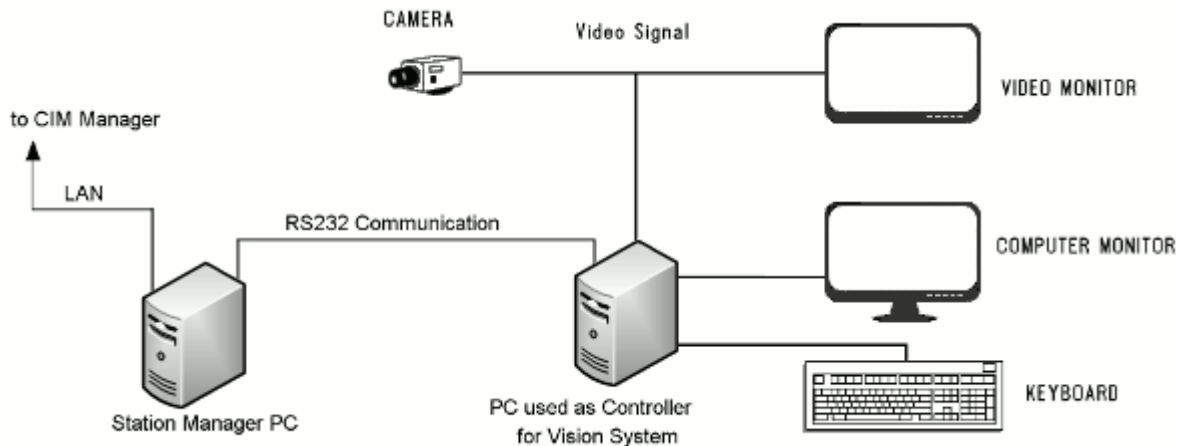
Figure 9-10: A QC Device Driver Passing Messages to and from a QC Device

The QC device driver receives a command message from the CIM Manager specifying the type of quality control test to run. It then performs the following steps:

1. Activates the specified test on the QC device (e.g. a Scan command for a ROBOTVISIONpro system).

2. Receives the test result from the device.

3. Compares the result to a range of acceptable values specified in the command message.

4. Sends a pass/fail status message back to the CIM Manager.

If the result is fail, the CIM Manager:

- Disposes of the defective part as specified by an ONFAIL process in the Part Definition table.

- Automatically reproduces the part.

Each QC test is defined as a separate process in the Machine Definition module. The test type and acceptable range of test results may be specified in the Parameters field of the Part Definition table.

If the quality control device is connected to an ACL controller (e.g. a barcode reader), the name of the ACL program that activates this device should be specified in the Program field of the Machine Process table in the Machine Definition module.

☞
Tips

It is possible to use a robot as a quality control device. You can write special ACL programs which perform quality control tests such as:

- Have a robot try to place a part in a mold. If the part is too big, you can detect the collision. If it is too small, you can detect free play when the robot tries to move the part after it has placed it in the mold.

- You can measure the dimensions of a part by reading the span of the robot's gripper when it is holding the part.

The following table shows how to set up test parameters for each type of QC device driver:

QC Device	Test Type	Acceptable Range of Values
ROBOTVISIONpro Camera	The system scans a part looking for an object(s) that was defined as a Pattern ID in the ROBOTVISIONpro software. (e.g. Are three screws in place?)	The number of objects the ROBOTVISIONpro system should find. If the minimum and maximum values are the same, the system must find this exact number in order for the part to pass the test.
Laser Scan Meter	Checks the diameter of a cylinder. Test type = 1.	Minimum and maximum values represent the tolerance surrounding the desired diameter.

There is a customized version of the quality control device driver for each of the quality control devices listed above. Since these three device drivers are essentially similar except for the internal message format used to communicate with the quality control device, this section discusses the operation of all three. In this discussion, these device drivers are interchangeable and are referred to simply as the quality control device driver. All QC device drivers "look" the same to the CIM Manager, i.e. it sends the same type of command message to each type and receives the same type of pass/fail status message from each.

A quality control device driver performs the following functions:

- Activates a test on a quality control device.
- Receives status messages from a quality control device and sends them to the CIM Manager.
- Allows you to test and debug a quality control device by sending commands from the Control Panel.
- Emulates a quality control device in Simulation mode.

The QC Control Panel

The QC Control Panel is a feature of the QC device driver. It allows you to perform the following functions:

- Determine the control mode the device driver is running in.
- Simulate the test results from a QC device.
- Monitor test results in real - time.
- Test the QC device by manually issuing command messages to it and observing the results.
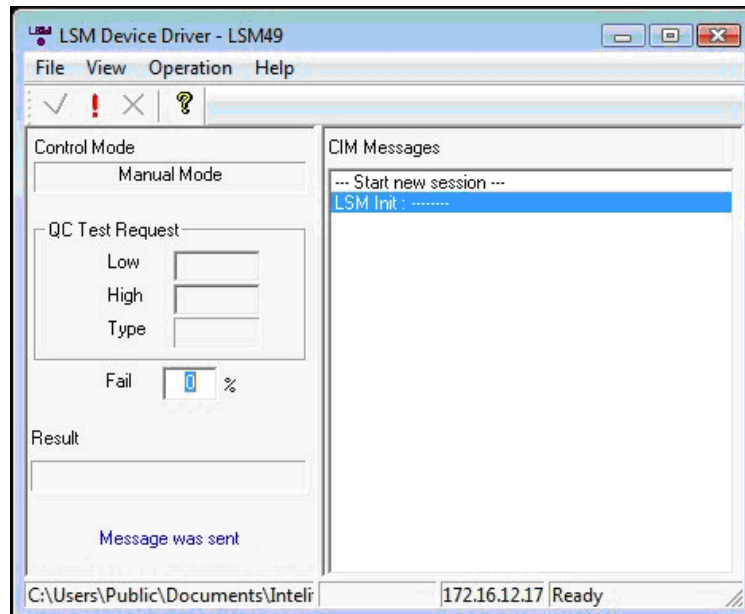
Figure 9-11: Control Panel for a Quality Control Device Driver

Control Modes for Quality Control Device Drivers

A QC device driver, like all other device drivers, can pass actual messages or can generate simulated messages. In Real Mode, the device driver relays messages between the OpenCIM system and the quality control device. In one of the simulation modes, the QC device driver can be used to emulate a quality control device or to test a device. For more details, refer to Device Driver Loading Options.

The table below shows the messages that can appear in the Control Mode box on the Control Panel. In each mode, the device driver treats command messages the same whether they originate from the CIM Manager or from selections you make from the device driver's Control Panel.

The activation examples show command lines from the Loader's INI file used to start the device driver in the designated mode. Bold command line switches highlight the specific switch used to invoke that mode.

Option	Description
Real Mode	Normal operating mode. The device driver relays command messages to the QC device from the CIM Manager. In turn, it broadcasts pass/fail status messages from the device to the OpenCIM network. **Activation:** LSMDriver.EXE LSMVD1.INI 13 /COM:2
Real Mode: Connected OK	The QC device driver shows that it has received the first message from the QC device on the serial port.
Cannot Open Com:n	The QC device driver could not open its serial port on the Station Manager PC.
Simulation Mode	The QC device driver receives commands as usual but emulates a quality control device by generating pass/fail status messages automatically based on the value in the **Fail %** field. In this mode, the device driver does not actually communicate with the QC device; only with the CIM Manager. **Activation:** LSMDriver.EXE LSMVD1.INI 13 /COM:2 **/SIMULATION**
Manual Mode	The QC device driver receives commands as usual but only generates pass/fail status messages when you click on the **Success** or **Fail** buttons. In this mode, the device driver does not actually communicate with the QC device; only with the CIM Manager. **Activation:** LSMDriver.EXE LSMVD1.INI 13 **/COM:0**
Standalone Mode	The Standalone mode enables you to input manager specific commands and execute them directly to the controller without any intervention from other system components.

Controlling a QC Device from the Control Panel

The buttons on the Control Panel allow you to send commands to the quality control device and status messages to the CIM Manager. These buttons are described below.

Option	Description
Check	Activates a test on the quality control device. This button only functions in Real Mode. It is useful for testing communications with the quality control device. The response from the device appears in the device driver's Status window.
	This button resends the last command message from the CIM Manager as shown in the fields Low, High, and Type described below. If no command message has yet been received, the default values are:
	Type = 1, High = 0, Low = 0
Success	Generates a status message to the CIM Manager indicating that a part passed its quality control test. Used in Manual mode.
Fail	Generates a status message to the CIM Manager indicating that a part failed its quality control test. Used in Manual mode.

The following fields on the Control Panel show the parameters associated with the last command that was sent to the quality control device. These values are used when you select the Check button (described above) to manually send a command to the device.

Option	Description
Low	The minimum acceptable test value.
High	The maximum acceptable test value.
Type	An ID number specifying which sort of quality control tests the device should perform.

The Fail % field allows you to control how the device driver responds when it is operating in a simulated mode:

Option	Description
Fail %	Used only in Simulation mode. Randomly determines how often the simulated test result will be Success or Fail (0% = always successful, 100% = always failure). Pass/fail results are generated randomly.
	The default failure percentage is read from the parameter SimulationFailPercent in the device driver's INI file. Changing this value on the Control Panel affects the current session but does not save the new value to the INI file.

QC Device Settings

Each quality control device driver uses a duplicate set of INI file parameter settings shown in the following figure(Figure 9-12). These settings relate to:

- Format of a quality control log file.
- Running the device driver in Simulation mode.
- RS232 communication settings.
- Appearance of the device driver's Control Panel on screen.

When you want to simulate the operation of a QC device, the QC device driver provides simulated test results to the CIM Manager stating whether a part passed or failed a QC test. The parameter *SimulationFailPercent* allows you to set the default failure percentage that a QC device driver uses when running in Simulation mode.

The following sections discuss particular settings for each type of quality control device driver.

ROBOTVISIONpro Settings

The ROBOTVISIONpro device driver works best with v2.3 or later of the ROBOTVISIONpro software. Use the parameter *Snap = Yes* to indicate version 2.3 or later.

You can use the *Frame* parameter to specify the frame area in the camera's field of view where the ROBOTVISIONpro system should scan for objects.

When you train the ROBOTVISIONpro to recognize a new object, the ROBOTVISIONpro software assigns a unique Pattern ID. Use this number as the test type when setting up the Parameter field in the Part Definition table.

Laser Scan Meter Settings

A laser scan meter is a straightforward quality control device that performs only one type of test. Use a test type of 1 for this device when entering values in the Parameter field of the Machine Process table or the Part Definition table.

```
[General]
CimSetupPath=C:\OPENCIM\SETUP\SETUP.CIM

[LSMDriverDefinitions]
QCReport = Yes
QCReportTemplateFile = VC2_QC.INI
QCReportFileName      =
QCReportFileMarker    =
QCReportFileDeleteOnStart =
SimulationFailPercent = 20
BaudRate=9600
Parity=None
DataBits=8
StopBits=1
XonXoff=No
MainWindowBkgndColors=40,150,100
MainWindowTextColors=100,50,200
```

Figure 9-12: Sample INI File Settings for a Laser Scan Meter

ViewFlex Device Driver

The ViewFlex Device Driver interfaces between the OpenCIM network and the Vision Machine System as a quality control device.

Each quality control test is defined as a separate process in the Machine Definition module. The quality control test consists of three parts:

- File – the Script File (.bas) that contains the Program (Function) to be executed.
- Program – the Function from the File that returns the results of the quality control process to the OpenCIM Manager as Pass/Fail or Error.
- Fail %. Simulate test results (only in Simulation Mode) by determining the Pass/Fail according to the percentage of failure that was input.
- The ViewFlex Device Driver performs the following functions:
- Activates a test on the Vision Machine System.
- Receives status messages from the Vision Machine System and sends to the OpenCIM Manager. (OpenCIM Messages in the ViewFlex.)
- Allows you to test and debug the Vision Machine System by sending commands from the Windows dialog box.
- Emulates the Vision Machine System in Simulation mode.

Adjustments

Adjustments can be performed ViewFlex.ini files. Each of which is described below.

User Interface
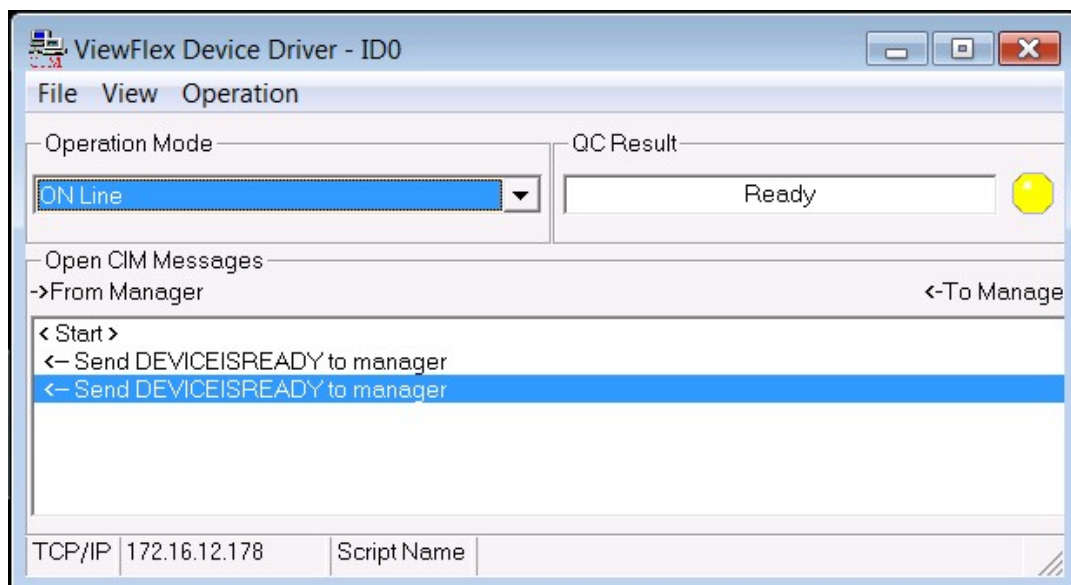
Figure 9-13: ViewFlex Control Panel

Control Modes

The control modes that the ViewFlex Device Driver can run in are:

On-Line (Real): Allows the ViewFlex Device Driver to wait for the following commands from the OpenCIM Manager: Snap, Load Script, Execute Program Script, and Send Results.

Manual: Allows you to test the ViewFlex Device Driver by manually issuing the commands Pass/Fail and observing the results.

The following buttons are available only in Manual Mode, after a request from the CIM Manager, has been issued:

Option	Description
✔	**Send Pass**.
✖	**Send Fail**.

- **Simulation**: Allows you to simulate test results by determining the Pass/Fail according to the percentage of failure programmed into the CIM Manager.
- **Debug:** Allows you to inspect every step of the script running the ViewFlex Device Driver, by simulating what will happen when in Real Mode.

The following buttons are available only in Debug Mode:

Option	Description
🗁	**Open File**: loads the script.
⬇	**Run Script**: runs the selected script.
⬇	**Step-By-Step**: runs each command line of the script individually.

Sample Script

A model is the pattern for which you are searching, and the image from which it is extracted. The following script is a sample for testing Pass/Fail models. The program attempts to find the "x" object (X.mod). If found, Fail is sent (QCR="Fail"). If the object is not found, the program attempts to find the "v" object (V.mod). If found, Pass is sent (QCR="Pass"). If neither is found, Error is sent (QCR="Error").

```
Attribute VB_Name = "Module1"
Function QCR() As String
Dim NumberFound As Integer
Snap1_1$=Insptr.ImgGetCur
Insptr.ImgSetCurrent Snap1_1$, R_Def$, ALL_BANDS
I_IMAGE16$ = Insptr.ImgConvertType(TO_8U)
Insptr.ImgSetCurrent I_IMAGE16$, R_Def$, ALL_BANDS
B_V_BLOB1$ = Insptr.BlobLoad("O:\ER14PRO_ASSEMBLY\WS3\Viewflex\V_CHECK.bst")
Insptr.BlobSetCur B_V_BLOB1$
Insptr.BlobCount
NumberFound=Insptr.BlobCount

Debug.Print NumberFound
If NumberFound=1 Then
  QCR="Pass"
Else
```

```
        QCR="Fail"
   End If


   Insptr.CloseAll
   End Function
```

ULS Device Driver

The ULS device driver operates the ULS Laser Engraver. In the same way that the CNC device driver controls all operations of a CNC machine, the ULS device driver controls all operations of the Laser Engraver.

The ULS device driver communicates with the Laser Engraver via parallel and RS232 links. It uses the parallel connection in order to download the appropriate engraving program (CorelDraw or Freehand file) to the device. The serial RS232 is used to establish communications between the laser machine and the ULS device driver so that programs from the laser machine's memory can be selected and messages transmitted to and from the device driver.

When a part is to be placed on the laser table, the CIM Manager sends a message to the device to descend to the lowest level and then rise to the loading level. The robot always picks & places a part from/to the same level.

Warning!

Do not place any objects in the laser machine's front cabinet. When the table descends to its lowest level, it could crash into objects that may have been placed there.

Downloading Print Files

The laser has two connectors at the back: a 9-pin connector from the device driver, and an LPT port for connecting to the computer from which the CorelDraw or Freehand files are downloaded.

The laser has two methods of file storage: one file saves only one file to the memory and when a new file is downloaded it replaces the old one; when multiple files are stored in the laser memory, each file is numbered sequentially according to the download order.

Warning!

Working with multiple files in on-line mode could cause the wrong file to be activated.

In one file mode, the fileload.bat file enables the CIM Manager to automatically download the specific file that is required for the process and later activate other files as ordered by the CIM Manager.

User Interface

The Laser Engraver interface enables you to perform the following:

- Download a file to the laser machine.
- Select a file for engraving.
- Delete all engraving programs from the laser machine.
- Display the properties of the communications port.
- Create a log file of all the messages sent during a session.
- Specify the TCP/IP status between the CIM Manager and the device driver.

The device driver can be in one of three states: Ready (yellow), Disabled (red), Working (green).
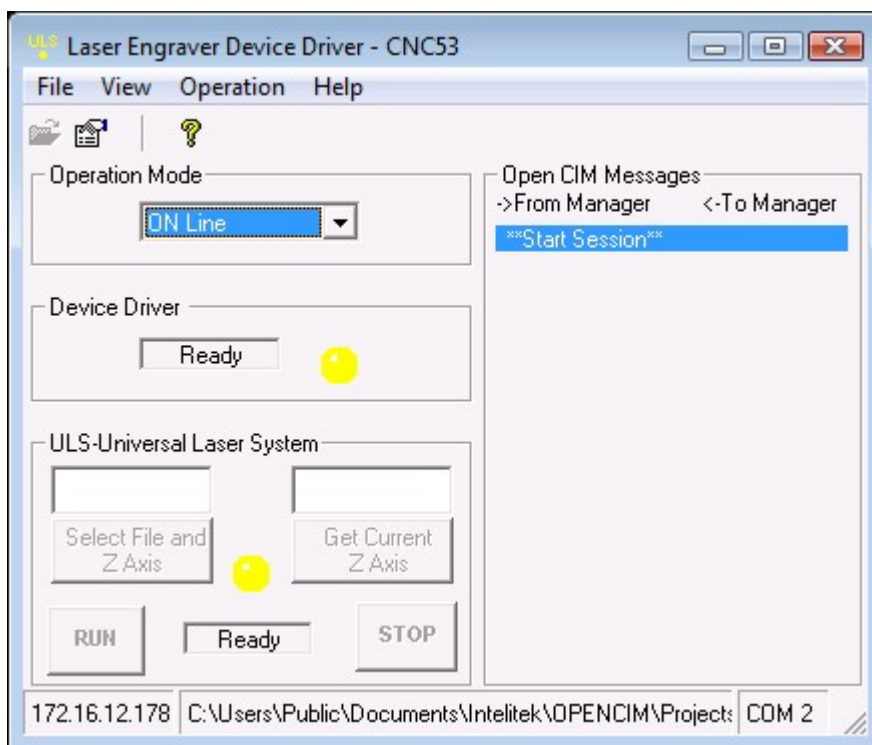
Figure 9-14: ULS Device Driver (On Line Mode)

Setting the Zero Point

When you activate the ULS device driver, you are reminded to set the machine's Z axis zero point.
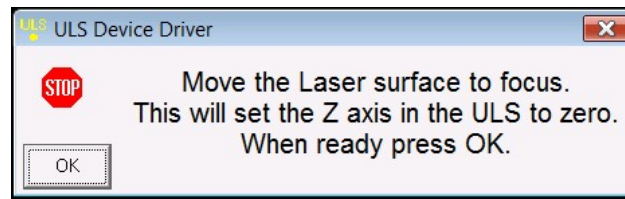
At this point, the laser's focal length should always be 2 inches. This ensures that the laser is focused and produces maximum efficiency.

To manually set the zero point:

1. Press the Z button from the main menu on the machine or from the file display menu.

2. Select the Focal Length option.

3. Place the gauge on the table (if a template is always used, place the gauge on the template).

⚠ Warning! To prevent impact, be careful not to raise the table with the gauge directly under the lens carriage.

4. Raise the table carefully to the position where the upper chamfered part of the gauge lies firmly against the side of the focus lens carriage.

5. Continue raising the table to the point where the gauge tilts outwards.

6. Gently lower the table to the *exact* point where the gauge straightens, i.e., it is aligned with the side of the carriage. This is the zero point of the laser engraver and the loading/unloading position of the robot.

7. Select the Set Focal Length option and confirm to set the zero point.

8. Once you have set the zero point, click **OK** in the ULS device driver to display the Laser Engraver Device Driver window, as shown in User Interface.

Control Modes

Like all the device drivers, the ULS device driver can operate in Real, Simulation and Manual modes; it can also be operated in standalone mode.

Option	Description
Real Mode	Normal operating mode. The device driver relays command messages to the Laser Engraver from the CIM Manager. In turn, it broadcasts start/finish/end status messages from the device to the CIM Manager.
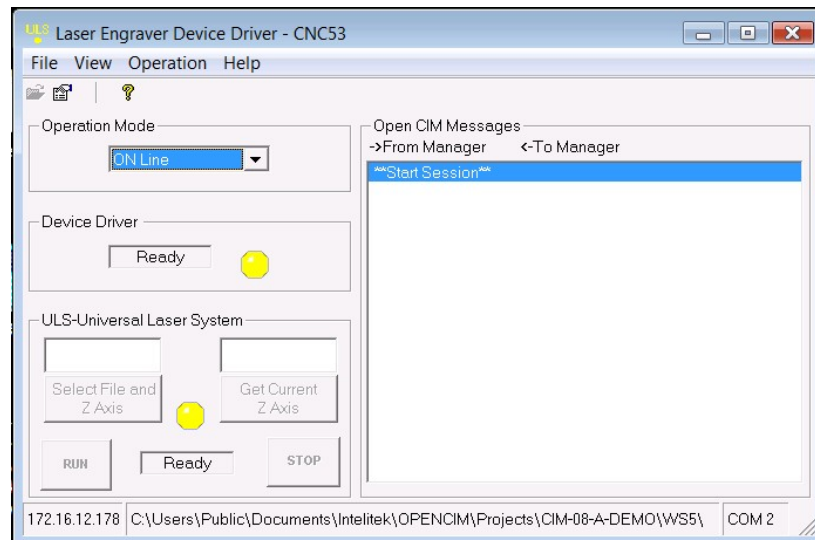
Figure 9-15: Laser Engraver Device Driver - On Line Mode

In the CIM Manager Part Definition, the user specifies the name of the engraving program as the "filename" and the Z-axis as the "parameter". The current Z-axis is displayed during operation.

Simulation Mode	The device driver emulates Real Mode by communicating with the CIM Manager. The Laser Engraver is disabled in this mode.
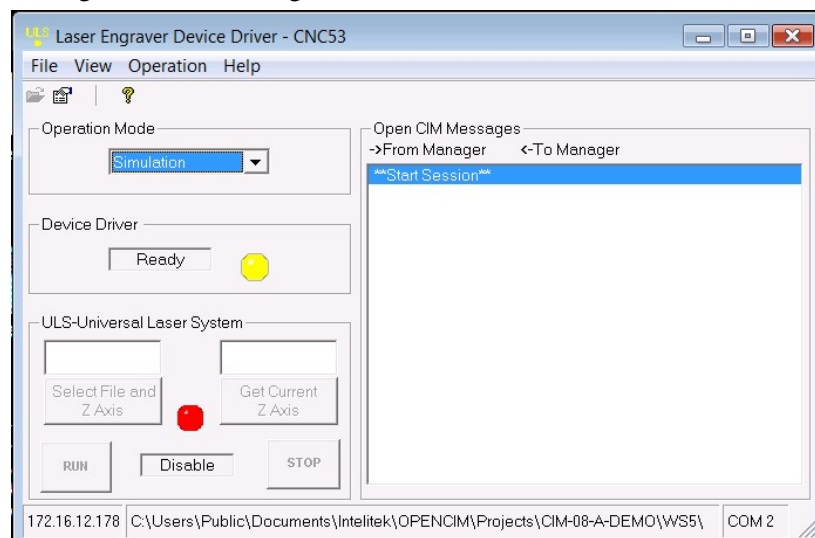
Figure 9-16: Laser Engraver Device Driver - Simulation Mode

Manual Mode   In this mode, each command received from the CIM Manager (displayed in the Action field) must be confirmed by the user (click Do).
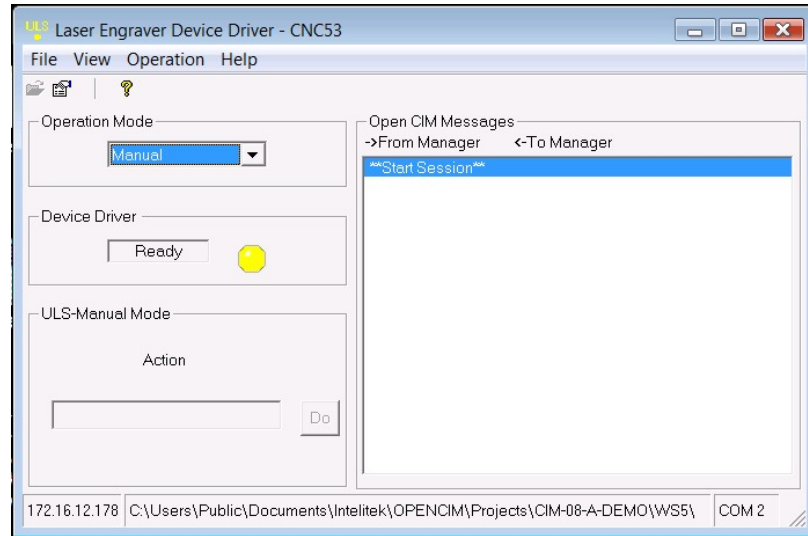
Figure 9-17: Laser Engraver Device Driver - Manual Mode

Standalone Mode   In this mode, the user is in direct control of the laser machine without any interference from the CIM Manager. It is therefore the user's responsibility to verify that the laser is operational.
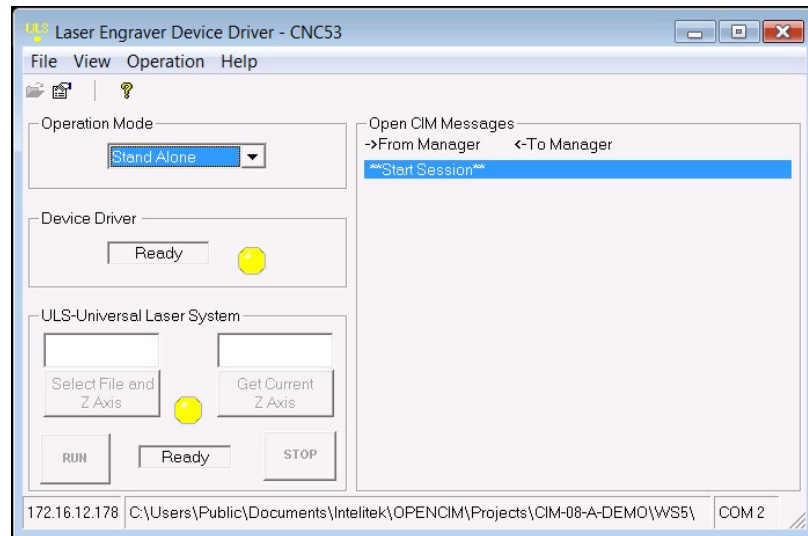
Figure 9-18: Laser Engraver Device Driver - Standalone Mode

The ▭STOP▭ button can be pressed at any stage during the operation.

BCR Device Driver

The BCR device driver operates the Bar Code Reader. In the same way that the CNC device driver controls all operations of a CNC machine, the BCR device driver controls all operations of the Bar Code Reader.

The BCR device driver communicates with the Bar Code Reader via parallel and RS232 links.

User Interface

The BCR device driver interface is accessed from the CIM DDLoader window, as described in the following procedure.
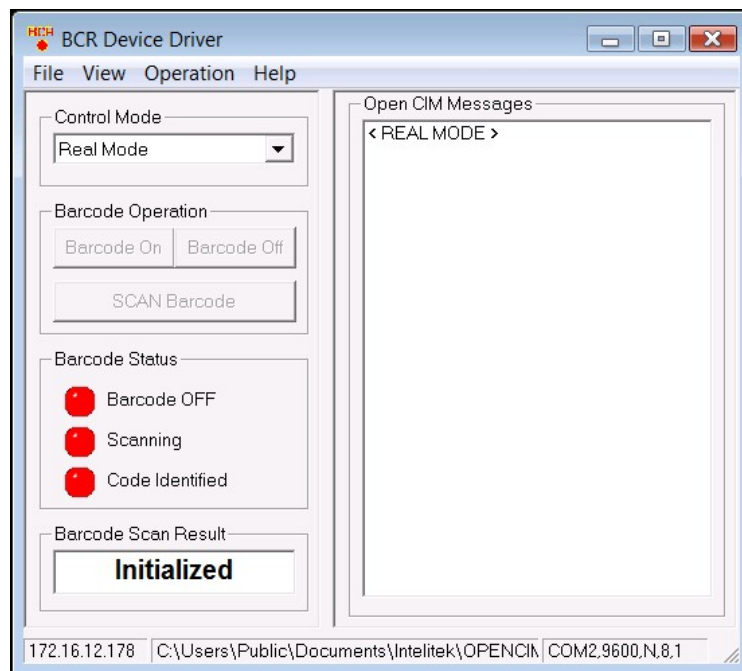
Figure 9-19: BCR Device Driver

To access the BCR device driver user interface:

1. Select the Loader icon of the workstation where the BCR device driver is installed. The CIM DDLoader window is displayed, as shown in Figure 9-2: DD Loader .

2. To operate the device driver in Real Mode check the **Load** column of the **BCRDRIVER**. To operate the BCR device driver in Simulation mode check the **Simulation** column of the **BCRDRIVER** and then click the **Load Selected Drivers**

   icon to load the BCR device driver software. The various modes of operation are described in the following section.

Control Modes

Like all the device drivers, the BCR device driver can operate in Real, Simulation and Manual modes. It can also be operated in Standalone mode.

Real Mode   Normal operating mode. The device driver relays command messages to the Bar Code Reader from the CIM Manager. In turn, it broadcasts start/finish/end status messages from the device to the CIM Manager.
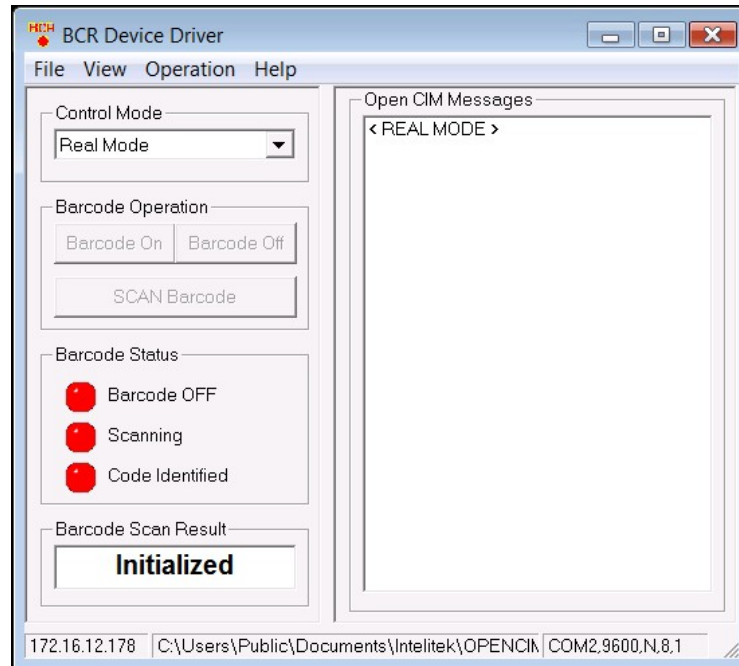
Figure 9-20: BCR Device Driver (Real Mode)

To operate the BCR device driver in Real Mode:

- From the Control Mode drop down list, select **Real Mode**. The result is displayed in the Barcode Scan Result area.

Simulation Mode
The device driver emulates Real Mode by communicating with the CIM Manager. The Bar Code Reader is disabled in this mode.
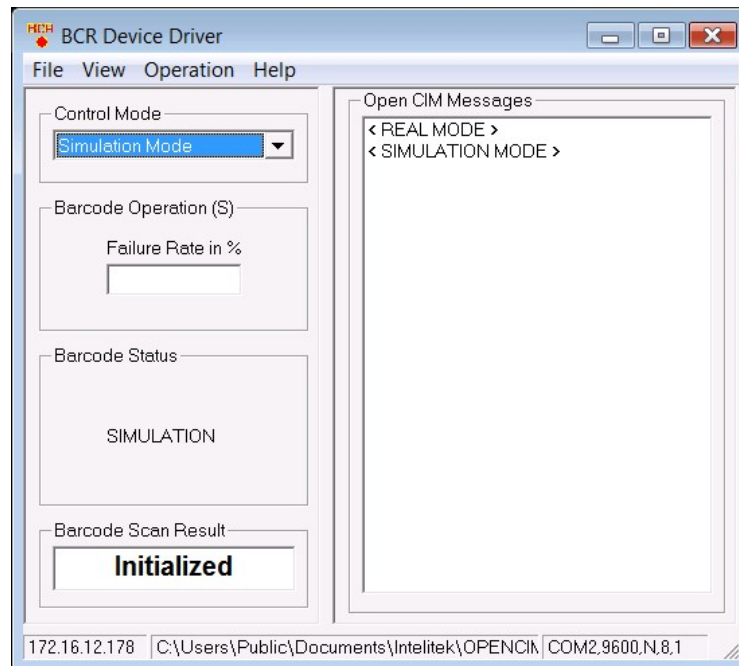
Figure 9-21: BCR Device Driver (Simulation Mode)

To operate the BCR device driver in Simulation Mode:

1. From the Control Mode drop down list, select **Simulation Mode**.

2. In the Barcode Operation area enter the failure rate percentage. For example, enter 50 to randomly determine a 50 percent failure rate. (meaning, 50 percent of the parts will pass the BCR test and 50 percent will fail).

Manual Mode   In this mode, the Bar Code Reader recives commands as usual from the CIM Manager, but only generates Pass/Fail status messages when you click on the **Pass** or **Fail** buttons.
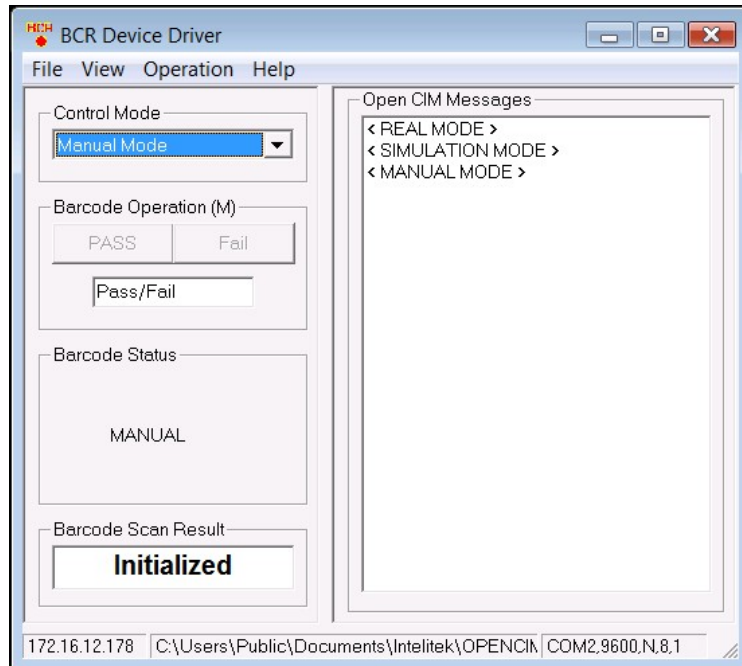
Figure 9-22: BCR Device Driver (Manual Mode)

To operate the BCR device driver in Manual Mode:

1. From the Control Mode drop down list, select **Manual Mode**.

2. Select **Pass** to indicate the part has passed the BCR test or **Fail** to indicate the part has failed its BCR test.

Standalone Mode

In this mode, the user is in direct control of the Bar Code Reader device without any interference from the CIM Manager. It is therefore the user's responsibility to verify that the Bar Code Reader Device is operational.
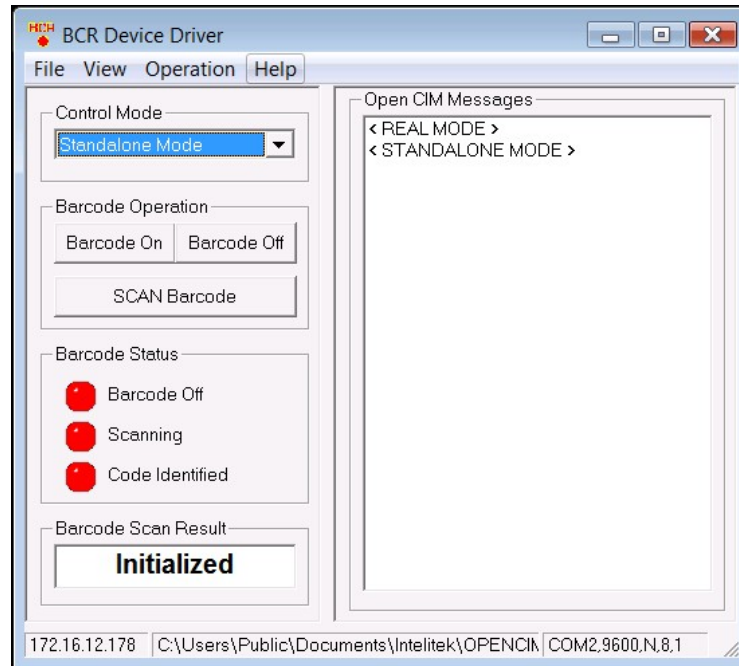
Figure 9-23: BCR Device Driver (Standalone Mode)

To operate the BCR device driver in Standalone Mode:

1. From the Control Mode drop down list, select **Standalone Mode**.
2. Select **Barcode On** to activate the barcode reader.
3. Select **SCAN Barcode** to scan the part in order to identify the Template Type or Template ID, as required. The result is displayed in the Barcode Scan Result area.
4. Press **Barcode Off** at any time to deactivate the barcode reader.

RFID Device Driver

The RFID device driver communicates with the RFID Reader and the CIM Manager.

The RFID device driver connects to the RFID Reader via an RS232 cable. For computers without a RS232 port, a USB to RS232 adapter may be used.

In order to use the RFID reader with OpenCIM, Template IDs must be assigned to RFID tags using the RFID Device Driver. For more information, refer to the subsection How to Assign a Template ID to an RFID Tag, on page 7-25.

User Interface

The RFID device driver interface is accessed from the CIM DDLoader window, as described in the following procedure.
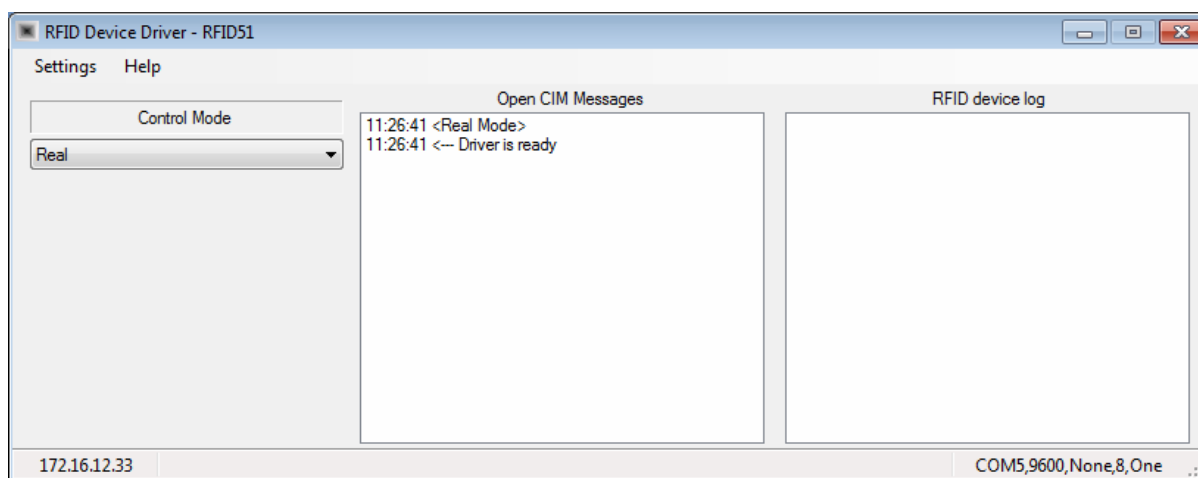
Figure 9-24: RFID Device Driver

To access the RFID device driver user interface:

1. Select the Loader icon of the workstation where the RFID device driver is installed. The CIM DDLoader window is displayed, as shown in Figure 9-2: DD Loader .

2. To operate the device driver in Real Mode check the **Load** column of the

   RFIDDRIVER. Then click the **Load Selected Drivers** icon to load the RFID device driver software. The various modes of operation are described in the following section.

Control Modes

Like all the device drivers, the RFID device driver can operate in Real, Simulation and Manual modes. It can also be operated in Standalone mode.

Real Mode   Normal operating mode. This is the default Driver mode. The device driver relays command messages to the RFID Reader from the CIM Manager. In turn, it broadcasts a QC result message to the CIM Manager.
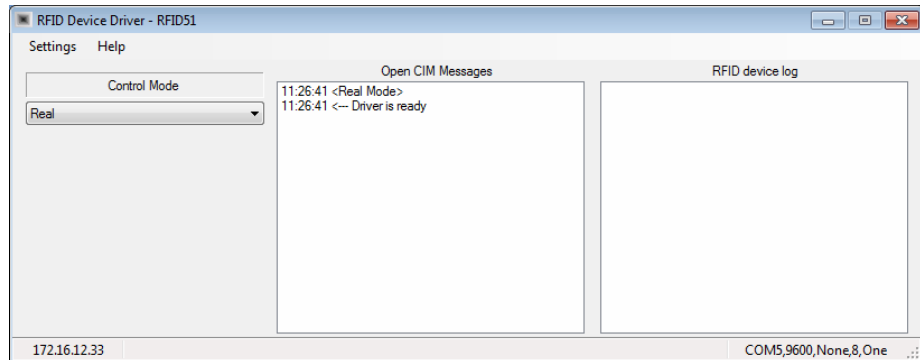
Figure 9-25: RFID Device Driver (Real Mode)

To operate the RFID device driver in Real Mode:

- From the Control Mode drop down list, select **Real**. A <Real Mode> message is displayed in the Open CIM Messages area.

Simulation Mode   The device driver emulates Real Mode by communicating with the CIM Manager. Communication with the RFID Reader is disabled in this mode.
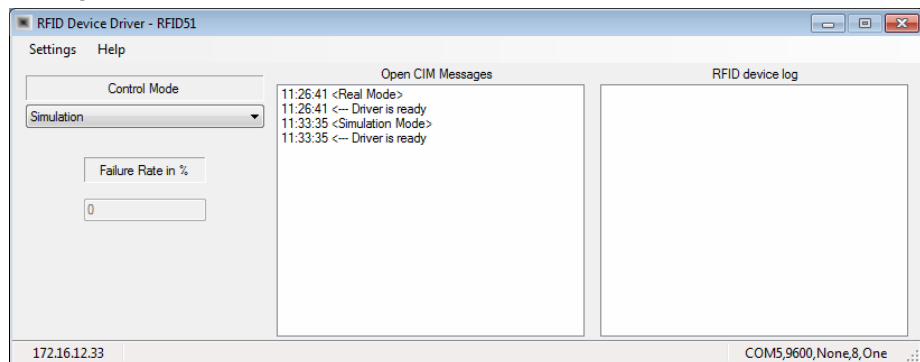
Figure 9-26: RFID Device Driver (Simulation Mode)

To operate the RFID device driver in Simulation Mode:

1. From the Control Mode drop down list, select **Simulation**.

   The RFID Operation area is updated with the failure rate percentage that is defined in the Machine Definition utility. For example, 50 indicates a 50 percent failure rate.

Manual
Mode

In this mode, the Device Driver receives commands from the CIM Manager, but only generates Pass/Fail status messages when you click on the **Pass** or **Fail** buttons.
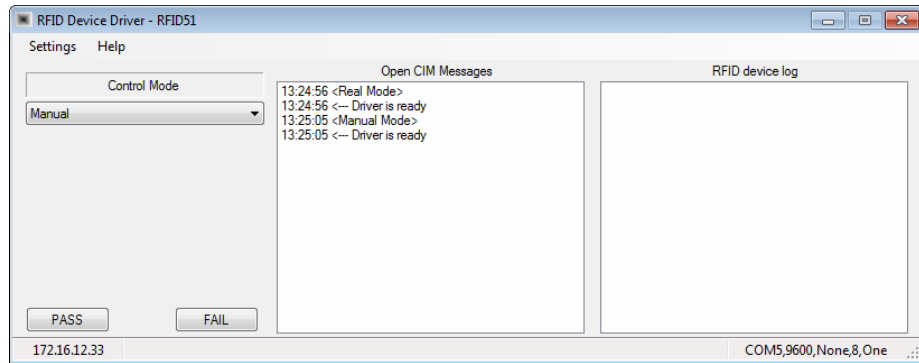
Figure 9-27: RFID Device Driver (Manual Mode)

To operate the RFID device driver in Manual Mode:

1. From the Control Mode drop down list, select **Manual**.

2. Select **Pass** to indicate the part has passed the RFID test or **Fail** to indicate the part has failed its RFID test.

Standalone
Mode

In this mode, the user is in direct control of the RFID Reader device without any interference from the CIM Manager. It is therefore the user's responsibility to verify that the RFID Reader Device is operational.
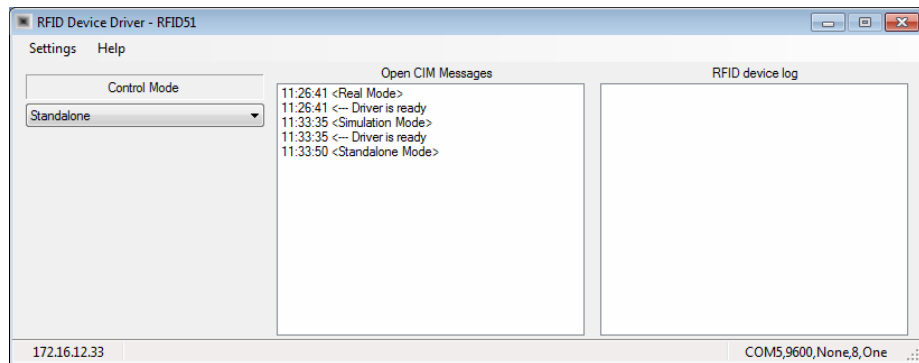
Figure 9-28: RFID Device Driver (Standalone Mode)

To operate the RFID device driver in Standalone Mode:

1. From the Control Mode drop down list, select **Standalone**.

2. Pass an RFID tag in front of the RFID reader. The reader will sound a beep and an entry for the RFID tag will be added to the RFID device log in the RFID Device Driver.

PLC Device Driver

The PLC device driver relays messages between the OpenCIM network and the programmable logic controller which controls the operation of the conveyor. These messages pertain to the movement of pallets on the conveyor.

This device driver runs on a PLC Manager PC which is usually designated as workstation 99. This PC is connected to the PLC via an RS232 link.

The PLC device driver:

- Receives a command message and tells a PLC control program to execute the corresponding function.
- Receives status messages from PLC control programs and broadcasts them on the OpenCIM network.
- Allows you to test and debug PLC control programs by sending commands from the Control Panel.
- Emulates pallets traveling on a conveyor in Simulation mode.

The PLC continually broadcasts the status of pallets on the conveyor as they move past stations. This flow of status messages enables you to see a real-time display of the conveyor in the Graphic Tracking module or on the Control Panel of the PLC device driver.

You can manually rearrange pallets on the conveyor while the CIM is running. However, if you manually change a pallet's payload, an error will eventually result since the payload in the CIM Manager's database will no longer correspond.

The PLC operates in a demanding real-time environment. It tracks the status and destination of every pallet on the conveyor without requiring constant communication with the CIM Manager. Each time a pallet arrives at a station, the PLC functions autonomously to stop the pallet; identify it; decide if this pallet is needed at this station, and if so, alert the CIM Manager. This sequence of events is continuous and is multiplied by the number of stations in the CIM.

In order to give the best possible response time, the PLC functions independently of the CIM Manager. When the CIM Manager needs a pallet at a station, it sends a command message to the PLC. The CIM Manager then waits for the PLC to inform it that the pallet has arrived. The PLC holds the pallet at the station until the CIM Manager sends a release command.

The CIM Manager does not track the continuous flow of pallets as they move around the conveyor. As with other devices, the CIM Manager specifies what it wants but does not get involved in the details of how to carry out the request.

PLC Messages

The PLC device driver receives the following command messages from the CIM Manager and relays them to the PLC. These commands correspond to the buttons on the PLC Control Panel.

Commands to the PLC Device Driver	Description
GetFree	Orders the PLC to stop the next empty pallet at the specified station. Used when a part (or empty template) needs to be picked up at this station (including the ASRS station).
Release	The CIM Manager allows a needed pallet continue on the conveyor if the station is busy when the pallet arrives. Releasing the pallet prevents a traffic jam on the conveyor. This can occur if the robot that loads/unloads pallets is busy or if a pallet cannot be unloaded because the buffers are full. The pallet's destination remains unchanged. **Pallet Carrying Template -** If the pallet's destination = this station, it will be stopped the next time it comes around to this station. **Empty Pallet -** If the pallet's destination = 99, the next empty pallet to arrive at this station will be stopped.
Deliver	Orders the PLC to stop the specified pallet at the specified station. The CIM Manager issues this command in order to assign a destination to a pallet at the time it is loaded with a template (i.e. every pallet carrying a template should have a destination).
Free	Release a pallet that was unloaded at this station. Flag it as available (i.e. destination = 99). This command is similar to Release except that the pallet's destination is changed to 99 to indicate that the pallet is empty and available.

The following status messages from the PLC are forwarded to the CIM Manager by the PLC device driver:

 A pallet carrying a template for this station has arrived.

 An empty pallet has stopped to pick up a template.

If the Graphic Tracking module is running, a Pass message can be generated each time a pallet passes through a station where it was not needed. These messages keep the real-time conveyor display updated. However, if the frequency of these messages slows down the system, you can improve performance by disabling them.

The following sample scenario demonstrates the role of the PLC device driver in relaying command and status messages between the CIM Manager and the PLC. This scenario assumes that a part is being picked up from the ASRS station 1 and delivered to production station 2.

Message **(Command messages in bold)** *(Status messages in italics)*	Description
GetFree **Stop an empty pallet at station 1**	The CIM Manager sends a command to the PLC to stop the next empty pallet that arrives at the specified station.
Empty pallet has arrived at station 1	The PLC responds with a status message when an empty pallet has arrived.
Deliver **Stop loaded pallet at station 2**	After the ASRS robot loads a part template on the pallet, the CIM Manager sends a command to the PLC specifying that the PLC should stop this pallet at station 2. The PLC releases the pallet from the ASRS station 1.
Loaded pallet has arrived at station 2	The PLC sends a status message to the CIM Manager when the pallet arrives at the station.
Free **Allow empty pallet to leave station 2**	After a robot removes the template from the pallet, the CIM Manager sends a command to the PLC to release this empty pallet. This empty pallet is now tagged as available (i.e. destination = 99). It circulates on the conveyor until the CIM Manager sends a request to the PLC for an empty pallet (return to step 1).

PLC Control Panel

The PLC Control Panel is a feature of the PLC device driver. It allows you to perform the following functions:

Monitor the location and destination of every pallet on the conveyor.

Test the PLC and conveyor by manually issuing command messages to the PLC.

Simulate the movement of pallets on the conveyor.

Determine the control mode the device driver is running in.
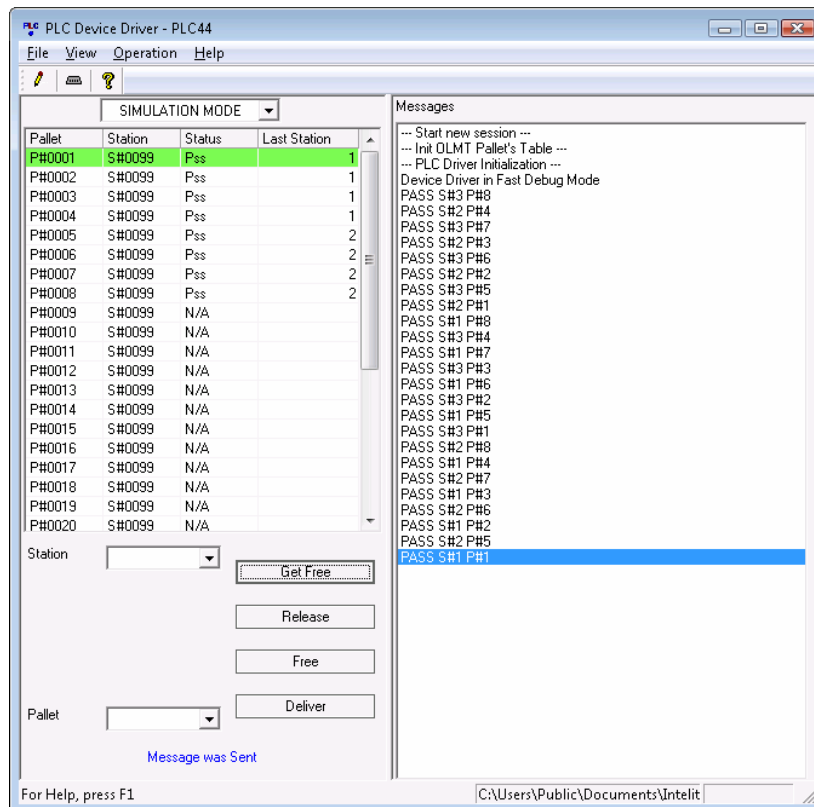
Figure 9-29: PLC Control Panel

Pallet Status Display

The Pallet Status display shows the status, location, and destination station for each pallet on the conveyor. This display is updated each time the PLC stops a pallet and identifies it at a station.

A code of 99 indicates an empty pallet that is available for use, i.e. it has no destination. The status of a pallet (listed in the Status column) can be one of the following:

Pallet Status	Description
Run	This is the default condition for all pallets when the CIM starts up. This status remains in effect until a pallet passes its first station and an update message from the PLC is received. If the Run status for a pallet never changes, this indicates that the pallet is not currently present on the conveyor.
	The Run status can also be assigned as a result of a Free command. The pallet retains this status only until it reaches the next station at which time it changes to either Pass or Arrive.
	A pallet with a Run status has a destination of 99, i.e. no destination has been assigned to it.
Arr (Arrive)	Assigned to an empty pallet that is being held at a station waiting to be loaded. The pallet is stopped at the station as a result of a GetFree command.
Stp (Stop)	Indicates that a loaded pallet has arrived at its destination.
Rls (Release)	A pallet required by this station has arrived, but the station is too busy to deal with the pallet. Usually this occurs as a result of a busy robot or when all the station's buffers are full.
	Rather than hold up traffic on the conveyor, the pallet continues on the conveyor. If the pallet is carrying a template for this station, it will be stopped the next time it comes around. If the pallet is empty, the next empty pallet will be stopped in its stead.
	The Release status changes to either Pass or Arrive when the pallet reaches the next station.
Pss (Pass)	Indicates that the pallet just passed through a station that was not its destination.

Controlling Pallets from the Control Panel

The buttons at the bottom of the Control Panel allow you to send commands to control the movement of pallets on the conveyor. The buttons described below correspond to the commands that the CIM Manager sends to the PLC device driver.

Before using these buttons, you must first select a station. In order to select a station for the operations shown below, use the list box of station numbers found along the right-hand edge of the Control Panel.

Option	Description
Deliver	Stops the specified pallet when it arrives at the specified station. Select the desired pallet by clicking on the line with the correct pallet ID. Then select a station before using this button.
GetFree	Stops the next empty pallet that arrives at the specified station. Select the desired station before using this button.
Free	Allows a pallet that was unloaded at this station to continue on the conveyor and flags it as available (i.e. assigns the pallet's destination station = 99). Select the desired station before using this button.
Release	Lowers the piston at the specified station to allow a pallet that is just passing through this station to continue moving along the conveyor. You can also use this button to allow a needed pallet to pass if the station is currently busy. Select the desired station before using this button.

It is NOT recommended to manually select *GetFree* or *Deliver* while the CIM is running in Real Mode. If you do, the CIM Manager will receive an unexpected status message indicating the arrival of a pallet that did not actually arrive. The CIM Manager will attempt to recover from this situation by issuing a Free command for this pallet.

Pallet Command Box

You can use the PLC Control Panel as a limited terminal to send commands to a controller. Commands that you type in the Pallet Command Box are sent out via the PC's serial port when you press [Enter]. Responses from the PLC are displayed in the Status window of the device driver.

This capability, which should only be used by PLC programmers, is useful for testing and debugging PLC control programs.

Simulating a Conveyor

When you want to run a CIM simulation, the PLC device driver can simulate the operation of pallets moving along the conveyor (when running in Simulation mode or Manual mode). You can set the following parameters in the appropriate INI file in order to customize the simulation of the conveyor:

- The number of pallets traveling on the conveyor (`SimulationStations`)
- The number and order of stations around the conveyor (`SimulationPallets`)
- The distance between stations (`SimulationPosPerStation`)
- The direction in which the conveyor moves (`SimulationDirection`)

Hydraulic Device Driver

The hydraulic device driver runs on a PC that controls the hydraulic station via Advantech's I/O PCI card. This section describes the installation, testing and operation procedures of the hydraulic device driver and includes the following

- Installing the Hydraulic Device Driver
- Testing the Hydraulic Device Driver in CIM (Standalone Mode)
- Operating the Hydraulic Device Driver in OpenCIM (Real Mode)
- (Optional) Operating the Hydraulic Device Driver in OpenCIM (Offline)

For additional details refer to the following references provided:

- Advantech DLL Drivers User's Manual and Programmers Reference.
- HYD-2800 Hydraulic Station Maintenance Manual.

Installing the Hydraulic Device Driver

After you have setup the hardware, as described in the HYD-2800 Hydraulic Station Maintenance Manual, the next step is to install Advantech's DLL driver software. This includes inserting the PCI-1711 card, installing the device driver and defining the settings. For additional details refer to the Advantech DLL Drivers User's Manual and Programmers Reference that was provided with the software.

To install the hydraulic device driver software:

1. From the CD-ROM drive install Advantech's Hydraulic device driver software (that supports the I/O card installed). From the initial window select the DLL Drivers V1.4 option and proceed to the next step of the installation where you select the required windows platform, as follows:

2. Select Windows 2000 as your platform.

    In Windows Vista/7: When prompted you must enter Administrator credentials or if you are already administrator click allow.

3. Follow the remaining instructions of the installation wizard to complete the installation process.

4. Power down the computer.

5. Remove the cover of the computer and insert the **PCI-1711** I/O card into one of the available PCI slots.

6. Replace the cover of the computer.

7. Connect the cable from the rear panel of the hydraulic robot to the computer and power up the computer.

8. From the windows **Start** menu, select **Advantech Driver v1.4** | **Device Installation**. The Advantech IO Device Installation window is displayed.
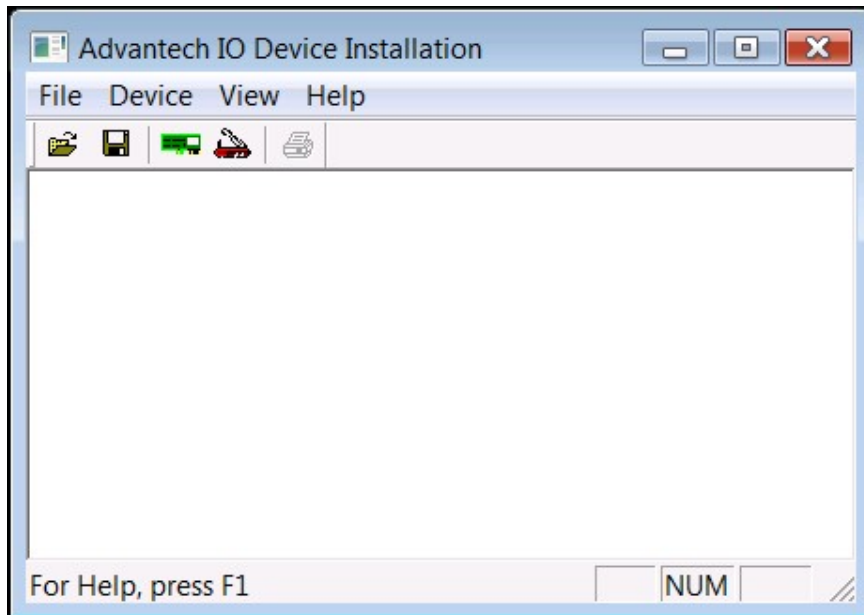
Figure 9-30: Advantech IO Device Installation

9. From the **Device** menu, select **Setup**. The Advantech IO Device Installation window is displayed.
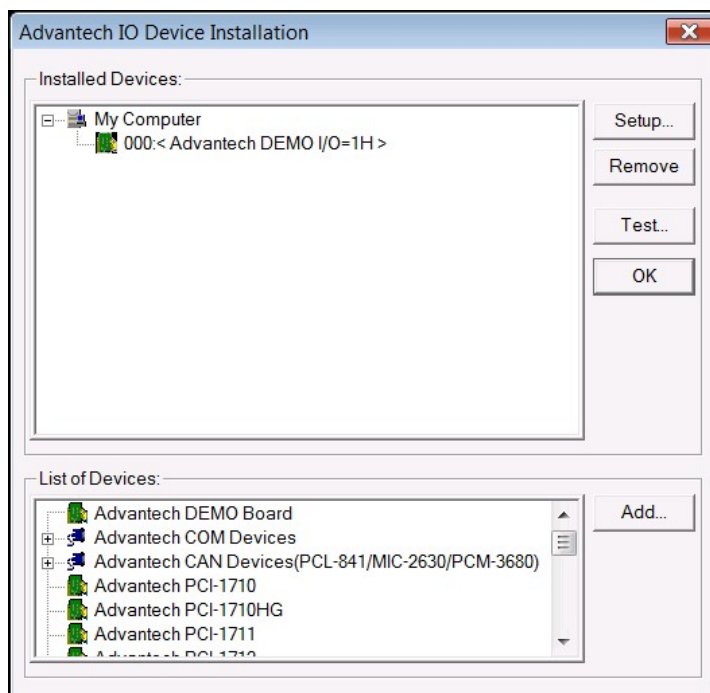
Figure 9-31: Advantech IO Device Installation

10. From the List of Devices area, select **Advantech PCI-1711** and then click **Add**. The **Advantech PCI-1711** device is displayed in the Installed Devices Area.

11. From the Installed Devices Area select **Advantech PCI-1711** and then click **Setup**. The PCI-1711 Device Setting window is displayed (in Windows XP).
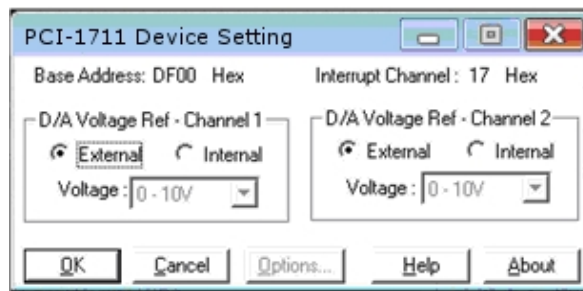
Figure 9-32: PCI-1711 Device Setting

12. Select both the **External** options from both the D/A Voltage Ref – Channel 1 and D/A Voltage Ref – Channel 2 areas and then click **OK**. A confirmation message is displayed.

13. Click **OK** to restart your computer in order for these changes to take effect.

Testing the Hydraulic Device Driver in CIM (Standalone Mode)

After installing the hydraulic device driver, the next step is to test the device driver to verify that the card is working properly in the CIM environment in standalone mode. This includes loading the hydraulic device driver from the CIM DDLoader window, and then activating the Load.pgm, Press.pgm and Unload.pgm applications.

To test the hydraulic device driver in CIM (standalone mode):

1. From the windows desktop double-click the Loader of **WS1** icon (where the hydraulic station is installed). The CIM DDLoader window is displayed.
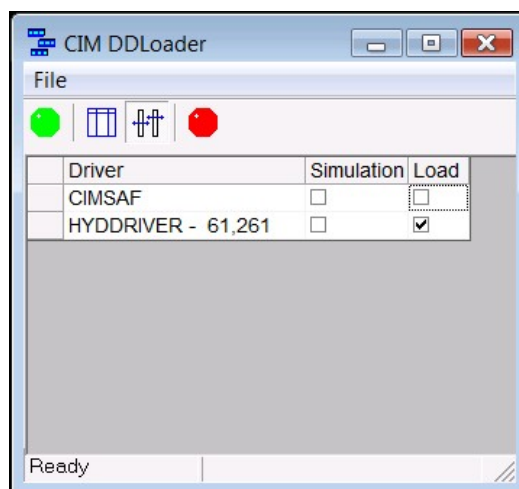
Figure 9-33: CIM DDLoader (WS1)

2. Select the **Load** column of the **HYDDRIVER** then click the

   Load Selected Drivers 🟢 icon to load the Hydraulic Robot Device Driver software.

3. From the windows system tray select **Hydraulic Robot Device Driver**. The Hydraulic Robot Device Driver window is displayed.
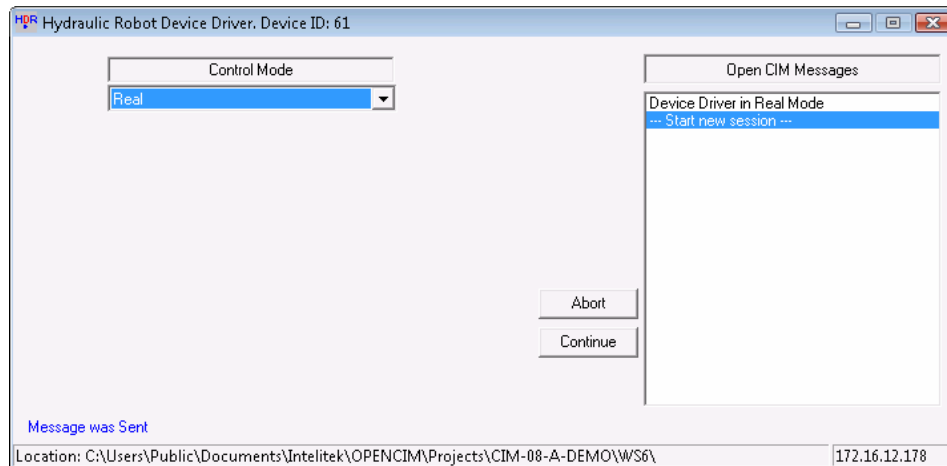
Figure 9-34: Hydraulic Robot Device Driver

4. From the Control Mode drop down list,.select **Standalone**. The window is expanded as follows.
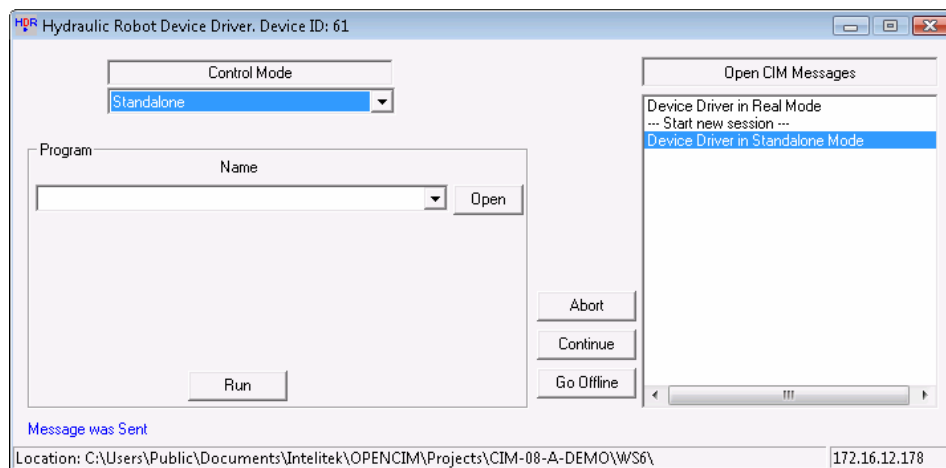
Figure 9-35: Hydraulic Robot Device Driver (Standalone)

5. Click **Open**. A standard navigation window is displayed.

6. Navigate and select the **Load.pgm** program and then click **Run**. The hydraulic robot to lifts template off the conveyer and places it on the jig.

7. Repeat Step 5, then navigate and select the **Press.pgm** program and click **Run**. The hydraulic robot compresses the template that is placed on the jig.

8. Repeat Step 5, then navigate and select the **Unload.pgm** program and click **Run**. The hydraulic robot removes the compressed template from the jig and places it back on the conveyer.

This indicates the hydraulic device driver is working properly in the CIM environment.

Operating the Hydraulic Device Driver in OpenCIM (Real Mode)

After installing and testing the hydraulic device driver, you can operate the hydraulic device driver in the OpenCIM environment in Real Mode. This includes, loading the Hydraulic Robot Device Driver from the CIM DDLoader window, and running the CIM Manager. The CIM Manager automatically activates the Load.pgm, Press.pgm and Unload.pgm applications. This indicates that a connection exists between the hydraulic device driver and the CIM Manager.

To operate the hydraulic device driver in OpenCIM (realmode):

1. Select the **Loader** icon of the workstation where the hydraulic device driver is installed. The CIM DDLoader window is displayed.
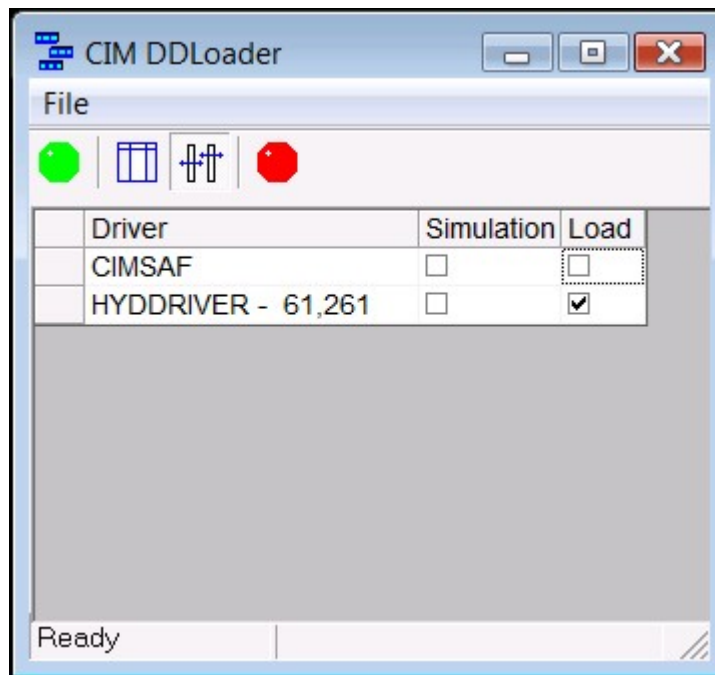
Figure 9-36: CIM DDLoader

2. Ensure that the **Simulation** column of the **HYDDRIVER** is deselected and then click

   the **Load Selected Drivers** ⬤ icon to load the Hydraulic Robot Device Driver software. The pump is automatically turned on. This indicates the station is working properly.

3. To review the CIM Messages, select **Hydraulic Robot Device Driver** from the system tray. The Hydraulic Robot Device Driver window is displayed, as shown in the following example:
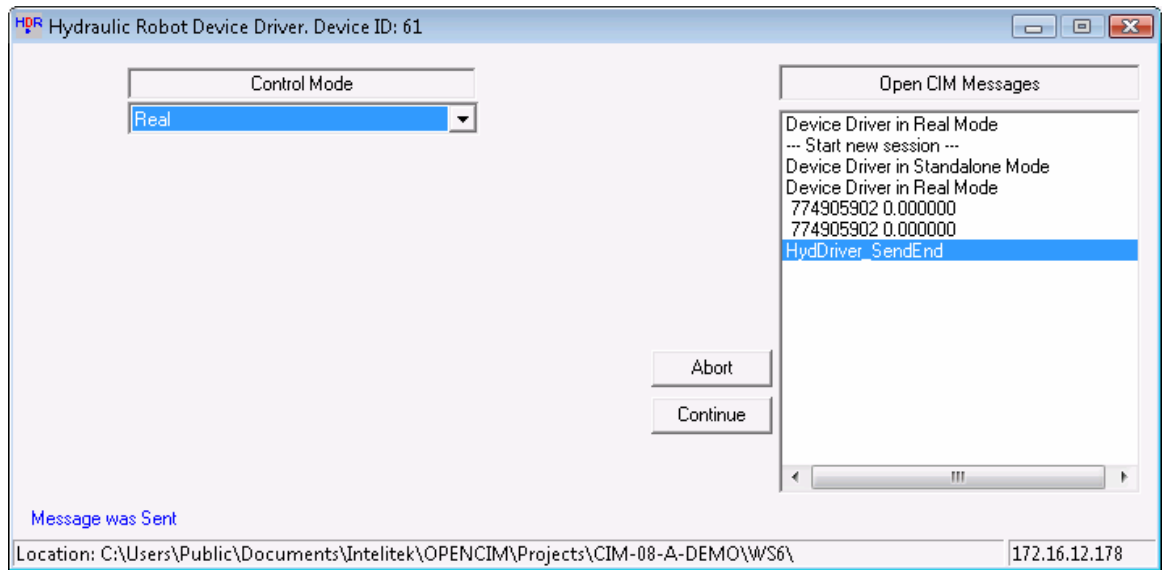
Figure 9-37: Hydraulic Robot Device Driver

☞Note   *To stop the hydraulic robot program execution, do one of the following:*

- *Press **F9** hotkey on your keyboard*
- *Click the **Abort** button from the Hydraulic Device Driver window.*
- *Click the **Emergency Stop** button on the front panel of the hydraulic station.*

*You can then click **Continue** from Hydraulic Device Driver window to proceed.(If the **Emergency Stop** button was pressed, you must first release it before clicking **Continue** to proceed.)*

(Optional) Operating the Hydraulic Device Driver in OpenCIM (Offline)

The Hydraulic Robot Device Driver window enables you to operate the hydraulic device driver offline and create your own programs according to your needs.

For additional details refer to the Advantech DLL Drivers User's Manual and Programmers Reference that was provided with the software.

To operate the hydraulic device driver in OpenCIM (offline):

1. Select the **Loader** icon of the workstation where the hydraulic device driver is installed. The CIM DDLoader window is displayed, as shown in Figure 9-36: CIM DDLoader.

2. Ensure that the **Simulation** column of the **HYDDRIVER** is deselected and then click the **Load Selected Drivers** icon to load the hydraulic device driver software. The pump is automatically turned on. This indicates the station is working properly.

3. From the windows system tray select **Hydraulic Robot Device Driver**. The Hydraulic Robot Device Driver window is displayed, as shown in Figure 9-37: Hydraulic Robot Device Driver.

4. From the Control Mode drop down list, select **Standalone**. The window is expanded as shown in Figure 9-35: Hydraulic Robot Device Driver (Standalone).

5. Select **Go Offline**. The Hyd-2800 window is displayed.
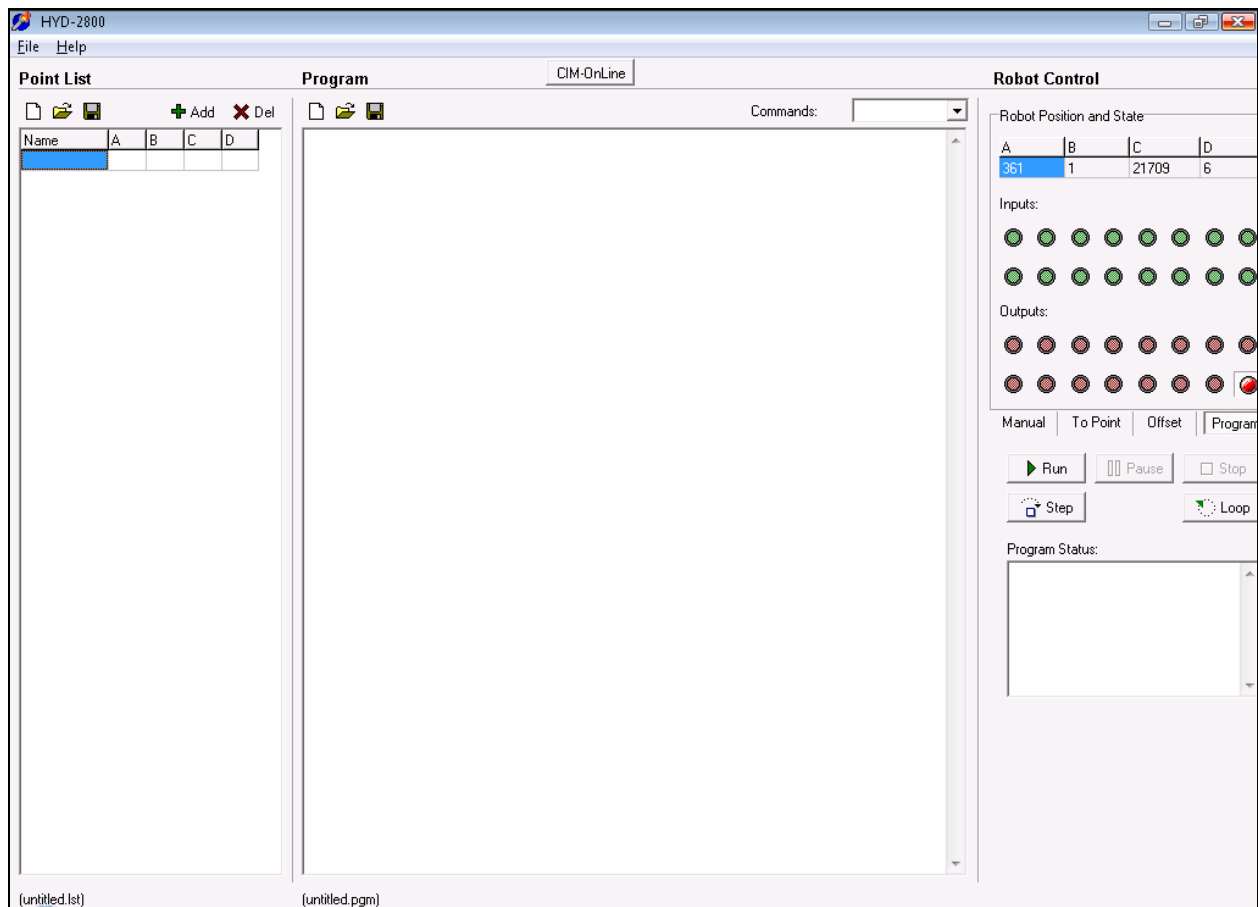
Figure 9-38: HYD-2800 (Offline)

6. You can now create your own programs according to your requirements.
   Click **CIM-Online** to return to the Hydraulic Robot Device Driver window at any time

7. After creating your program you can define it in the OpenCIM Manager application, as follows:

Launch the OpenCIM Manager application.

Select Utility Programs | Machine Definition. The CIM Machine Definition window is displayed, see "Machine and Process Definitions" in Chapter 5, for further details.

Define the process to be performed by the hydraulic robot and enter the program name in the Program column.

Pneumatic Device Driver

The pneumatic device driver runs on a PC that controls the pneumatic station via the Vuniq software. This section describes the installation, configuration, operation and troubleshooting procedures of the pneumatic device driver and includes the following:

- Installing the Vuniq Software
- Configuring the Vuniq Application Settings
- Operating the Pneumatic Device Driver in OpenCIM - Real Mode
- Pneumatic Station Troubleshooting

For additional details refer to the following references provided:

- PN-2800 Pnuematics Application – Operators Manual, Instructor's Manual
- PN-2800 Pnuematics System – Student Manual

Installing the Vuniq Software

After setting up the hardware station, as described in the Pneumatics Application Operator's Manual, the next step is to install the Vuniq software that supports the pneumatic station hardware. This includes inserting the licence plug, copying the application files to the default directory and more.

To install the Vuniq software:

1. From the CD-ROM drive install the Vuniq for Windows application 4.5 and follow the instructions provided in the installation wizard. When the installation is complete, the

   Vuniq Vuniq icon is displayed on your desktop.

2. Install the license plug in the parallel port on the PC.

3. Copy the **\\Projects\LIB\Vuniq40** directory from manager computer on local network to the [default system drive (for example C:)]**vuniq40** directory on the workstation computer. This directory contains the files that are required to configure the Vuniq Application Manager to work with the pneumatic station.

&✐Note   *These directories containing the files required to configure the Vuniq Application Manager are the default directories and can be changed as required.*

4. From the **Start** menu select **Vuniq** or double-click the **Vuniq** Vuniq icon from your windows desktop. In Windows Vista/7 you should right-click the icon and select "Run as Administrator". The Vuniq for Windows main window is displayed and the **Vuniq**

   icon appears on your system tray.

5. From the Windows system tray, right click the **Vuniq** icon. A popup menu is displayed.

---

6. Select **Application Manager**. The Vuniq Application Manager window is displayed.
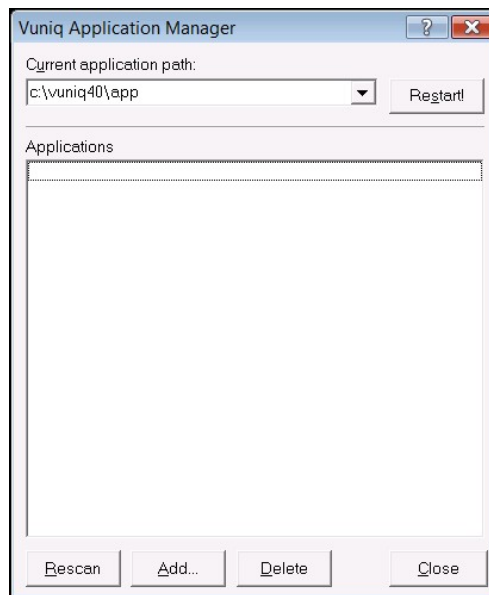
Figure 9-39: Vuniq Application Manager (Pneumatics)

7. In the Current application path field, enter the **C:\Vuniq40\pn2000** path that was defined as the default directory during the installation process.

8. Click **Restart**. Pneumatics Station window (shown in Figure 9-41: Pneumatics Station) is displayed in the background. Click **Close** to close the Vuniq Application Manager window.

Configuring the Vuniq Application Settings

After installing the Vuniq software, the next step is to define the Vuniq application settings. This includes defining the port number and verifying that communication exists between the pneumatics station and the computer.

To define the Vuniq Application Settings:

1. From the Windows system tray, right-click the **Vuniq** icon. A popup menu is displayed.

2. Select **Application Settings**. The Vuniq Applications Settings window is displayed.

3. From the Category area, select **Communications**, the **General** tab appears in the Communications area.
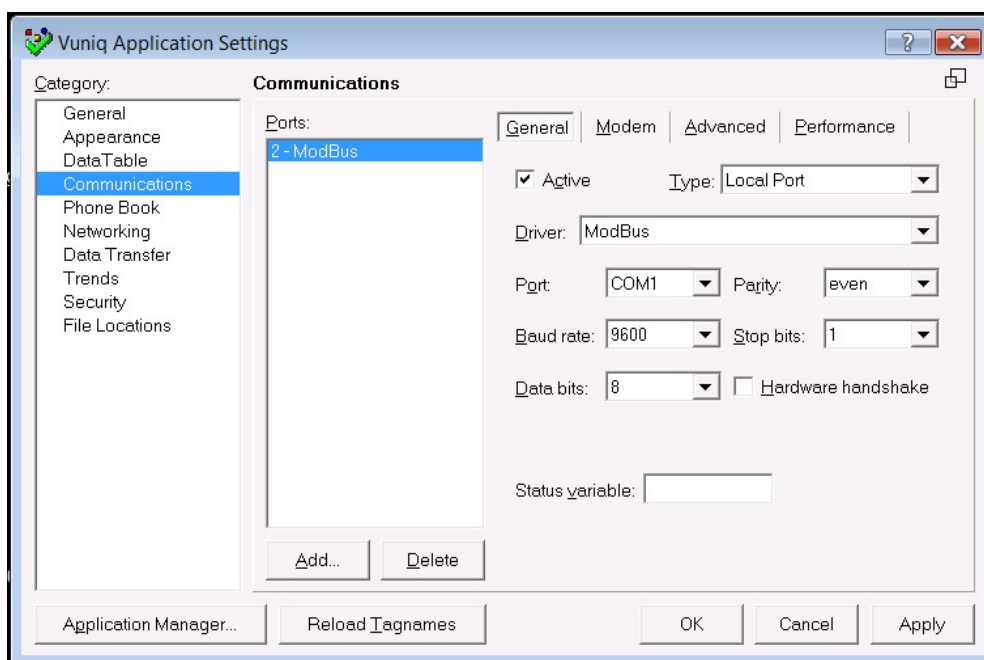
Figure 9-40: Vuniq Applications Settings

4. From the **Port** drop down list select the required port number. (The default port number is COM1) and click **OK**.

5. Right-click the **Vuniq** icon. A popup menu is displayed.

6. Select **Operator Screens**. The Pneumatics Station window is displayed.
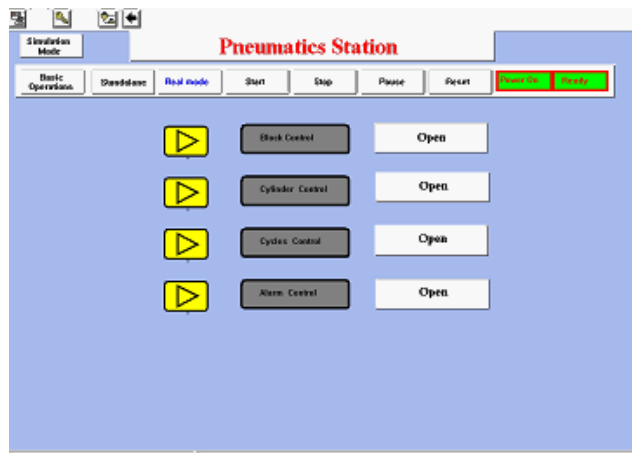
Figure 9-41: Pneumatics Station

7. Verify that the Power On button is green to indicate that the communication between the pneumatics station and the computer is working properly.

✍Note   *If the* Power On *button is not displayed in green press* Start *or press the PLC ON button on the front panel of the pneumatics station.*

8. Right-click the **Vuniq** icon. From the displayed popup window select **Quit** and then click **Yes** at the displayed confirmation message to exit the Vuniq application.

Operating the Pneumatic Device Driver in OpenCIM - Real Mode

After installing the software and defining the settings, you can operate the pneumatic device driver in the OpenCIM environment in real mode. This includes, loading the Pneumatic Robot Device Driver from the CIM DDLoader window, and running the CIM Manager.

To operate the pneumatic device driver in OpenCIM (realmode):

1. Select the **Loader** icon where the pneumatic station is installed. The CIM DDLoader window is displayed.
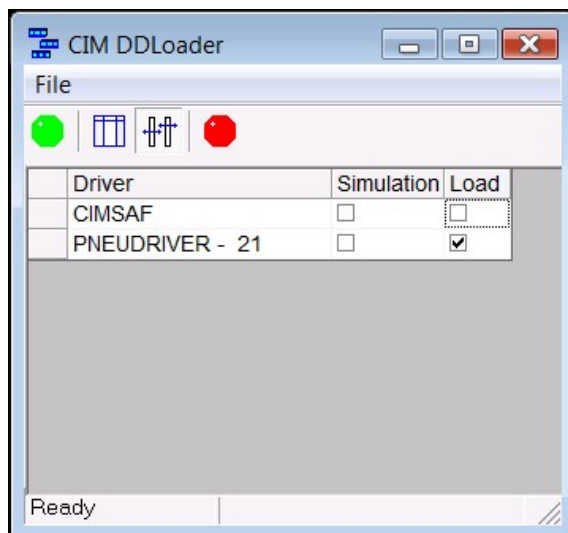
Figure 9-42: CIM DDLoader (PNEUDRIVER)

2. Ensure that the **Simulation** column of the **PNEUDRIVER** is deselected and then click the **Load Selected Drivers** ⬤ icon to load the Pneumatic Station Device Driver software. This activates the pneumatic device driver and displays the Pneumatics Station window. Verify that the [Power On] and [Ready] buttons are displayed in green, to indicate that the station is working properly in the OpenCIM environment.

3. To review the CIM Messages, select **Pneumatic Station Device Driver** from the system tray. The Pneumatic Station Device Driver window is displayed, as shown in the following example:
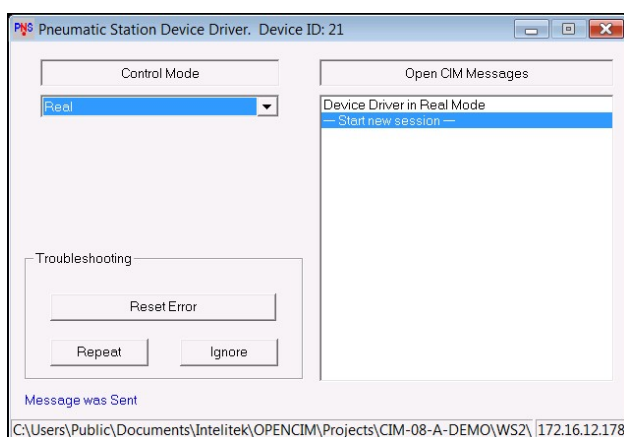
Figure 9-43: Pneumatic Station Device Driver

&⌒Note   *To stop the pneumatic robot program execution, press the Emergency Stop button on the front panel of the pneumatic station.*

Pneumatic Station Troubleshooting

Whenever a problem is detected with the operation of the pneumatic station device driver in the OpenCIM environment, the pneumatic station Error led is automatically turned on and the Pneumatic Station Device Driver window appears (as shown in Figure 9-44: Pneumatic Station Device Driver – Error Message). This section describes how to correct the example error messages that may appear.

Station Not Ready - Error Message

This message can appear due to the following reasons:

The Emergency button is pressed on the pneumatic station

Solution: Release the Emergency button and then press the [Start] and [Real mode] buttons.

The pneumatic station is working in Standalone Mode or Basic Operations mode.

Solution: Deselect the [Standalone] button or [Basic Operations] button and then click the [Real mode] button in the Pneumatics Station window.

The PLC ON button on the pneumatic station is turned off.

Solution: Press The PLC ON button or click the [Start] button and then click the [Real mode] button in the Pneumatics Station window.

Block is Missing – Error Message

This message appears when a block is missing. To correct the error follow the procedure described below.

An example of an error message is displayed below.
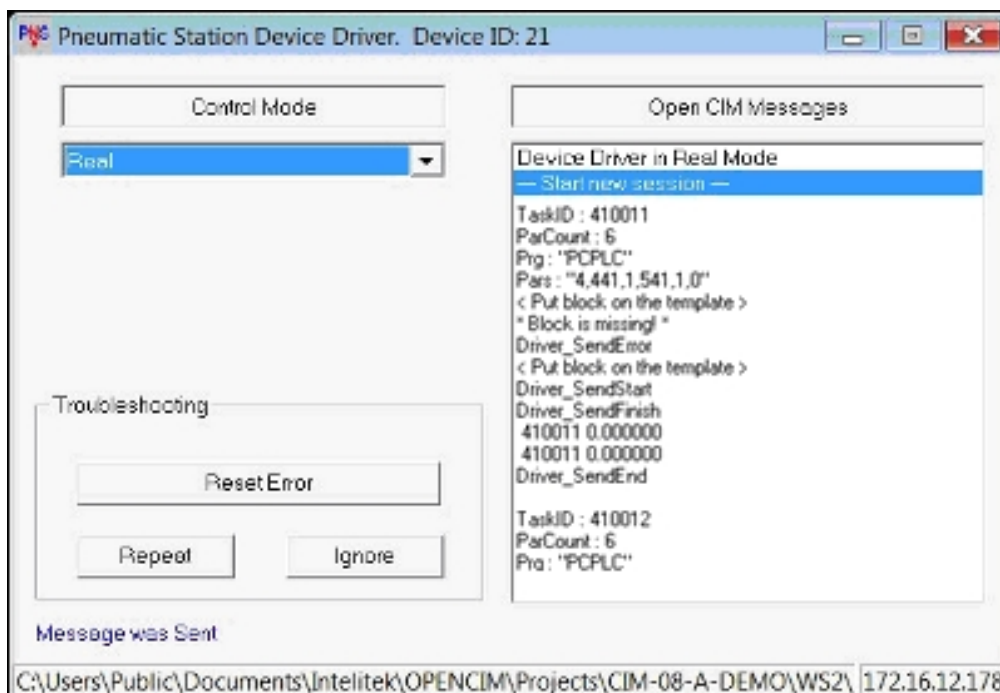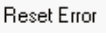
Figure 9-44: Pneumatic Station Device Driver – Error Message

To reset the error and continue with the pneumatic process:

1. From the Pneumatics Station Device Driver window select the [Reset Error] button to reset the error status on the pneumatic station. The pneumatic station's Error led is turned off.
2. Manually Place the block on the template or in the feeder as required.

3. Select the required option, as follows:

Ignore	Select this option after manually placing the block on the template.
Repeat	Select this option after manually placing the block in the feeder to enable the pneumatic station to proceed with the process.

The pneumatic station will proceed with the process.

Process Device Driver

The process device driver runs on a PC that controls the process station via the Vuniq software. The station is controlled via PLC, using an RS232 interface. This section describes the installation, configuration, operation and troubleshooting procedures of the process device driver and includes the following:

- Installing the Vuniq Software
- Configuring the Vuniq Application Settings
- Operating the Process Device Driver in OpenCIM - Real Mode
- Viewing the Processing Station Information
- Process Station Troubleshooting

For additional details refer to the following references provided:

- PS-2800 Process Control – Maintenance Manual
- PS-2800 Coating System – Student Manual

The process station consists of seven baths. Each bath has a specific function. The station performs the following tasks in the OpenCIM environment:

- Lifts the template and raw material off the conveyer and places it on the buffer.
- Lifts the raw material off the template and performs the process that is displayed in the Vuniq software. At the end of this process the processed part is placed back on the template.
- Lifts the template and processed part off the buffer and places it back on the conveyer.

Installing the Vuniq Software

After setting up the hardware station, as described in the PS-2800 Process Control – Maintenance Manual, the next step is to install the Vuniq software that supports the process station hardware. This includes inserting the licence plug, copying the application files to the default directory and more.

To install the Vuniq software:

1. From the CD-ROM drive install the Vuniq for Windows 4.5 application and follow the instructions provided in the installation wizard. When the installation is complete, the

   Vuniq Vuniq icon is displayed on your desktop.

2. Install the license plug in the parallel port on the PC.

3. Copy the **\\Projects\LIB\Vuniq40\proc** directory from manager computer on local network to the [default system drive (for example C:)]**vuniq40** directory on the workstation computer. This directory contains the files that are required to configure the Vuniq Application Manager to work with the process station.

&⁓Note   *These directories containing the files required to configure the Vuniq Application*
         Manager are the default directories and can be changed as required.

4. From the **Start** menu select **Vuniq** or double-click the **Vuniq** Vuniq icon from your
   windows desktop. In Windows Vista/7, right-click the icon and select "Run as
   Administrator". The Vuniq for Windows main window is displayed and the **Vuniq**

    icon appears on your system tray.

5. From the Windows system tray, right click the **Vuniq** icon. A popup menu is
   displayed.

6. Select **Application Manager**. The Vuniq Application Manager window is displayed.
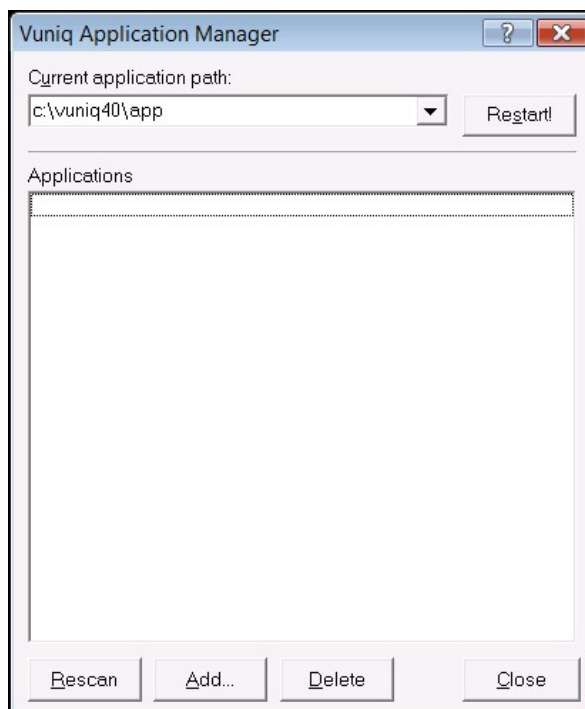
Figure 9-45: Vuniq Application Manager (Process)

7. In the Current application path field, enter the [default system drive (for example
   C:)]**Vuniq40\proc** path that was defined as the default directory during the installation
   process.

8. Click **Restart**. The Main Process Station window (shown in Figure 9-47: Main Process
   Station) is displayed in the background. Click **Close** to close the Vuniq Application
   Manager window.

Configuring the Vuniq Application Settings

After installing the Vunig software, the next step is to define the Vuniq application settings. This includes defining the port number and verifying that communication exists between the pneumatics station and the computer.

To define the Vuniq Application Settings:

1. From the Windows system tray, right-click the **Vuniq** icon. A popup menu is displayed.

2. Select **Applications Settings**. The Vuniq Application Settings window is displayed.

3. From the Category area, select **Communications**, the **General** tab appears in the Communications area.
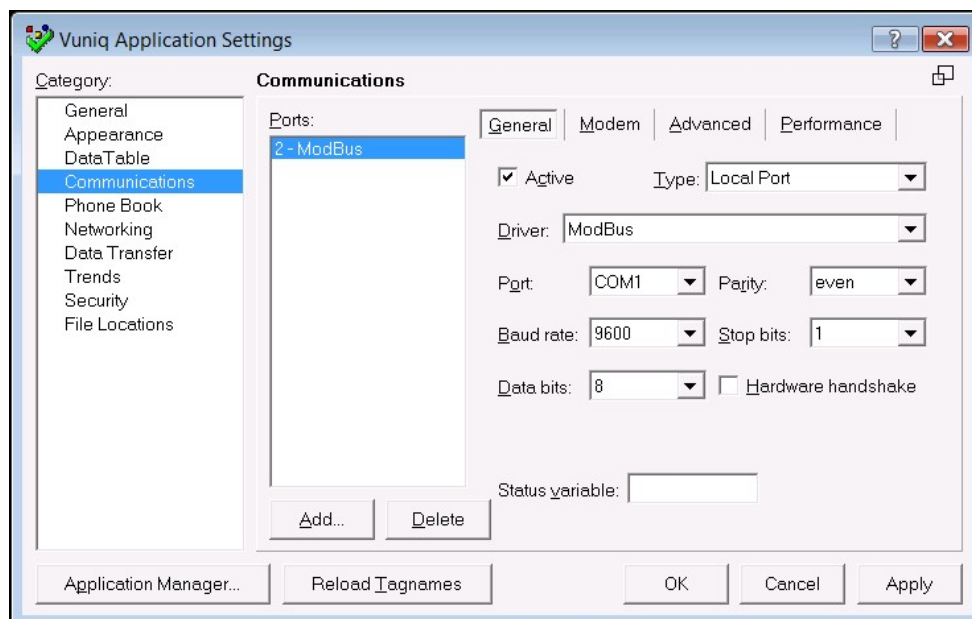
Figure 9-46: Vuniq Applications Settings (Process)

4. From the **Port** drop down list select the required port number. (The default port number is COM1) and click **OK**.

5. Right-click the **Vuniq** icon. A popup menu is displayed.

6. Select **Operator Screens**. The Main Process Station window is displayed.

Figure 9-47: Main Process Station

7. Verify that the ▭Power On button is green to indicate that the communication between the process station and the computer is working properly.

☞Note   *If the* ▭Power On *button is not displayed in green press* ▭Start *or press the PLC ON button on the front panel of the process station.*

8. Right-click the **Vuniq** ▭ icon. From the displayed popup window select **Quit** and then click **Yes** at the displayed confirmation message to exit the Vuniq application.

Operating the Process Device Driver in OpenCIM - Real Mode

After installing the software and defining the settings, you can operate the process device driver in the OpenCIM environment in Real Mode. This includes, loading the Process Station Device Driver from the CIM DDLoader window, and running the CIM Manager.

To operate the process device driver in OpenCIM (realmode):

1. Select the **Loader** icon where the process station is installed. The CIM DDLoader window is displayed.
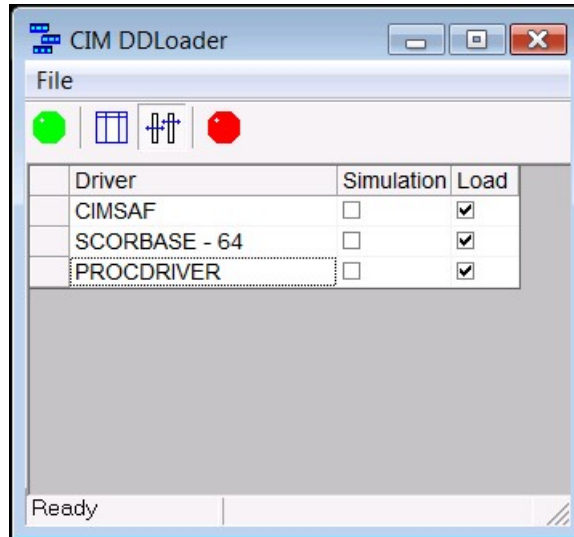
Figure 9-48: CIM DDLoader (process)

2. Ensure that the **Simulation** column of the **PROCDRIVER** is deselected and then click the **Load Selected Drivers** ⬤ icon to load the Process Station Device Driver software. This activates the process device driver and displays the Main Process Station window.

Verify that the [Power On] and [Ready] buttons are displayed in green, to indicate that the station is working properly in the OpenCIM environment.

3. To review the CIM Messages, select **Process Station Device Driver**. The Process Station Device Driver window is displayed, as shown in the following example:
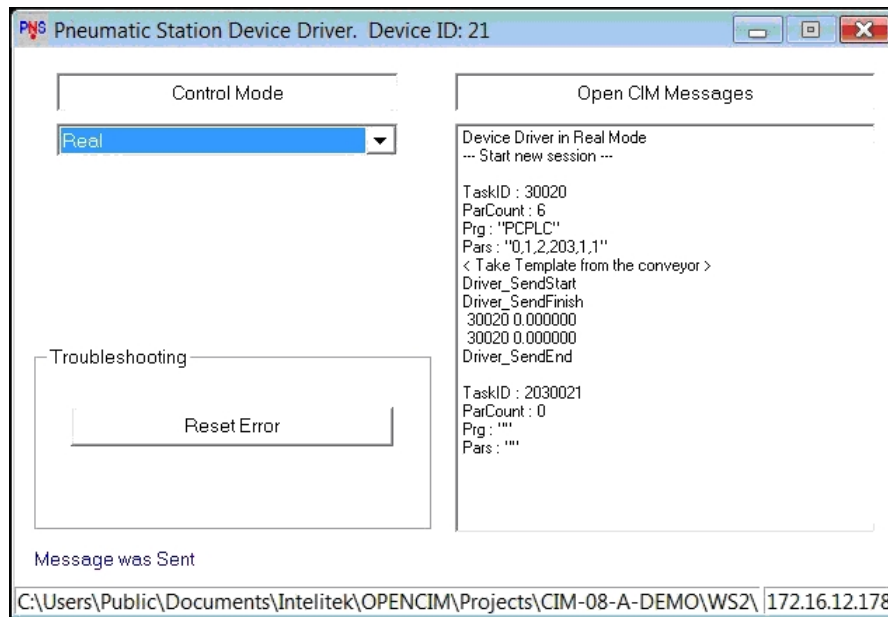
Figure 9-49: Process Station Device Driver

&⌒Note   *To stop the process robot program execution, press the Emergency Stop button on the front panel of the process station.*

Viewing the Processing Station Information

You can view the processing stages at any time throughout the process from the Process Station - Program Screen window. Information includes, cycle, time set, bath number and more.

To view the processing stages:

1. Right-click the **Vuniq** icon. A popup menu is displayed.

2. Select **Operator Screens**. The Process Station - Program Screen window is displayed.
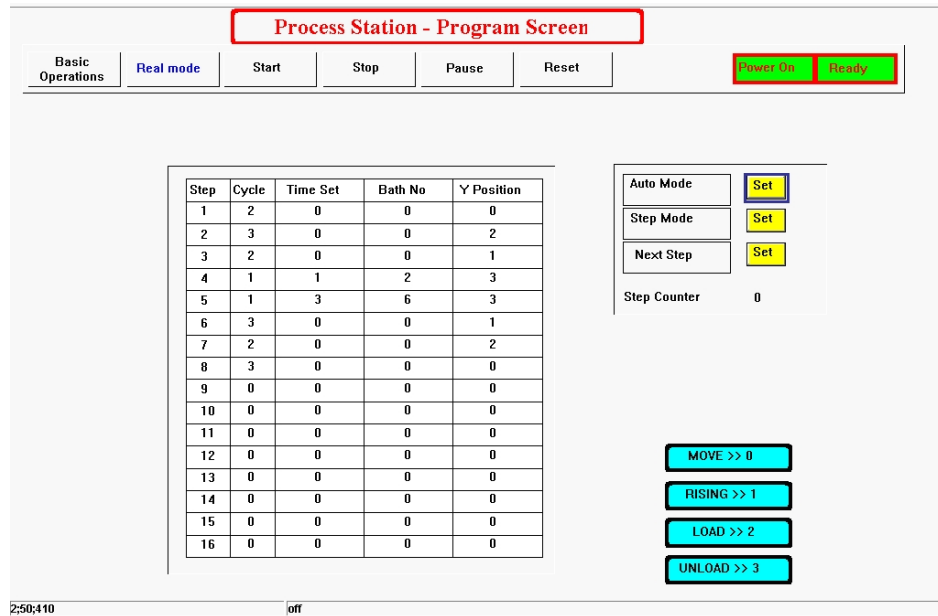
Figure 9-50: Process Station – Program Screen
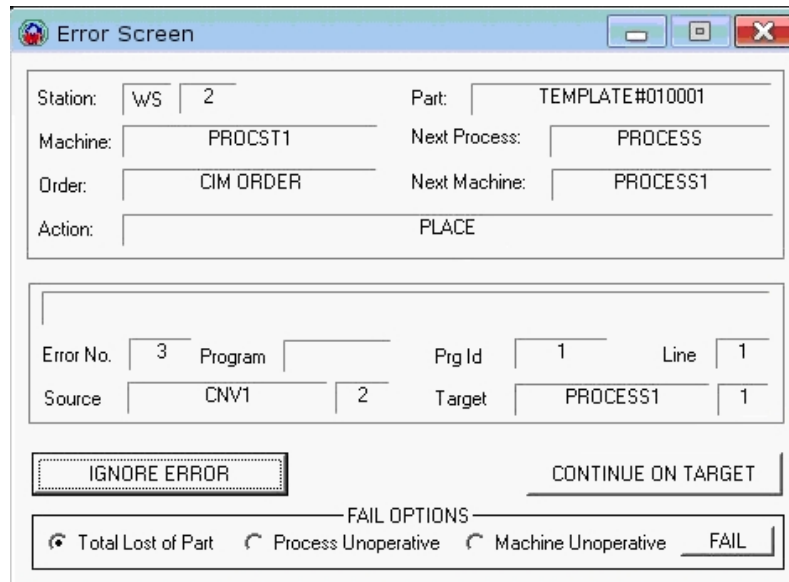
3. Verify that the button is green to indicate that the communication between the process station and the computer is working properly.

☞Note   *If the* ▢ *button is not displayed in green press* ▢ *or press the PLC ON button on the front panel of the process station.*

You can now view the various stages of the process, such as the amount of time that a specific part remains in a specific bath and so on.

Process Station Troubleshooting

Whenever a problem is detected with the operation of the process station device driver in the OpenCIM environment, the process station Error led is automatically turned on and an Error Screen appears in the CIM Manager application (as shown in the example below).
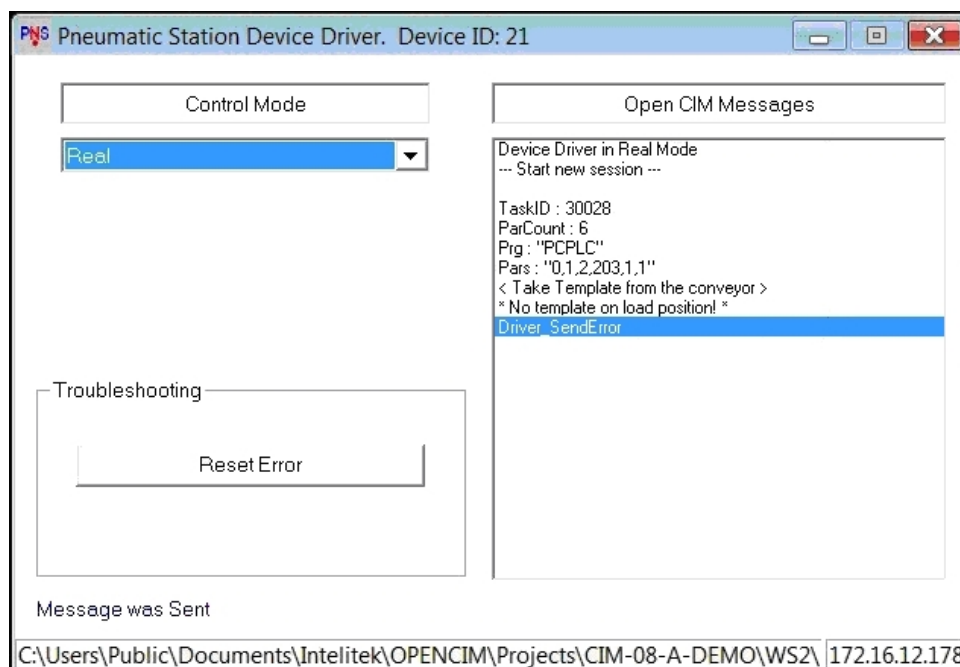
Figure 9-51: CIM Manager Error Screen

To correct process station errors:

1. From the CIM Manager Error Screen, click IGNORE ERROR. The Process Station Device Driver window appears displaying the error that occurred in the system.

Figure 9-52: Process Station Device Driver - Error Message

2. Right-click the **Vuniq** icon. A popup menu is displayed.

3. Select **Operator Screens**. The Main Process Station window is displayed, as shown in, *Figure 9-47: Main Process Station*.

---

4. From the Main Process Station window, deselect the [Real mode] button and then select the [Basic Operations] button to enter basic operations mode.

5. From the General area, select the [Control] button that is located to the right of the Manipulator Control option. The Process Station - Manipulator Control Screen is displayed.
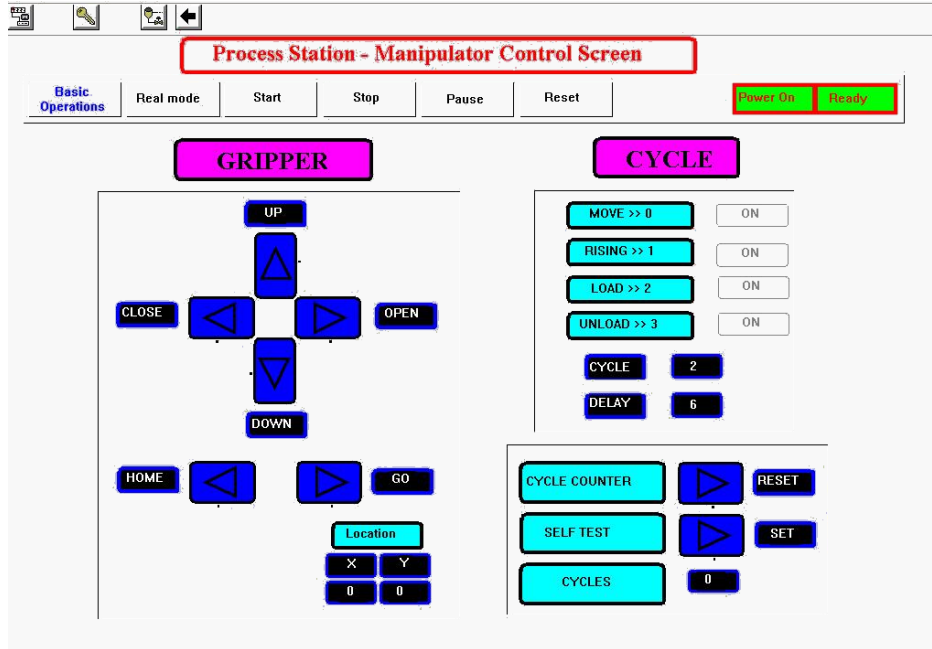
Figure 9-53: Process Station - Manipulator Control Screen

6. From the Gripper area, select the arrow [◁] button that is located to the right of the [HOME] option. The robot is sent to the home position.

7. Deselect the [Basic Operations] button and then select the [Real mode] button to enter Real Mode.

8. Click the icon to return to the Main Process Station window (shown in Figure 9-47: Main Process Station).

9. From the CIM Manager main window, select the red **Stop** icon to stop the processes in the OpenCIM environment.

10. Correct the problem manually and then press the green **Start** icon to restart the process.

10

Web Viewer

The Web Viewer application enables you to remotely access a specific CIM Manager cell and track the production cycle of the CIM cell from the various view tabs. This includes, the graphic display of the operations being performed, the actions performed on system devices as they occur in the production cycle, information regarding the storage locations and more.

This chapter describes how to install and access the Web Viewer client. It includes the following:

- **Installing the Client**, describes how to install the web viewer client application.
- **Accessing the Web Viewer**, describes how to activate the web viewer application.
- **Web Viewer Main Window**, introduces the elements of the web viewer main window.

Note

Before accessing the Web Viewer, you must first verify that the IIS is installed and that the the DCOM has been configured (described in Chapter 4, Installation). You must then verify that the Web Viewer mode is activated (described in CIM Modes, Chapter 6, Operating CIM Manager).

Installing the Client

The Web Viewer client is installed through Internet Explorer (version 5.0 and higher). If required you can save the installation file to a specified directory for future installations.

To install the Web Viewer client:

1. From your **Start** menu, click the ![IE icon] **Internet Explorer** icon. The Internet Explorer main window is displayed.

2. In the Address field enter the IP or hostname of the computer containing the CIM Manager that you want to install from, as in the following examples:

   - http://<hostname>/webcimviewer/start.asp

   OR

   - http://<IP Address>/ webcimviewer/start.asp

The OpenCIM Web Viewer Initial page is displayed.
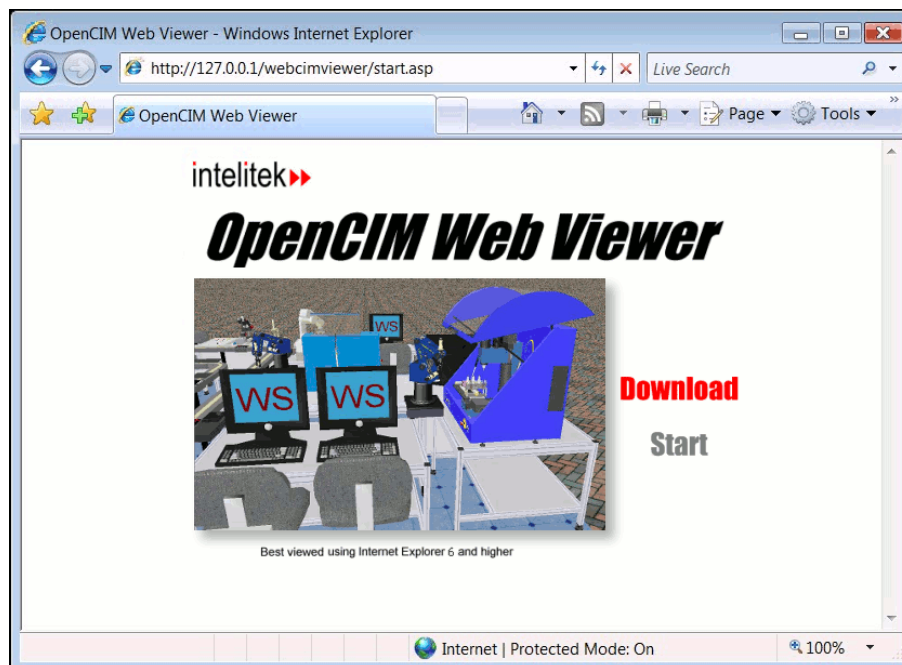
Figure 10-1: Web Viewer Installation Window

3. Click **Download** the File Download window is displayed.
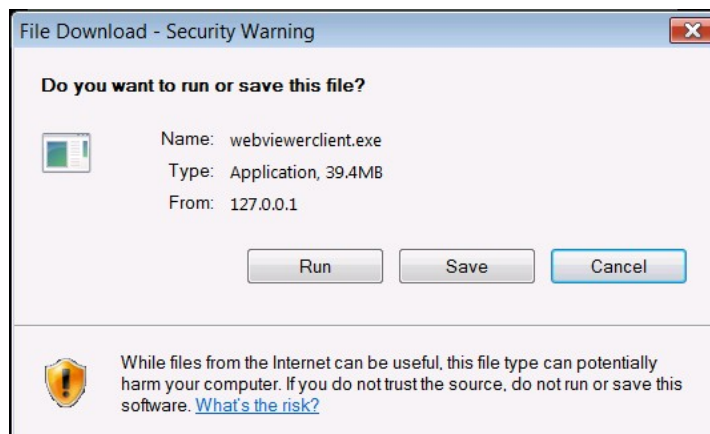
Figure 10-2: Web Viewer Download Window

4. Select the required option, as follows:

- Select **Run** in order to run this program from its current location.

- The Web Viewer Client Setup window is displayed. When the installation is complete, click **Finish**.

    OR

- Select **Save** in order to save this program to a specified directory for future installations.

After you have installed the client the next step is to access Web Viewer, as described in Accessing the Web Viewer.

Accessing the Web Viewer

Before accessing the Web Viewer, ensure that the CIM Manager that you want to access is currently open.

To access the Web Viewer:

1. From your **Start** menu, click the **Internet Explorer** icon. The Internet Explorer main window is displayed.

2. In the Address field enter the IP or hostname of the computer containing the CIM Manager that you want to access according to the following examples:

   - http://<hostname>/webcimviewer/start.asp

   OR

   - http://<IP Address>/webcimviewer/start.asp

The OpenCIM Web Viewer Initial window is displayed, as shown in Installing the Client.

3. Click **Start**. The Open CIM Web Viewer appears displaying the Graphic Display tab, as shown in *Graphic Display Tab*.

Note

If the CIM Manager of the computer that you want to access is currently inactive, an information dialog box is displayed, informing you that CIM Manager that you currently want to access is currently closed.

Web Viewer Main Window

The Web Viewer main window contains the following elements, each of which is described in the sections that follow:

- **Graphic Display Tab,** displays a 3D display of the operation being performed in the CIM cell.
- **View Scheduler Tab**, displays the scheduler information of the CIM cell.
- **View Program Tab,** displays the A-Plan (meaning, the production work order) of the CIM cell.
- **View Leaf Tab**, displays the production activities in the CIM cell.
- **View Order Tab**, displays the current manufacturing order.
- **View Storage Tab**, displays the current location of parts in the CIM cell.
- **View Device Tab**, displays the actions performed by system devices.
- **View Pallet Tab,** displays the pallets in the CIM cell the current status of each pallet.
- **Web Viewer Status Bar,** displays CIM cell information, such as current status, elapsed time and more.
- **View About Tab,** displays dialog with information about Web CIM Viewer software and CIM software of the Cell that is current in view.

Graphic Display Tab
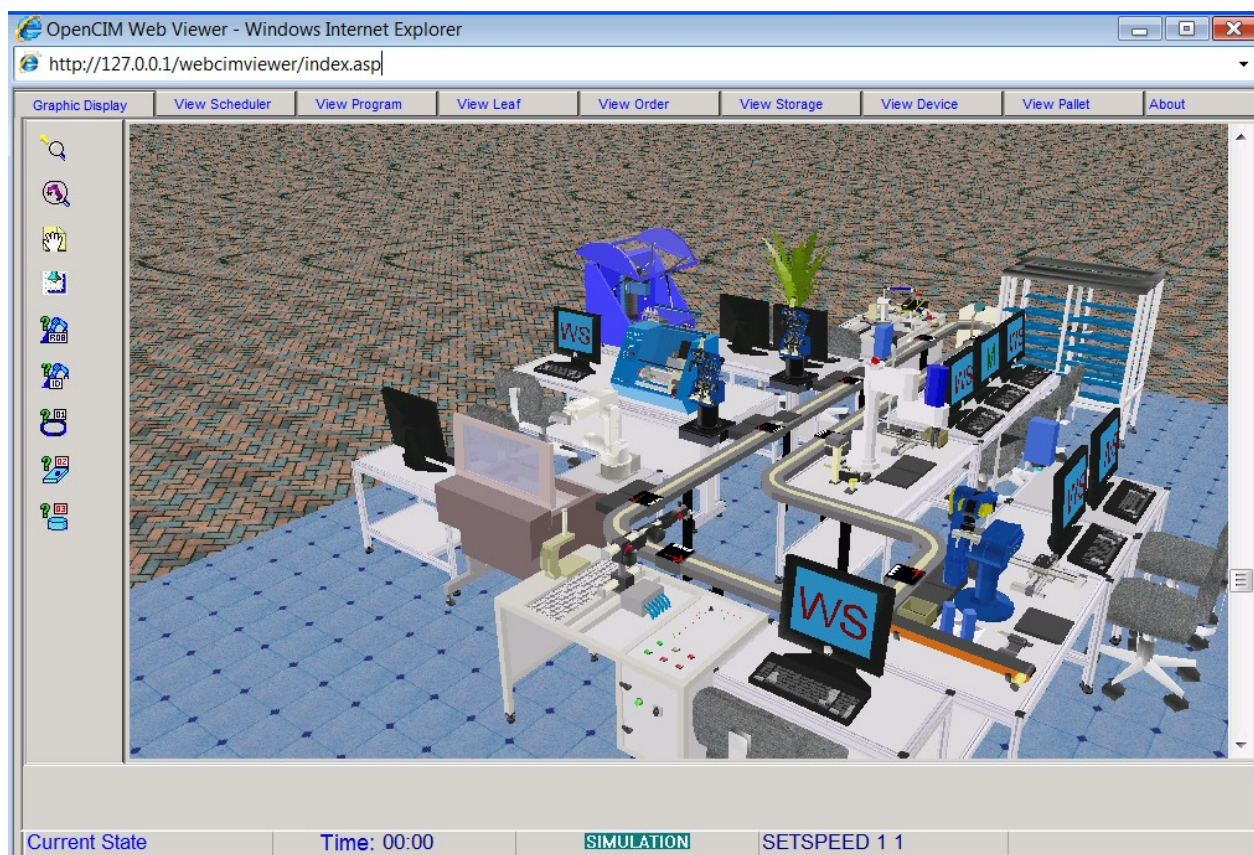
Figure 10-3: Graphic Display Tab

The Graphic Display tab enables you to view the graphic display that currently appears on the CIM Manager of the computer that you are currently accessing. For further information refer to *Graphic Display and Tracking* in *Chapter 6, Operating CIM Manager*.

View Scheduler Tab
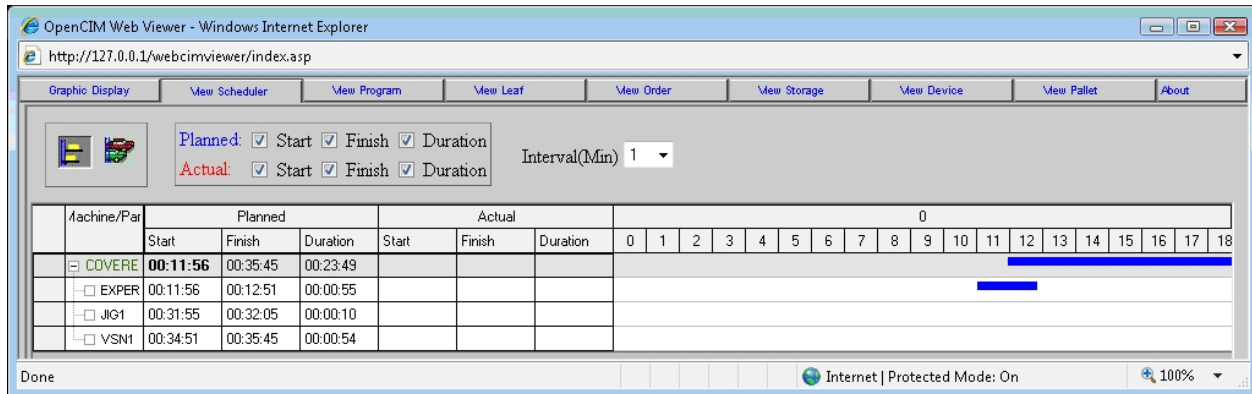
Figure 10-4: View Scheduler Tab

The View Scheduler tab enables you to view the current (planned and actual) scheduler information of the CIM Manager that you are accessing. For further information refer to CIM Scheduler in Chapter 6, Operating CIM Manager.

View Program Tab
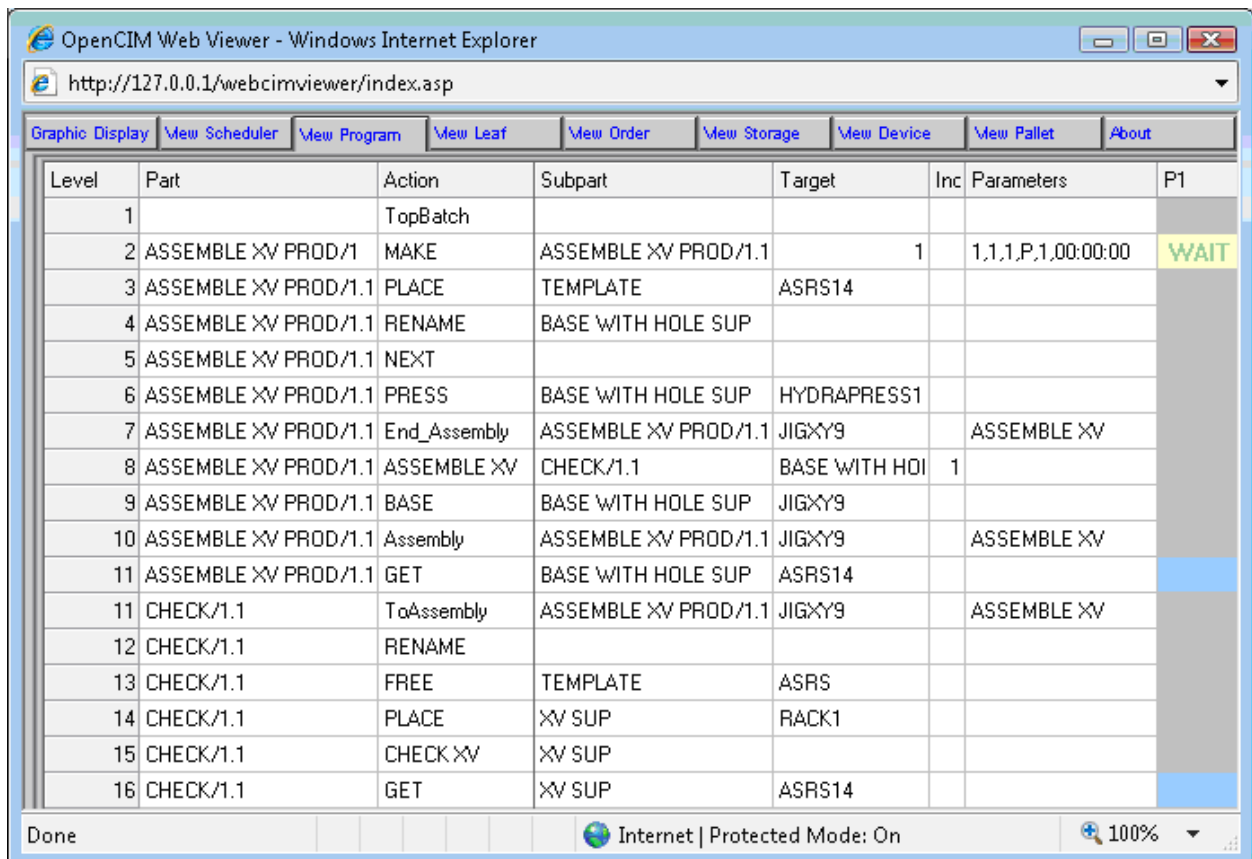
Figure 10-5: View Program Tab

The View Program tab enables you to view the current production work order of the CIM Manager that you are currently accessing. This enables you to track the current production status of the CIM Manager. For further information refer to *Program View* in *Chapter 6, Operating CIM Manager.*

View Leaf Tab

Figure 10-6: View Leaf Tab

The View Leaf tab enables you to view the description of the current production activities in the current CIM cell of the CIM Manager that you are accessing, describing the current operation being performed on each item as well as the operation that will follow immediately. For further information refer to Leaf View in Chapter 6, Operating CIM Manager.

View Order Tab

Figure 10-7: View Order Tab

The View Order tab enables you to view a copy of the manufacturing order of the current cell of the CIM Manager that you are accessing. For further information, refer to Order View in, Chapter 6, Operating CIM Manager.

View Storage Tab

Figure 10-8: View Storage Tab

The View Storage tab enables you to view a list of every storage location defined in the CIM system. For further information, refer to Storage View in Chapter 6, Operating CIM Manager.

View Device Tab
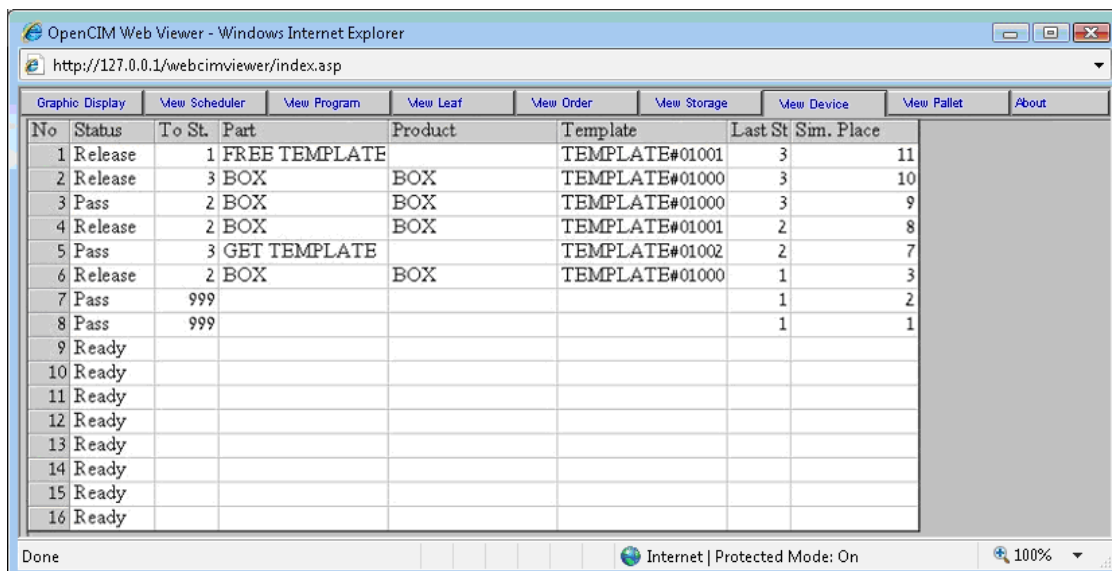
Figure 10-9: View Device Tab

The View Device tab enables you to view a list of every robot and machine (including QC devices) as well as a description of the current action being performed in the current CIM Cell of the CIM Manager that you are accessing. For further information, refer to Device View in Chapter 6, Operating CIM Manager.

View Pallet Tab
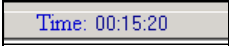
Figure 10-10: View Pallet Tab

The View Pallet tab enables you to view a list of every pallet in the CIM Cell and a description of its current status. For further information, refer to Pallet View in Chapter 6, Operating CIM Manager.

Web Viewer Status Bar

The Web Viewer status bar consists of the following elements:

Option	Description
Status:	Displays information regarding the status of the CIM Manager that you are accessing, as follows: • Manager is Open: Displayed as soon as the CIM Manager is opened. • Manager is Closed: Displayed when the CIM Manager is closed. • Synchronizing Graphics: Displayed when synchronizing graphics of the current state of the CIM Manager that you are acccesing. • Current State: Displayed when the current graphic state of the CIM Manager is displayed in the Graphic Display tab.
Time:	The time that has passed since the start of production in the CIM Manager.
Mode:	CIM Manager modes of operation: • **Simulation Mode:** The CIM Manager does not communicate with device drivers. This mode does not require either hardware or device drivers in order to operate. • **Real Mode:** The CIM Manager commmunicates with all device drivers, whether or not hardware is in use. This mode requires that all device drivers which are need for a specfic application be loaded, so that the CIM Manager can transmit and receive messages.
Graphic Messages:	Graphic messages that appear describing the graphic process. The **Order is Completed** message appears at the end of the production process, indicating the order has been complete.

View About Tab

The Web Viewer About tab displays a dialog with information about CIM software and Web CIM Viewer software, and a shortcut to the Intelitek web site for support and information.

Figure 10-11: View About Tab

11

OpenCIM Programming

This chapter provides various programming and advanced OpenCIM features. It includes the following sections:

- **Robotic Programming for OpenCIM**, describes the robotic programming commands in OpenCIM.
- **CNC Programming for OpenCIM**, describes the CNC programming commands in OpenCIM.
- **Robot and CNC Interface**, describes the CNC synchronization mechanism in OpenCIM.
- **Optimization Enhancement Using Open Source**, describes how to add new algorithms to the existing list of algorithms in the CIM Optimization Manager.
- **Experimenting with Production Strategies Using the A-Plan**, describes the structure of the A-Plan and provides instructions how to modify it.

Robotic Programming for OpenCIM

In the course of production, OpenCIM uses a set of robotic programs which control the movement of robots and the operation of peripheral devices connected to a robotic controller. When you want to teach a robot (or other device) to perform a new task or to achieve better performance at a task, you need to edit or create robotic programs. This need arises when you:

- Install a new device at a station.
- Move a device or a robot to a new location.
- Add or change a process or part definition (in the Machine Definition or Part Definition utility programs) that relies on a robotic program.
- Add or change robot tasks or parameters. For example: speed (slow, medium, fast) or the way the robot moves (linear, circular, etc.).
- Add or change the parameters of the devices attached to the robotic controller.

This section describes how to write robotic programs in the OpenCIM environment. These programs direct a robot (or other device attached to a robotic controller) to:

- Move an object from place to place (pick-and-place operation).
- Perform certain system functions (e.g. how to react in case of a robot collision).
- Perform some other production process (e.g. bar code, pneumatic screwdriver, CNC interface, etc.)

The Pick-and-Place Strategy

OpenCIM uses the pick-and-place strategy to minimize the number of custom robotic programs required to transfer parts between locations at a station.

Having fewer programs yields several benefits:

- Less programming effort to write the original programs.
- Fewer changes to be made in the event devices are added, deleted, or moved.
- Fewer problems and easier to debug since all GET and PUT programs share a common structure.
- Requires less memory in the robotic controller

Instead of writing individual robotic programs to move a part between two locations (a point-to-point approach), the pick-and-place strategy provides a more systematic method that requires only two programs for each location, a GET and PUT. A GET program picks up a part from a location. A PUT program places a part at a specific location. GET and PUT programs for different locations are designed to work together to allow the robot to take a part from any location (GET) and deliver it to any other location (PUT).

For example, if there are six locations at a station, it would require 30 point-to-point programs to cover all the possible combinations of moving a part between any two locations. The pick-and-place approach requires only 12 programs (6 GETs and 6 PUTs).

A Free Movement Zone is the key to getting GET and PUT programs to work together. This zone is a region approximately ½ meter above work surface in which the robot can move freely and quickly between locations without encountering any obstacles.

The first operation in a GET or PUT program is to move the robot in a straight line to a position directly above the location where it is needed. From there, a program uses a set of detailed robotic commands to maneuver the robot arm to a point where it can pick up or place a part at a specific location. When a GET or PUT begins executing, it assumes that the robot arm is in the Free Movement Zone waiting for a command.

A typical pick-and-place scenario is shown in the figure below. In this scenario, the CIM Manager sends a command to move a part from buffer 2 to a ROBOTVISIONpro camera for a quality control test. This pick-and-place command is sent to the robotic device driver. The device driver in turn sends commands to the robotic controller to run the corresponding GET and PUT programs described below.
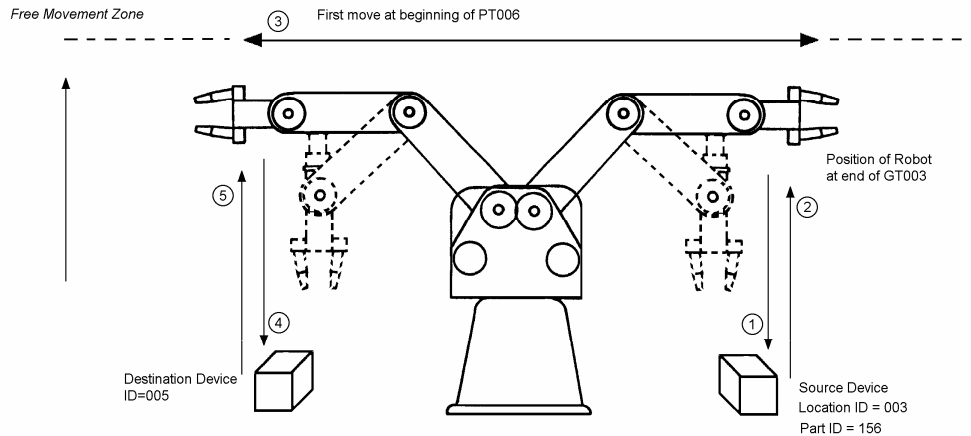
Figure 11-1: Robot Movements Controlled by Separate GET and PUT Programs

GET (from buffer 2)
1. Move the empty robot arm in a straight line through the Free Movement Zone to a point directly above the buffer.
2. Lower the arm and grab the part from the template sitting in the buffer.
3. Raise the arm back up to the Free Movement Zone directly above the buffer.

PUT (under the camera)
1. Move the robot arm holding the part through the Free Movement Zone to a point above the ROBOTVISIONpro camera's viewing area.
2. Lower the arm and set the part within the camera's field of view.
3. Raise the arm back up to the Free Movement Zone directly above the camera.

At end of the GET, the robot is holding the part in the Free Movement Zone while it waits for the PUT program to begin executing.

At the end of the PUT, the robot is empty and it waits in the Free Movement Zone for the next GET operation.

Overview of a Pick-and-Place Command

The CIM Manager tells a robot to move a part/template from one device at a station to another by sending a pick-and-place command to the appropriate robotic device driver. The device driver tells the controller to run the robotic programs GET and PUT that are associated with the locations specified in the pick-and-place message.

Each device has a GET program associated with it which tells a robot how to move in order to pick up a part at this location. Similarly, each device has a PUT program which tells a robot how to place a part at this location. The names of these robotic programs take the form of GTxxx and PTxxx (for ACL programs) and GETxxx and PUTxxx (for Scorbase programs), where xxx is the ID of the device.

The device driver tells the controller to run the appropriate GTxxx and PTxxx that are associated with the locations specified in a pick-and-place command.

Each GET program is dedicated to picking up an object from a single location. Each PUT program is dedicated to delivering an object to a single location.

In order to move a part from any location at a station to any other location, all GET and PUT programs are designed to be used together in any combination.

For example, to move a template from the ASRS to a pallet waiting on the conveyor, a pick-and-place command would specify running the following robotic programs:

- GT002 - Take template from ASRS (002 = ASRS device ID).
- PT001 - Put template on conveyor pallet (001 = device ID for conveyor)

Note that the device IDs for GET and PUT are different. If they were the same this would mean that the robot was returning the part/template to the same location where it had just picked it up.

All GET and PUT programs for a robot must be designed to work together. This entails that:

- They read the same set of pick-and-place parameters (stored in global variables).
- When a program ends, it must leave the robot in a position that enables it to move in any subsequent direction (since you do not know at the time of writing the programs where the next GET or PUT will send the robot).
- They use the same synchronization mechanism which allows a GET program to activate any PUT program.

Note
The names of the Robotic programs are different for ACL programs and Scorbase programs, as follows:
- *GTXXX and PTXXX are used for ACL programs.*
- *GETXXX and PUTXXX are used for the Scorbose programs.*

Teaching Robot Positions

The path that a robot follows is made up of points called robot positions. These positions can be "taught" using a teach pendant or robotic software, for example, ACL or Scorbase. The coordinates associated with these positions are normally stored in the required program file See the robotic documentation for a complete discussion of how to teach robot positions.

☞ **Tip**
When planning robot movements, take into consideration obstacles caused by parts under production. For example, after placing a part, the robot may not be able to retrace its movements without colliding with the part it just placed.

Also note that the same part at the same location may require two different robot positions before and after an assembly operation.

Response Messages (from Robotic Programs to the CIM Manager)

The programs GET and PUT send the Start, Finish and End status messages to the CIM Manager via the robotic device driver, each of these messages is described in the table below.

Status Message	Description
Start	The GET program sends a Start message to report that the robot has grasped the part/template and has moved clear of the source device. This Start message indicates that the source device can continue with other processing even while the robot continues to move.
	The Start message can speed up time critical operations in the CIM. For example, strategic placement of the Start message can be used to expedite the flow of pallets on the conveyor. As soon as a robot has lifted a template from a pallet, the GET program sends a Start message. The CIM Manager can then release the pallet even while the robot continues to move the template.
Finish	The PUT program sends a Finish message to report that the robot has placed the part/template at the destination device. The Finish message indicates that the destination device can now proceed to process the part/template even while the robot continues to move back to its idle position.
	The Finish message can be used to speed up time-critical operations in a manner similar to the Start message. For example, as soon as a robot has placed a template on a conveyor pallet and moved out of the way, the PUT program can send a Finish message. The CIM Manager can then release the pallet even while the robot continues to move to its final resting position.
End	The PUT program sends an End message to report that the robot has completed this pick-and-place operation and is now available for the next command.

☞ **Tip**
In order to avoid delaying the conveyor unnecessarily, send the Start and Finish messages as soon as possible when writing GET and PUT programs that deal with moving a part template to/from a conveyor pallet.

For most other devices, the Finish and End messages typically come one right after the other at the end of the PUT program.

Sending Response Messages from ACL Programs to the CIM Manager

The code that actually sends the response messages from the ACL program is contained in the macros .START, .END, .FINISH. These are found in the PROGRAM_GET XXX and PROGRAM_PUT XXX example programs.

To send response messages from ACL Programs to the CIM Manager:

- Insert the following macros in the appropriate places in your own GET and PUT programs:

- Start: The Start message is sent by the macro **.START**.

- Finish: The Finish message is sent by the macro **.FINISH.**

- End: The End message is sent by the macro **.END**.

The macros for all three messages, Start, Finish, and End, take care of returning the command sequence number stored in the parameter variable $ID. This value allows the CIM Manager to identify the source of the message.

Sending Response Messages from the Scorbase Software to the CIM Manager/Device Driver

The Send Message window is used for creating the Send Message command in the Scorbase program. When the program is activated and begins to run, the Send Message command is then forwarded to the CIM Manager or device driver, as described in the following procedures:

To send response messages from Scorbase to the CIM Manager:

1. From the Scorbase main window, select **Window | Teach & Edit**. Select **Options | Pro** from the Menu bar. From the Workspace window expand the **Program Flow** tree and double click **SendMessage…**. The Send Message window is displayed:
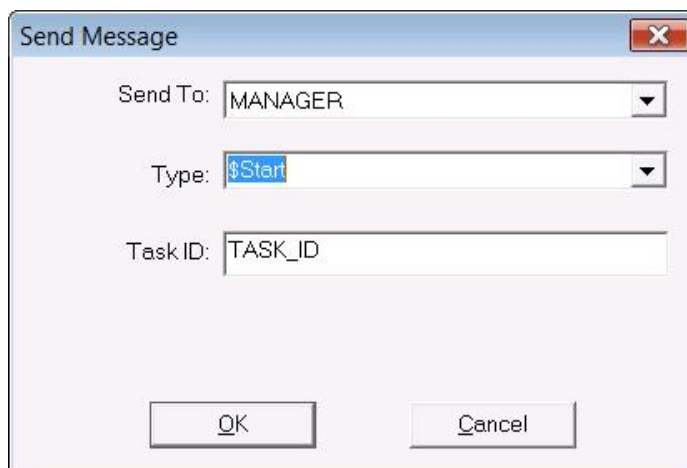
Figure 11-2: Send Message Window – CIM Manager Option Selected

2. From the **Send To** dropdown list select the **MANAGER** option.

Note

*In the **Send To** dropdown list the **Manager** option is displayed by default. In the **Task ID** field the **TASK_ID** option is displayed by default.*

3. From the **Type** dropdown list select the required response message as follows:

- $Start
- $Finish
- $End

4. Click **OK.** The response message is displayed in the program window.

Note

This procedure is applicable only when the Scorbase device driver is in online mode.

To send response messages from Scorbase to the Device Driver:

1. From the Scorbase main window, select **Window | Teach & Edit**. Select **Options | Pro** from the Menu bar. From the Workspace window expand the **Program Flow** tree and double click **SendMessage…**. The Send Message window is displayed.
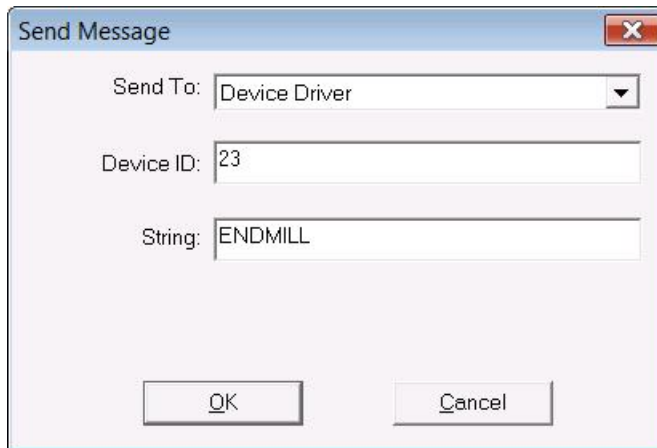
Figure 11-3: Send Message Window – Device Driver Option Selected

2. From the **Send To** dropdown list select the **Device Driver** option.
3. In the **Device ID** field enter the ID of the device.
4. In the **String** field enter the string that the Device Driver is waiting to receive.
5. Click **OK**. The response message is displayed in the program window.

Note   *This procedure is applicable only when the Scorbase device driver is in online mode.*

Viewing the Response Messages in CIM Manager

You can monitor the progress of robotic programs at run-time by looking at the Program, Leaf or Device Views in the CIM Manager program. When a robot is performing a pick-and-place command, the following messages let you follow the progress of the GET and PUT programs as they execute.

Run-Time Message	Description
ON	The **Start** macro in the GET program has executed.
OFF	The **Finish** macro in the PUT program has executed.
Blue Box	The **End** macro in the PUT program has executed.

Pick-and-Place Parameters

The CIM Manager sends a set of parameters to a robotic device driver whenever it issues a pick-and-place command. The device driver in turn activates the PCPLC program in the robotic controller which receives these parameters and assigns them to the following global variables:

Robotic Pick & Place Parameters		Description
Scorbase	**ACL**	
TASK_ID	$ID	A sequence number generated by the CIM Manager for each command (a pick-and-place command in this case). Whenever a robotic program sends a status message, it includes this command ID so that the status message can be associated back to the original command.
PART_ID	PART	The ID number of the part that the robot is to handle. This number corresponds to the Part ID field in the Part Definition form. For each part type, the instructions for how the robot grasps the part is defined in the robotic program (ACL or Scorbase). For example, the positions for each part ID is stored in an array (such as, CIM[ ]). The part ID can be used to calculate an index into this array.
		A template is identified with a Part ID of zero.
SOURCE_DEVICE_ID	$DEV1	The device ID of the source location where the robot will pick up the part/template.
SOURCE_DEVICE_INDEX	INDXG	For a source device that has multiple compartments (e.g. a storage rack), this parameter specifies in which compartment (or buffer) the robot will find the part/template.
TARGET_DEVICE_ID	$DEV2	The device ID of the target location where the robot will place the part/template.
TARGET_DEVICE_INDEX	INDXP	For a target device that has multiple compartments (e.g. a storage rack), this parameter specifies in which compartment (or buffer) the robot should place the part/template.
PICK_AND_PLACE_NOTE	$NOTE	This parameter is available for user-defined purposes to send special instructions to a GET or PUT program. For example, when the part to be handled is a template (with ID=0), the note parameter will indicate the part ID of the part on the template. This enables the Scorbase programmer to include special instructions for specific parts, such as, slower movements for sensitive parts and more.

Writing Scorbase Source Code

The Scorbase program consists of two parts. The first part (lines 1 through 14) is created automatically when you select the parameters from the Pick and Place window, the second part (from line 15 to the end) contains GET XX and PUT XX programs for each device in the CIM system. These GET XX and PUT XX programs are written by the programmer.

An example of a Scorbase program is displayed, as follows:

```
Remark: $ Beginning of the automatically generated code
Call Subroutine $PICK_AND_PLACE_0,1,3,15,2,0
Set Subroutine $PICK_AND_PLACE_0,1,3,15,2,0
Set Variable TASK_ID = 110000
Set Variable PART_ID = 0
Set Variable SOURCE_DEVICE_ID = 1
Set Variable SOURCE_DEVICE_INDEX = 3
Set Variable TARGET_DEVICE_ID = 15
Set Variable TARGET_DEVICE_INDEX = 2
Set Variable PICK_AND_PLACE_NOTE = 0
Call Subroutine AUTOEXEC
Call Subroutine GET001
Call Subroutine PUT015
Return from Subroutine
Remark: $ End of the automatically generated code
Remark: Intelitek Open CIM robot device driver demonstration
Remark: The ER-4u robot serves the Conveyor (CNV1)
Remark: The ER-4u robot serves the MINI-ASRS (M6AS2)
Set Subroutine GET015
Print to Screen: GET TEMPLATE FROM MINI-ASRS (M6AS2)
Go to Position 21 Speed 5
Open Gripper
Set Variable ABOVE_TEMPLATE_POSITION = SOURCE_DEVICE_INDEX +
110
Go to Position ABOVE_TEMPLATE_POSITION Speed 8
Set Variable AT_TEMPLATE_POSITION = ABOVE_TEMPLATE_POSITION -
100
Go Linear to Position AT_TEMPLATE_POSITION Speed 2
Close Gripper
Go Linear to Position ABOVE_TEMPLATE_POSITION Speed 2
Go to Position 21 Speed 5
Go to Position 20 Speed 5
Go Linear to Position 2 Speed 2
Open Gripper
Go to Position 20 Speed 5
Send Message $Start to MANAGER for task TASK_ID
Return from Subroutine
Set Subroutine PUT015
Print to Screen: PUT TEMPLATE ON MINI-ASRS (M6AS2)
```

```
Open Gripper
Go to Position 20 Speed 5
Go Linear to Position 2 Speed 3
Close Gripper
Go Linear to Position 20 Speed 2
Go to Position 21 Speed 5
Set Variable ABOVE_TEMPLATE_POSITION = SOURCE_DEVICE_INDEX +
110
Go to Position ABOVE_TEMPLATE_POSITION Speed 8
Set Variable AT_TEMPLATE_POSITION = ABOVE_TEMPLATE_POSITION -
100
Go Linear to Position AT_TEMPLATE_POSITION Speed 2
Open Gripper
Go Linear to Position AT_TEMPLATE_POSITION Speed 2
Go to Position 21 Speed 5
Send Message $Finish to MANAGER for task TASK_ID
Go to Position 20 Fast
Send Message $End to MANAGER for task TASK_ID
Return from Subroutine
Set Subroutine GET001
Print to Screen: GET TEMPLATE FROM CONVEYOR (CNV1)
Open Gripper
Go to Position 10 Fast
Go Linear to Position 1 Speed 3
Close Gripper
Go Linear to Position 10 Speed 2
Go to Position 20 Speed 5
Go Linear to Position 2 Speed 3
Open Gripper
Go Linear to Position 20 Speed 2
Send Message $Start to MANAGER for task TASK_ID
Return from Subroutine
Set Subroutine PUT001
Print to Screen: PUT TEMPLATE ON CONVEYOR (CNV1)
Go to Position 20 Speed 5
Go Linear to Position 2 Speed 2
Open Gripper
Send Message $Finish to MANAGER for task TASK_ID
Go Linear to Position 20 Speed 3
Send Message $End to MANAGER for task TASK_ID
Return from Subroutine
```

Note *If the programmer defined an AUTOEXEC subroutine in the program, then the* `Call AUTOEXEC SUBROUTINE` *will appear in the automatically generated section of the program (lines 1 through 14).*

Writing ACL Source Code

The Robot programs are comprised of all the associated programs necessary to run the robot. The Robot programs are divided into individual programs as follows:

- Get and Put programs of each device.
- Quality Control programs of each peripheral QC device (e.g. Bar code).
- Process (PRL programs) programs of each peripheral machine (e.g. automatic screwdriver, CNC machine)

Structure of the ACL Program

File	Description
CIMSYS.DMC	Located in the Projects\LIB\ACL LIB/ACL directory. This file contains system macros.
CIMSYS.SYS	Located in the Projects\LIB\ACL LIB/ACL directory. This file contains system programs.

All of the following files are located at the current directory of Robotn:

File	Description
WSn.DNL	Includes all of the files that are to be downloaded from your station PC to the ACL controller.
PROLOGn.DNL	In this file you modify system programs, such as Global System variables (e.g. SPSA - speed slow group A), predefined positions or any other settings that need to be predefined for that station. This file contains all the documentation for positions and I/O.
GET/PUT.DNL	Pick-and-place programs.
QC.QCL	Quality Control programs.
PROCESS.PRL	Process programs.
EPILOGn.DNL	In this file you modify Initialization System programs.

You can manage all of your Robot programs through the Robot Programs window. Any editing changes, however small, should be made to the original DNL, QCL or PCL files.

Device Definition

The Setup directory contains a file which defines numbers to their ACL logical name. This file includes all the devices in the system. For example:

```
#IFNDEF _DEVICE_DMC
  #DEFINE _DEVICE_DMC
  #DEFINE CNV1          001
  #DEFINE ASRS          003
  #DEFINE ASMBUF        004
  #DEFINE BFFR1         005
  #DEFINE FDR1          006
  #DEFINE TRASH1        007
  #DEFINE RACK1         008
  #DEFINE RDR1          009
  #DEFINE LATHE1        010
  #DEFINE MILL1         011
#ENDIF
```

ACL Macro Programs

The Projects\LIB\ACL directory contains the file **CIMSYS.DMC**. This file includes the system programs and macros which you will use in writing your own ACL programs:

- Part ID Group (PID)
- Robot Movements
- Synchronization
- Open/ Close Gripper
- Speed
- GET/ PUT Programs
- QC Programs
- Process Programs

The following listing is the source code of CIMSYS.DMC. This file contains system programs, macros, and global variable definitions that are needed when writing your own ACL programs.

```
;
;                         OpenCIM-ACL Program
;
;                      This Is The Cim Macro File
;                          For All Stations

;Global definitions
#IFNDEF  _CIMSYS_DMC
  #DEFINE _CIMSYS_DMC
  #DEFINE _CIMSYSM_DMC

  ;Part  macro
  #MACRO PID                ; part id  1=1..10    2=11..20   etc.
       SET PID = PART - 1
       SET PID = PID / 10
       SET PID = PID +1
  #ENDM

  ;Group B movement macros
  #IF  .__GROUP_B
                       ;Check if Group B is Define

    #MACRO     MOVEDB
      MOVED      .1 .2
    #ENDM

    #MACRO     MOVEB
      MOVE       .1 .2
    #ENDM
  #ELSE
    #MACRO     MOVEDB
    #ENDM

    #MACRO     MOVEB
    #ENDM
  #ENDIF

  ;Synchronize macros
  #MACRO   STARTSYNC
    SET       $SYNC = 0
  #ENDM

  #MACRO  SYNC          ;Should be at the beginning of
                        ;all the put programs.
    WAIT     $SYNC = 1   ;Wait to the end of get.
    SET      $SYNC = 0
  #ENDM

  #MACRO  ENDGET         ;At the end of all the get programs.
     SET      $SYNC = 1  ;Can start the put program.
  #ENDM

  ;Gripper macros
  #MACRO OPEN             ;Open the gripper.
    GOSUB  OGRIP
  #ENDM

  #MACRO CLOSE            ;Close the gripper.
    GOSUB  CGRIP
  #ENDM

  ;Speed macros

  #MACRO FAST             ;Set group A and B speed to fast.
    .FASTA
    .FASTB
```

```
#ENDM

#MACRO MEDIUM          ;Set group A and B speed to medium.
  .MEDIUMA
  .MEDIUMB
#ENDM

#MACRO SLOW            ;Set group A and B speed to slow.
  .SLOWA
  .SLOWB
#ENDM

#MACRO FASTA           ;Set group A speed to fast.
  SPEEDA    SPFA
#ENDM

#MACRO MEDIUMA         ;Set group A speed to medium.
  SPEEDA    SPMA
#ENDM

#MACRO SLOWA           ;Set group A speed to slow.
  SPEEDA    SPSA
#ENDM

#IF   .__GROUP_B
                       ;Check if group B is define.
  #MACRO FASTB         ;Set group B speed to fast.
    SPEEDB    SPFB
  #ENDM

  #MACRO MEDIUMB       ;Set group B speed to medium.
    SPEEDB   SPMB
  #ENDM

  #MACRO SLOWB         ;Set group B speed to slow.
    SPEEDB   SPSB
  #ENDM
#ELSE
  #MACRO FASTB         ;Set group B speed to fast.
  #ENDM

  #MACRO MEDIUMB       ;Set group B speed to medium.
  #ENDM

  #MACRO SLOWB         ;Set group B speed to slow.
  #ENDM
#ENDIF

;Programs macros  (for GTxxx and PTxxx).
#MACRO   PROGRAM       ;The header of all the programs.
  PROGRAM  .1
  DEFINE    $I
#ENDM

#MACRO   PROGRAM_GET   ;The header of get programs.
  PROGRAM  GT.1
  DEFINE    $I
  SET       $I = 1
  .STARTSYNC
#ENDM

#MACRO   PROGRAM_PUT   ;The header of put programs.
  PROGRAM  PT.1
  DEFINE    $I
  SET       $I = 1
#ENDM

#MACRO   END_GET       ;The tail of get programs.
  .ENDGET
```

```
      END
   #ENDM


   #MACRO    END_PUT        ;The tail of put programs.
      END
   #ENDM


   ;Controller D.D protocol macros
   #MACRO GETID            ;Get ID from D.D to $ID
     LABEL  11
     READ "?%ID?" $ID
     IF     $ID = 0        ;Check if the ID is zero (invalid).
       GOTO    11
     ENDIF
     SET    $I = 1
   #ENDM

   #MACRO GETIDL       ;Get local ID for a not PCPLC program.
     LABEL  11
     DEFINE $IDL        ;$IDL hold the local ID (instead of $ID).
     READ "?%ID?" $IDL
     IF     $IDL = 0   ;Check if the ID is zero (invalid).
       GOTO    11
     ENDIF
     DEFINE $I
     SET    $I = 1
   #ENDM

   #MACRO GETPAR          ;Get Parameter from D.D.
     PEND $PDF FROM $PDD
     PRINTLN
     READ "?%PAR?" .1
     PRINTLN
     POST 1 TO $PDD
   #ENDM

   ;Controller manager protocol macros for global ID
   #MACRO START            ;Send start to manager.
     PEND $PDF FROM $PDD
     PRINTLN
     PRINTLN "%START" $ID
     PRINTLN
     POST 1 TO $PDD
   #ENDM

   #MACRO FINISH           ;Send finish to manager.
     PEND $PDF FROM $PDD
     PRINTLN
     PRINTLN "%FINISH" $ID
     PRINTLN
     POST 1 TO $PDD
   #ENDM

   #MACRO END           ;Send end to manager.
     PEND $PDF FROM $PDD
     PRINTLN
     PRINTLN "%END" $ID .1 .2
     PRINTLN
     POST 1 TO $PDD
   #ENDM

   #MACRO QC            ;Send quality control result to manager.
     PEND $PDF FROM $PDD
     PRINTLN
     PRINTLN "%QC" $ID .1 .2
     PRINTLN
     POST 1 TO $PDD
   #ENDM
```

```
#MACRO ERROR          ;Send robot error message to manager.
  PEND $PDF FROM $PDD
  PRINTLN
  PRINTLN "%ERROR " $ID .1 .2
  PRINT    .3
  PRINT    ".4 .5 .6 .7 .8 .9"
  PRINTLN
  POST 1 TO $PDD
#ENDM

;Controller manager protocol macros for local ID
#MACRO STARTL       ;Send start of local program to manager.
  PEND $PDF FROM $PDD
  PRINTLN
  PRINTLN "%START" $IDL
  PRINTLN
  POST 1 TO $PDD
#ENDM

#MACRO FINISHL      ;Send finish of local program to manager.
  PEND $PDF FROM $PDD
  PRINTLN
  PRINTLN "%FINISH" $IDL
  PRINTLN
  POST 1 TO $PDD
#ENDM

#MACRO ENDL         ;Send end of local program to manager.
  PEND $PDF FROM $PDD
  PRINTLN
  PRINTLN "%END" $IDL .1 .2
  PRINTLN
  POST 1 TO $PDD
#ENDM

#MACRO QCL          ;Send quality control result of local
                    ;program to manager.
  PEND $PDF FROM $PDD
  PRINTLN
  PRINTLN "%QC" $IDL .1 .2
  PRINTLN
  POST 1 TO $PDD
#ENDM

#MACRO ERRORL       ;Send robot error message of
                    ;local program to manager.
  PEND $PDF FROM $PDD
  PRINTLN
  PRINTLN "%ERROR " $IDL .1 .2
  PRINT    .3
  PRINT    ".4 .5 .6 .7 .8 .9"
  PRINTLN
  POST 1 TO $PDD
#ENDM

;Controller Other Devices protocol macros
#MACRO  STOP    ;Stop a device (like conveyor) or pcplc
process.
  STOP    GT.1
  STOP    PT.1
#ENDM

#MACRO CNCREQ   ;Request from VC2_CNC
  PEND $PDF FROM $PDD
  PRINTLN
  PRINTLN "%CNCREQ .1 .2 .3 .4 .5 .6 .7 .8 .9"
  PRINTLN
  POST 1 TO $PDD
#ENDM
```

```
  #MACRO CNCSTR      ;String to VC2_CNC
    PEND $PDF FROM $PDD
    PRINTLN
    PRINTLN "%CNCSTR .1 .2 .3 .4 .5 .6 .7 .8 .9"
    PRINTLN
    POST 1 TO $PDD
  #ENDM

#ENDIF
```

The file *WS1.DNL* contains sample ACL source code. This is the type of file you would edit when you want to change the way a robot moves.

ACL System Programs

The .\LIB\ACL directory contains the file ***CIMSYS.SYS***. This file includes system programs for:

Name	Identity (TP Run #)	Explanation
HOMES	1	Prepares the robot to work in the OpenCIM environment.
CIM	2	Reserved for future use.
RESET	3	Resets the variables and the programs.
CLEAR	4	Reserved for future use.
CIMP	5	Attaches the vectors CIM and CIMB to the teach pendant.
USER1	6	User-defined program.
USER2	7	User-defined program.
USER3	8	User-defined program.
USER4	9	User-defined program.
INIT	10	Reserved for future use.
OGRIP	11	Opens the gripper.
CGRIP	12	Closes the gripper.
DIAG	13	The ACL device driver runs this program when it receives an asterisk (*) from the controller (Diagnostic).
PCPLC	14	The ACL device driver runs this program in order to download OpenCIM parameters to the ACL controller (Pick-and-Place).
$REST	15	The system reset program.
AUTO	16	This program runs each time the device driver is loaded and the ACL controller is turned on.
INITC	17	This program is activated each time you load the ACL device driver.

Note

If you want to modify one of the ACL System program files, enter these changes in the unique PROLOGn.DNL file only.

ACL Variables

In the ACL controller there are three types of global variables and two types of local variables.

The global variables are:

- **ACL Controller System Variables**

Variable	Description
IN[16]	Input status
ENC[6]	Encoder
TIME	Time
LTA	Last time for group A
LTB	Last time for group B
MFLAG	Motion Bitmap
ERROR	Error
OUT[16]	Output status
ANOUT[6]	Direct analog current to axis

For more information refer to the ACL Reference Guide.

- **ACL Controller User Variables** (OpenCIM System Variables)

Variable	Description
ERRPR	Error program
ERRLI	Error line
$SYNC	Synchronization
$PDD	Pend/ post printing to ACL programs
$PDF	Pend/ post printing to ACL programs
SPFA	Speed fast group A
SPMA	Speed medium group A
SPSA	Speed slow group A
SPFB	Speed fast group B
SPMB	Speed medium group B
SPSB	Speed slow group B
$ID	ID message number from the ACL device driver
PART	Part number
PID	Part ID group
$DEV1	Get device number
$DEV2	Put device number
INDXG	Get device index
INDXP	Put device index
$NOTE	Note number
P1	Last position in the device group A
P2	Position before P1
P3	Position before P2
P4	Position before P3
P5	Position before P4
P6	Position before P5

Variable	Description
P7	Position before P6
P8	Position before P7
P9	Position before P8
P10	Position before P9
TB	Time needed for group B
NEWB	New group B position
LASTB	Last group B position
KB	Time constant variable for group B
$NEW	Temporary new position for group B
$LAST	Temporary last position for group B
PB1	Last position in the device group B
PB2	Position before PB1
PB3	Position before PB3

- **ACL Controller User Variables** (OpenCIM User Variables)

✆ Note : *From the ATS, type* LISTVAR *to display a list of variables.*

The local variables are:

- **OpenCIM Local System Variables** (ACL Controller Variables)

Variable	Description
$I	
$IDL	$ID Local

- **OpenCIM User-Defined Local Variables** (ACL Controller Variables)

If you want to modify one of the ACL System program files, enter these changes in the unique PROLOGn.DNL file only (e.g. If you want to change the SPFA, go to the EPILOGn.dnl file and type SETSPFA=80).

PROLOG (PROLOGn.DNL)

In the *PROLOGn.DNL* file you can modify system programs, such as Global System variables (e.g. SPSA - speed slow group A), predefined positions or any other settings that need to be predefined for a particular station.

The following examples show how the PROLOGn.DNL file can be modified:

Modify Operating System Programs

```
; ********* HOME ************
PROGRAM    HOMES /Y
* HOMING THE ROBOT
HOME
* HOMING THE L.S.B
HHOME 7
END
```

Define Positions

The principle here is to have fewer positions for better performance and faster backup/restore.

Definition of a CIM position vector: P= lower point, P+10= highest point.

In the file PROLOG.DNL, all the positions are as follows:

1. Divide all of the positions into groups of 10 (ten).
2. The template positions 1 through 20 are for conveyors and buffers (e.g. conveyor 1/11, first buffer 2/12, etc.).
3. Put=Get + 100
4. The last two spaces are used for OPENSPACE and OPENSPACE FOR TEMPLATE.

Example of how to define a position:

```
;********** POINTS **********
DIMP CIM[xxx]

; CIM:
;   1..9      P1          TEMPLATE
;  10..19     P2          TEMPLATE
;  21..29     P1          GET PART FROM BFFR1
;  31..39     P2          GET PART FROM BFFR1
;  41..49     P1          GET PART FROM
;  51..59     P2          GET PART FROM
;  61..69     P1
;  71..79     P2
;  81..89     P3
;  90..91
;  92..93
; 101..109    P1
; 111..119    P2
; 121..129    P1          PUT PART AT BFFR1
; 131..139    P2          PUT PART AT BFFR1
; 141..149    P1          PUT PART AT ASMBUF
; 151..159    P2          PUT PART AT ASMBUF
; 161..169    P2          OPENSPACE
; 171..179    P1          OPENSPACE FOR TEMPLATE
; 181..189    FREE
; 191..199    FREE
```

GET/PUT Programs

The GET/PUT.DNL files contain the pick-and-place programs.

GET/PUT Program Structure

The following table shows the sample GET and PUT programs found in the file *.DNL. The customization instructions explain how to add your own code to complete these programs.

Example Programs	Customization Instructions
`.PROGRAM_GET xxx`	⇒ Replace the *xxx* with the device ID as defined in DEVICE.DMC.
`; move robot`	⇒ Insert a command to quickly move the robot to a point above the source device (in the Free Movement Zone).
`; grab part/template`	⇒ Insert commands to lower the robot arm to the source device and grab the part/template. Use the part ID and index parameters (**PART**, **INDXG**) as needed to grab the object.
`; move robot`	⇒ Insert commands to move clear of the source device.
`.START`	⇒ This macro informs the CIM Manager that the source device is free.
`; move robot`	⇒ Insert commands to continue moving the robot to a point above the source device (in the Free Movement Zone) where the PUT program will take over.

Example Programs	Customization Instructions
`.END_GET`	⇒ This macro activates the PUT program.
`.PROGRAM_PUT` *xxx*	⇒ Replace the *xxx* with the device ID as defined in DEVICE.DMC.
`.SYNC`	⇒ This macro waits for the GET program to finish moving the robot that is carrying the part/template into position.
`; move robot`	⇒ Insert a command to quickly move the robot to a point above the target device (in the Free Movement Zone).
`; move robot`	⇒ Insert commands to lower the robot arm to the target device. Use the part ID and index parameters (**PART**, **INDXP**) as needed to set the part/template in its place.
`; deliver part/template`	⇒ Insert commands to move clear of the target device.
`.FINISH`	⇒ This macro informs the CIM Manager that the part is in place and the target device is ready to be activated.
`; move robot`	⇒ Insert commands to move the robot to its standard idle position (in the Free Movement Zone).
`.END`	⇒ This macro informs the CIM Manager that the robot is ready to perform the next GET operation.
`.END_PUT`	⇒

The following figure shows an example of GET and PUT programs after customization.

```
.PROGRAM_GET .ASRS
.FAST
MOVED   CIM[11]
.SLOW
.OPEN
MOVED   CIM[1]
.CLOSE
MOVED   CIM[11]
.START
.END_GET

.PROGRAM_PUT .ASRS
.SYNC
.FAST
MOVED   CIM[11]
.SLOW
MOVED   CIM[1]
.OPEN
MOVED   CIM[11]
.FINISH
.END
.END_PUT
```

Figure 11-4: Sample GET and PUT Programs After Customization

Synchronizing the GET and PUT

The ACL device driver activates the appropriate GET and PUT programs simultaneously. Since a robot must first pick up a part before it can place it, the GET program must execute before the PUT program. A synchronization mechanism is used to suspend the PUT program until the GET program is finished.

The synchronization mechanism is handled automatically with macros if you base your ACL programs on the sample GET and PUT programs found in CIMACL.DNL. The following sample code shows the synchronization code after macro expansion.

```
PROGRAM GT001 - Get Template from Conveyor
    ******************************************
    Set  $SYNC = 0.    ; causes PUT program to wait
    .
    .
    Set  $SYNC = 1     ; activates PUT program
    End


    PROGRAM  PT001 - Put Template in Buffer
    ****************************************
    Wait $SYNC = 1     ; wait for GET program to set activate flag
    Set  $SYNC = 0     ; reset the activate flag
    .
    .
```

Figure 11-5: Synchronization Example for GET & PUT (Object Code)

QC Programs

The ACL can only hold integers. If you want to test integer quality control, you can send your QC result to the ACL device driver (Barcode) once. If you want to test a decimal number you need to send it string by string.

Integer Quality Control

ACL Source Format*.QCL Format	ATS / ACL Controller *.CBU Format	

`.PROGRAM QC`
`.GETIDL`
	`PROGRAM QC`
	`**************`
	`LABEL  11`
	`READ  "?%ID?"`
	`       $IDL`
	`IF`
	`       $IDL = 0`
	`  GOTO        11`
	`ENDIF`
	`SET        $I = 1`
		1. Receives the "ID" message number from the ACL device driver.

. . .	. . .	2. Process to read the QC test.

ACL Source Format*.QCL Format	ATS / ACL Controller *.CBU Format	

ACL Source Format	ATS /ACL Controller	

`SET QCRES = xx`
`  .QCL QCRES`
 	
`END`
	`SET QCRES = xx`
	`        PEND`
	`        $PDF FROM $PDD`
	`        PRINTLN`
	`        PRINTLN`
	`        "%QC" $IDL`
	`QCRES`
	`        PRINTLN`
	`        POST`
	`        1 TO $PDD`
	`    ENDIF`
	`    END`
		3. This variable receives the QC result. You send the result to the ACL device drivers.

		4. I. The CIM Manager sends two values (a package), a higher limit and a lower limit. Each package has its own sequence # so that it can be identified by the ACL device driver. These are the values that you defined in the Part Definition form and in the field parameters. II. The ACL device driver sends the sequence # for the package via the RS232 link to the ACL controller. III.The ACL controller runs a QC test and receives a value. This value is sent back to the ACL device driver. IV.The ACL device driver verifies that the value it received is within the higher and lower limits of the original package (values) sent. If the value is within the limits, then the ACL device driver sends a message that the QC is OK. If the value is not within the limits then the ACL device driver sends a FAIL message.

Process Programs

The Process programs include all of the utility programs necessary to operate the ACL Input/Output (e.g. if you want to open the door of a CNC).

EPILOG (EPILOGn.DNL)

The EPILOGn.DNL file can be used to modify initialization system programs (INITC, RESET) in the following ways:

ACL Source Format EPILOGn.DNL	ATS /ACL Controller *.CBU Format

`PROGRAM        RESET /Y`	`PROGRAM RESET`
	`*********************`
	`GOSUB  $REST`

ACL Source Format EPILOGn.DNL	ATS /ACL Controller *.CBU Format
```GOSUB $REST GOSUB BCOFF STOP RVP DELAY 50 RUN RVP DELAY 50 GOSUB SEMER END PROGRAM INITC /Y STOP RVP DELAY 30 RUN RVP .STOP .CNV1 .STOP .ASRS .STOP .BFFR1 .STOP .FDR1 .STOP .RACK1 .STOP .RDR1 .STOP .TRASH1 .STOP .ASMBUF POST 1 TO $PDD GOSUB SEMER END```	```GOSUB BCOFF STOP RVP DELAY 50 RUN RVP DELAY 50 GOSUB SEMER END PROGRAM INITC ********************** STOP RVP DELAY 30 RUN RVP STOP GT001 STOP PT001 STOP GT003 STOP PT003 STOP GT005 STOP PT005 STOP GT006 STOP PT006 STOP GT008 STOP PT008 STOP GT009 STOP PT009 STOP GT007 STOP PT007 STOP GT004 STOP PT004 POST 1 TO $PDD GOSUB SEMER END```

ACL Off-Line Utilities

OpenCIM requires the use of the ACLoff-line utility program version 1.65 or later. The following ACL features are used when writing structured ACL code for OpenCIM:

- Symbolic constants (#DEFINE)
- User defined macros (#MACRO, #ENDM)
- Include files (#INCLUDE)
- Download Flags
- # IF
- # IFDEF
- # IFNDEF
- # ELSE
- # ENDIF
- Parameter passing using global variables.
- Synchronization between programs running simultaneously.
- Using the ACLoff-line utility program to send ACL source code to the ACL controller.

You should be familiar with the system information contained in the include files CIMSYS.SYS and CIMSYS.DMC before you start writing your own ACL programs for the OpenCIM environment. This file contains system macros, programs, and global variable definitions which perform the following functions:

- Synchronize the running of GET and PUT programs.
- Start and end GET and PUT programs.
- Error handling.
- Send status messages to the CIM Manager, to a CNC machine, and to other CIM entities

⚠️ **Caution**

You should NOT edit the CIMSYS.SYS or CIMSYS.DMC files without guidance from Intelitek technical support.

The program DOWNLOAD.EXE sends ACL source code files to the controller. While it is sending programs to the ACL controller, this downloader checks syntax and substitutes the program code associated with #DEFINE, #MACRO, and #INCLUDE. These three language directives function similarly to their counterparts in C or Assembly language. For more details, see the ACLoff-line program User's Manual and the ACL Reference Guide.

Note

You must run the DOWNLOAD utility program on the Station Manager PC that is connected to the controller in order to download programs (because the downloader requires an RS232 connection to the controller). However, the ACL source code files can reside on any PC accessible via the network. Before trying to download code with the DOWNLOAD utility, be sure to close the ACL device driver for this controller. Otherwise, a conflict will occur when the ACL program tries to open the same serial port as the device driver.

Robot Programs

The Robot Programs consist of all the relevant programs necessary to run the robot. These programs are divided into two categories: generic and unique, "generic" referring to the fact that the program can be used by all the robots and "unique" referring to programs that are specific for the robot at that station.

The Robot Programs are located in two places:

- Generic Windows XP: [default system drive (for example C:)]\Documents and Settings\All Users\Documents\Intelitek\OpenCIM\Projects\LIB\ACL

    Windows Vista/7: [default system drive (for example C:)]\Users\Public Documents\Intelitek\OpenCIM\Projects

- Unique \Project#\WS#\ROBOT#

The following tables describe the file name conventions used for the ACL system programs and sample applications supplied with OpenCIM. You should use these naming conventions in your own programming to simplify technical support. You can use the ACL off-line utility, the Robot Programs window or the text editor of your choice to edit these files.

We recommend that you use a Robot Programs window when editing these files. Save the files in ASCII format.

Tip

It is recommended that you add a Robot Programs program group on a Station Manager PC. This group should contain the following icons, as shown in the figure below.

- **ATS**: *Terminal emulation program which allows you to interact with the ACL controller connected to a Station Manager PC.*
- **Download**: *Sends ACL programs from a Station Manager PC to an ACL Controller.*
- **Download Report**:
- **Notepad WS1**: *Allows easy editing of a station's configuration file.*
- Additional **Notepads**: *for editing other .DNL files.*
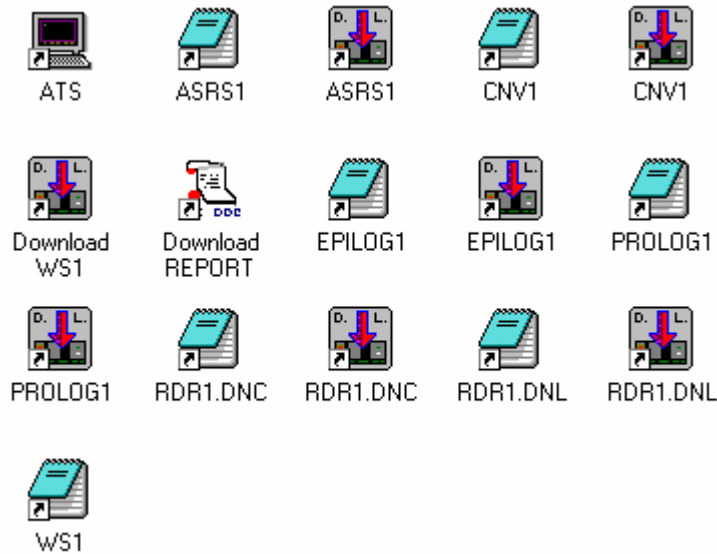
Figure 11-6: Robot Programs Group Window

Generic Robot Programs

File Extension	Meaning
*.DNB	**DowNload -** ACL source **Blocks** that are later copied to the DNL.
*.DMC	**Defines & MaCros -** An include file (library file) containing user-supplied defines and macros. The contents of this file are inserted in a DNL file at the point specified by an INCLUDE command.
*.SYS	All of the **SYStem** programs necessary to operate and communicate in the OpenCIM environment.
*.QCB	All the **Quality Control Block** programs and Communication Block programs necessary for peripheral devices that connect directly to the ACL controller.
*.PRB	All the **Process Block** programs necessary to **Communicate** with devices (e.g. CNC machine).
*.DLD	**DownLoad** utilities.

Unique Robot Programs

File Extension	Meaning
*.DNL	**DowNLoad -** *ACL source code* file that is sent to an ACL controller using the ACL Downloader.
*.QCL	This file is copied from a QCB file. After the original file is modified to the editing application, the file is then unique to the specific QC device.
*.PRL	This file is copied from a PRB file. After the original file is modified to the editing application, the file is then unique to the specific process activation (e.g. a program that speaks through I/O with a CNC machine).

**Editing and
Downloading a Robot
Program Separately**

1. In the Robot Programs window, select the Edit icon for the file you want to edit; the file is displayed.
2. Edit the file.

3. Select the Download icon for the file you just edited; the file is downloaded to the controller.

4. Select the Download Report icon . Verify that the file ends with `>>>>>END DOWNLOAD FILE! "Device file name"`

**Downloading All of
the Robot Programs at
One Time**

〰 Note

If any Robot program files need to be edited, edit them before downloading.

In the Robot Programs window, select the Download Station icon ; the files for the entire station are downloaded to the controller. When the files are downloaded together, they are sent in the following order:

1. CIMSYS.SYS
2. PROLOG*n*.DNL
3. DEVICES.DNL, DEVICES.QCL and DEVICES*.PCL
4. EPILOG*n*.DNL

You should not edit ACL object code by using the ATS utility or by backing up a program from a controller and modifying it. Either of these methods would result in an ACL file that is very difficult to maintain.

Editing ACL object code (i.e. an *.CBU file) from the controller results in a program which is out of sync with the original DNL file. This code is harder to read since all #DEFINE, #MACRO, and #INCLUDE statements have been expanded in the controller and all remarks have been deleted.

Note that the ATS utility can still be used for the following functions in the OpenCIM environment:

- Configure the controller
- Back up and restore robot positions and downloaded ACL programs
- Teach robot positions
- Test a robot
- Debug ACL programs by running them manually

Adding a New Pick-and-Place Operation

When you add a new device at a station (e.g. CNC machine, storage rack, assembly jig, etc.), you must write two new ACL programs (GT*xxx* and PT*xxx*) that enable a robot to pick up and deliver parts (or templates) from this device.

A program which directs a robot to pick up a part/template from a specific location is called GT*xxx*. The *xxx* represents the unique three-digit device ID for a source location that is found in the file SETUP.CIM.

Similarly, a program which directs a robot to deliver a part/template to a specific location is called PTxxx. Once again, xxx represents a unique device ID, this time for the target location.

For each pick-and-place operation, the CIM Manager sends a set of parameters to the ACL controller. The ACL program PCPLC reads these parameters from the ACL device driver and assigns them to a set of global variables. These variables are used to pass the parameters to the appropriate GET and PUT programs.

A GET program's main function is to pick up a part/template, which involves the following:

- Direct the robot to grasp a part/template
- Move the robot to a safe position; clear of the source device
- Send a Start status message
- Continue moving the robot to an intermediate point from which it can reach any device
- Activate the PUT program

A PUT program's main function is to place a part/template in a designated location. This operation involves the following steps:

- Wait for an activation signal from a GET program
- Move the robot to the target location
- Set the part/template down at the target location
- Move the robot clear of the target device
- Send a status message that the part/template is in place and ready to be processed
- Move the robot to safe position from which it can reach any device
- Send a status message that the robot is ready to perform the next operation

To write a set of GET and PUT programs for a new device that has been defined in SETUP.CIM, follow the steps outlined below:

Summary	Details
1. Add the device to the file DEVICE.DMC (connects the name of the device and the # of the device).	1. Use a text editor to insert a symbolic constant into the file DEVICE.DMC. For example: `#DEFINE ASRS    002` In this example, 002 is the device ID (3 digits required) and ASRS is the symbolic name you will use throughout your ACL programs to refer to this device. It is much easier to maintain ACL programs that use symbolic constants (e.g. `.ASRS`) instead of literal device IDs (002). If a device ID ever changes, it is only necessary to make the change in one place, i.e. in the file DEVICE.DMC. You should observe the following conventions when assigning device IDs in order to simplify technical support: 001 Pallet at this conveyor station 002 ASRS (or other central storage)
2. Copy the files from ..\LIB\ACL to ..WSn\ROBOTn	1. If you have a workstation with an ACL robot then add the directory ROBOTn. 2. Copy the files from ..\LIB\ACL as follows:

Summary	Details
	2.1. If the Type (FLD #5) from SETUP.CIM is: A, B, C, D, F, K, L, Q, J, M, S, X, Y, Z, then copy from: to: ..\LIB\ACL\WS.BLK ..\WSn\ROBOTn\WSn.DNL ..\LIB\ACL\PROLOG.BLK ..\WSn\ROBOTn\PROLOGn.DNL ..\LIB\ACL\EPILOG.BLK ..\WSn\ROBOTn\EPILOGn.DNL ..\LIB\ACL\CONFIG.DLD ..\WSn\ROBOTn\CONFIG.DLD Refer to "OpenCIM Setup File: SETUP.CIM" for more information. (Physical Name is FLD #3, Logical Name is FLD #4) **Rules for Creating DNL Files** 2.2 If the Object Type in the SETUP.CIM (FLD #5) is: A, B, C, D, F, K, L, J, Q, M, S, X, Y, Z, then copy from: to: ..\LIB\ACL\[FLD #3].DNB ..\WSn\ROBOTn\[FLD #4].DNL 2.3 If FLD #5 is: L, Q or Y, then copy from: to: ..\LIB\ACL\[FLD #3].QCB ..\WSn\ROBOTn\[FLD #4].QC 2.4 If FLD #5 is: D or M, then copy from: to: ..\LIB\ACL\[FLD #3].PRB to ..\WSn\ROBOTn\
3. Create two new icons: to edit the DNL and to download the DNL.	1. From the Robot Programs window, drag an icon; a new icon appears. 2. Select the new icon (click once) and press [Alt + Enter]; the Program Item Properties dialog box appears. 3. Type in the name of the new device in the Description field. 4. Type in the name of the new device in the command line for the Edit icon or type in the new Download file name for the Download icon. 5. Select OK.

Summary	Details

4. Editing the DNL, QCL and PRL files.	1. From the Robot Programs window, select the icon of the file you want to edit; the file is displayed.
	2. Type in the positions. Verify that the new positions you are entering do not already exist and that all the details of the position can be found in PROLOG.DNL.
	3. Set the accurate speed.
	
	Replace the *xxx* in the first line of the GET and PUT programs with the device name that was defined in DEVICE.DMC. Note the period preceding ".PROGRAM" and the space and the period preceding the device name (the leading period indicates a macro or symbolic constant). For example:
	
	`                Generic`
	`                Unique`
	`.PROGRAM_GET xxx   ➜    .PROGRAM_GET`
	`.ASRS`
	`.`
	`.`
	`.PROGRAM_PUT xxx   ➜    .PROGRAM_PUT`
	`.ASRS`

5. Use standard ACL procedures for teaching a robot to grab and release a part from the source and target locations.	1. Use a teach pendant (or other means) to train the robot how to move and grasp. 2. Enter the coordinates of the source and target locations in the appropriate array. 3. Determine how the robot should grip the part/template. 4. Fill in the appropriate robot movement commands in the GET and PUT programs. If the location has multiple compartments (or buffers), you can use the variables `INDXG` and `INDXP` to help calculate the array index for CIM[ ] (the array that contains all robot positions). These index parameters specify the cell location when dealing with a storage rack or ASRS.

When a robot wants to insert (or remove) a part in a CNC machine, it must tell the machine when to open its door and chuck so the robot can enter the machine. The GET and PUT programs associated with a CNC machine must communicate with CNC script programs that open and close the door and chuck on the machine.

Robot Errors

Crash

When the ACL controller detects that a robot has collided with something, it responds as follows:

- The controller goes into COFF mode (Controller Off) and immediately stops all motors on this robot.
- The controller immediately terminates all programs which are trying to move this robot. If any subsequent program attempts to move this robot, it will terminate with a run-time error.
- The ACL device driver automatically executes the DIAG program. (This program should is loaded at initialization time as part of CIMSYS.SYS.)
- The DIAG program sends a status message to the CIM Manager to reports the collision. See "Sample ACL Programs," for a listing of the DIAG program.

The CIM Manager displays the Device Error screen to alert the operator.

Emergency

Refer to your system user manual for details.

CNC Programming for OpenCIM

This section introduces the CNC script language and describes how to write CNC programs in the OpenCIM environment.

CNC Script Language

This section provides an overview of the CNC script language and describes various parameters, such as the initialization, parameters the system variables, CNC script error messages and so on.

Introduction

The CNC Script Language is a language which allows you to write programs to control a CNC machine. These programs can initiate machine operations by turning the machine's control lines on and off. Programs receive information from the machine via status lines. These control lines and status lines are connected to a station manager PC via an interface board. This board maps the control lines to output ports on the PC, and maps the status lines to input ports. Each line corresponds to a bit in an I/O port. CNC script programs communicate with a CNC machine by reading and writing these bits.

The CNC Script Interpreter is part of the CNC Device Driver. When the CNC Device Driver receives a system message containing a CNC command, it finds the corresponding program in the file CNC_SCR.DBF and runs it.

Language Overview

The following table lists the commands and parameter variables in the CNC Script Interpreter:

Code	Function
V1 – V16	Parameter variables read at initialization time.
P1 – P8	Parameter variables containing values passed to CNC script programs at run time.
BV0, BV1	Current value of the two output ports.
PORT0, PORT1	PC I/O port addresses mapped to CNC control and status lines.
SetBit( )	Modifies the bits of the specified output port.
Wait( )	Suspends program execution for a specified duration.
WaitBit( )	Suspends execution until the specified bit(s) are set to one or until the specified bit pattern appears on an input port.
WaitBitLow( )	Suspends execution until the specified bit(s) are set to zero or until the specified bit pattern appears on an input port.
PulsBit( )	Turns on the designated bits of an output port for a specified duration.
Draw( )	Prints a line to the screen.
Draw2( )	Prints two lines to the screen.
SendMsg( )	Sends a predefined message to another CIM entity.
DownLoadD( )	Sends a G-Code program to the CNC machine via RS232.
SendStr( )	Sends a string to the specified OpenCIM device (e.g. robot).
WaitStr( )	Suspends execution until the specified string arrives from the

Code	Function
	Open–CIM-CIM network.
WaitFile( )	Suspends execution until the specified file is created.
MSDOS( )	Performs any MS-DOS command.
MSWINDOWS( )	Launches any MS-WINDOWS executable file.
ABORT( )	Unconditionally aborts the current CNC device driver program.

Initialization Parameter Variables

The CNC Script Interpreter supports up to 16 general purpose parameter variables and two special purpose variables which are defined at initialization time. The general purpose variables contain values that are 0 to 80 bytes long. These variable names, *V1 - V16*, are fixed. The two special purpose variables, *PORT0* and *PORT1*, contain the addresses of I/O ports used to interface to the CNC machine.

The CNC Device Driver assigns the values of the parameter variables *V1-V16*, *PORT0*, and *PORT1* at initialization time. It reads these values from the section *[CNCDriverDefinitions]* in the file CNC.INI.

```
These variables are global to all CNC script programs.
Their values do not change during execution of a script
program.
```

General purpose variables can contain the following types of values:

Variable Type	Range	Example
Integer	Integers range in value from 0 - 2,147,483,647	V8 = 60000
String	A set of ASCII characters enclosed in quotation marks. A string can range in length from 0 - 80 characters.	V1 = "Door Open"
Bit Mask	A string of 8 ASCII text characters enclosed in quotation marks. Each character is either 0 or 1.	V16 = "01000111"

The following table describes the special purpose variables:

Variable	Description	Default Value
PORT0	The address of the first input and output ports on the PC used to communicate with a CNC machine.	0x500 (for both input and output ports)
PORT1	The address of the second PC input and output ports on the PC used to communicate with a CNC machine.	0x501 (for both input and output ports)

Passing Run-Time Parameters to Programs

Up to eight run-time parameters can be passed to a CNC script program. Each parameter contains a value that is 0 to 80 bytes long. The parameter names, *P1-P8* are fixed.

You specify a string of parameter values when you invoke a CNC script program. Parameters in this string are separated by commas. The first value in the string is assigned to the parameter variable *P1*, the second to *P2*, etc. The parameter string can be a maximum of 32 characters long (including the comma separators). The rules regarding parameter values described in the previous section also apply to run-time parameter values.

The following sample parameter string contains both numeric and string values:

```
1000,Please wait,60,Finished
```

The way in which you specify the parameter string depends on which of the following methods you are using to invoke the CNC script program:

Network Message	A CIM message sent to a CNC Device Driver contains two strings, the name of the CNC script program immediately followed by its parameter string.
CNC Control Panel	The string containing the list of parameters is specified in the Parameters window of the Control Panel.

The CNC Device Driver initializes the values of run-time parameters read from the file CNC.INI.

The values of run-time parameter variables do not change after they have been passed into a CNC script program.

System Variables

System variables BV*0* and BV*1* are used to read the current value of their respective output ports during execution of a script program. These variables are global to all CNC script programs for a device driver.

These system variables are assigned in the device driver's INI file in the section *[CNCDriverDefinitions]*. These 8-bit values, which range from 0–255 are assumed to be the initial state of the control lines of a CNC machine when the system is turned on.

The following table shows the CNC script system variables and their default values (i.e. the values used if no assignment appears in CNC.INI):

System Variable	Description	Default Value
BV0	The current status (Port Value) of output port # 0.	0
BV1	The current status (Port Value) of output port # 1.	0

Command Arguments and Syntax

Numeric command arguments can take one of the following forms:

Integer	`0 – 2147483647`
Parameter Variable	`V1 – V16, P1 – P8`

Spaces that appear in a command's argument list are ignored. Case is not significant in the spelling of command names or for variable names in the argument list. The following examples are equivalent:

```
SetBit(PORT1, BV1, &, V16)
SetBit (PORT1,BV1,&,V16)
```

Editing CNC Script Language Programs

All CNC script programs for a station (those that you write and those that come with the system) are stored in the script file CNC_SCR.DBF. More than one CNC device driver can share the same script file. This file normally resides either:

- On the server in a subdirectory designated for this station
- On the station manager PC running the CNC Device Driver

Use a dBASE editor to write your CNC programs to the appropriate CNC_SCR.DBF file. If a CNC device driver is running, you must close it before you begin editing its CNC_SCR.DBF file. Otherwise you will get an **Access Denied** error message.

Enter the name of each CNC script program in the *REQUEST* column. A program name can be up to 32 characters long. It can contain any combination of letters, numbers, spaces, and punctuation.

The *ACTION* column contains the CNC script commands that comprise a program. There is no limit on the number of commands contained in a program.

The *RETURN* column is reserved for internal use. Do not enter any values in this column.

This file uses the following format:

```
REQUEST         |  ACTION          |  RETURN  |
Program Name 1  |  script command 1 |         |
                |  script command 2 |         |
                |  script command 3 |         |
                |  script command 4 |         |
                |  script command 5 |         |
End             |                  |         |
Program Name 2  |  script command 1 |         |
                |  script command 2 |         |
                |       :          |         |
                        :
```

When the CIM Manager wants to run a CNC program, it issues a Run command to the appropriate CNC Device Driver at a station. The device driver finds the program in the file CNC_SCR.DBF and executes it.

```
Request              Action
Return

go in RS232 receive  pulsbit( port0, bv0, "00000001", 500 )
                     Draw("-- go to RS232 receive --")
end
go in auto           pulsbit( port0, bv0, "00000010", 500 )
                     draw("-- go in auto --")
end
start for 0001       pulsbit( port0, bv0, "00000100", 500 )
                     draw("-- start for 0001 --" )
                     draw2( v16, "-- machine is running -- " )
end                  waitbit( port0, "00000001", 10000 )
open door            pulsbit( port0, bv0, "00001000", 500 )
                     draw( "-- open door --" )
                     draw2( v16, "-- door is opened --" )
end                  waitbit( port0, "00000010", 10000 )
close door           pulsbit( port0, bv0, "00010000", 500 )
                     draw( "-- close door --" )
                     draw2( v16, "-- door is closed --" )
end                  waitbit( port0, "00000100", 10000 )
open clamping device pulsbit( port0, bv0, "00100000", 500 )
                     draw( "-- open clamping device -- " )
                     draw2( v16, "-- clamping device is opened" )
end                  waitbit( port0, "00001000", 10000 )
close clamping device pulsbit( port0, bv0, "01000000", 500 )
                     draw( "-- close clamping device --" )
                     draw2( v16, "-- clamping device is closed" )
end                  waitbit( port0, "00010000", 10000 )
feedhold             pulsbit( port0, bv0, "10000000", 500 )
                     draw2( "-- feedhold --", "*** ALARM ***" )
end                  waitbit( port0, "00100000", 10000 )
start for 0002       pulsbit( port1, bv0, "00000001", 500 )
                     draw( "-- start for 0002 --" )
                     draw2( v16, "-- machine is running --" )
end                  waitbit( port0, "00000001", 10000 )
pinole out           pulsbit( port1, bv0, "00000010", 500 )
                     draw( "-- pinole out --" )
                     draw2( v16, "-- pinole is out --" )
end                  waitbit( port0, "01000000", 10000 )
pinole in            pulsbit( port1, bv0, "00000100", 500 )
                     draw( "-- pinole in --" )
                     draw2( v16, "-- pinole is in --" )
end                  waitbit( port0, "10000000", 10000 )
```

Figure 11-7: Sample CNC Programs in the file CNC_SCR.DBF

CNC Script Error Messages

The error messages listed below appear in the CNC Status window when the CNC Script Interpreter encounters an invalid statement.

`"(" expected OR ")" expected`	• A parenthesis surrounding the command's argument list is missing.
`End of program not found`	• No End statement was found for the current program in the Request column of CNC_SCR.DBF.
`Invalid command name`	• The command name is not valid. Check the spelling.
`Program not found`	• The program name is not valid. Check the spelling.
`Invalid system variable - Use BV0 or BV1`	
`Invalid bit mask`	• The bit mask must be an 8-character string composed of 1s and 0s.
`Invalid port address - Use PORT0 or PORT1 parameter variable`	• Replace the invalid address with the variable PORT0 or PORT1. OR Check the value assigned to parameter variables PORT0 and PORT1 in CNC.INI.
`Unexpected number of arguments`	• There are either too few or too many values in the argument list for this command.
`Unexpected string in argument`	• An argument contains a string value instead of a numeric value.
`Unrecognized bitwise operator`	• A character other than &, |, ^, or ~ was specified as a bitwise operator.

CNC Script Language Commands

This section describes the CNC script language commands. Each of which is described in detail in the following sections.

DownloadD()

DownLoadD( )	Sends a G-Code file to the CNC machine via RS232	
Name:	DownLoadD(*FileName, MemArea*)	
Inputs:	*FileName* • Full DOS path of G-code program to be sent to the CNC machine	• 0 – 9999
	MemArea • Region in the CNC machine's memory where this program is to be loaded	• 1 – 5

Purpose

Normally a G-code file is assigned to a CNC process in the Machine Definition module. In this case, the CIM Manager takes care of automatically downloading this file prior to invoking the process and the *DownLoadD( )* command is not needed. You should only use the DownLoadD( ) command if you cannot set up a downloading batch file.

Function

The *DownLoadD( )* command sends the G-code file FileName to the CNC machine using the RS232 port specified in the device driver's command line. For machines capable of retaining more than one program, you can specify the memory region where the current program is to reside.

If the parameters *Loader* and *TaskLoadedMark* have been defined, the device driver uses the batch file specified by *Loader* to perform the download (recommended). Otherwise, the device driver's internal downloader is used.

Examples

```
DownLoadD(
Windows XP: [default system drive (for example
C:)]\Documents and Settings\All
Users\Documents\Intelitek\OpenCIM\ WSn\filename,
Windows Vista/7: [default system drive (for example
C:)]\Users\Public Documents\Intelitek\OpenCIM\WSn\filename,
memory area (optional))
```

This statement sends the file MILLPART.G to memory region 1 in the CNC machine.

```
DownLoadD(P1, P2)
```

When testing a machine, the following statement receives the G-code file and memory region that were defined in the Parameter field of the device driver's Control Panel.

Draw( )

Draw( )		*Prints a line to the screen*	
Name:	`Draw(line_of_text)`		
Inputs:	`line_of_text`	• String to display in the CNC Status window	• 0 - 80 characters

Purpose

Viewing status messages during the operation of a CNC machine can be helpful in troubleshooting problems and verifying the proper functioning of the machine. Messages can also instruct the operator of the machine when a manual procedure must be performed.

These messages can be particularly helpful when dealing with machines which have either no display of their own or only a very limited status panel.

Function

The *Draw( )* command prints the string `line_of_text` in the CNC Status window. This window appears on the PC running the CNC Device Driver. The *Draw( )* command prints each string on a new line.

The message string can be from 0 - 80 characters long. This argument can consist of either a string literal enclosed in quotation marks (e.g. "*Drilling in progress*") or a parameter variable (*V1 – V16, P1 – P8*).

Examples

```
Draw(P3)
Draw(V5)
Draw("Door Open")
```

Draw2( )

Draw2( )		*Prints 2 lines to the screen*	
Name:	`Draw2(text_line_1, text_line_2)`		
Inputs:	`text_line_1`	• First string to display in the CNC Status window	• 0 - 80 characters
	`text_line_2`	• Second string to display in the CNC Status window	• 0 - 80 characters

Purpose

When there is more than one line of text to display, it is convenient to use the *Draw2( )* command to print two lines of text with one command call. Using the *Draw2( )* command is also faster than making two successive calls to the *Draw( )* command. This can be an advantage when running in a busy, real-time environment.

Function

The *Draw2( )* command prints the strings `text_line_1` and `text_line_2` in the CNC Status window. This window appears on the PC running the CNC Device Driver. The *Draw2( )* command prints each string on a new line.

Each message string can be from 0 - 80 characters long. These arguments can consist of either a string literal enclosed in quotation marks (e.g. "*Machine overheated!*") or a parameter variable (*V1 – V16, P1 – P8*).

        Draw2(P3, V2)

        Draw2(V16, V5)

        Draw2("Part ready", P8)

PulsBit( )

PulsBit( )	*Turns on the designated bits of an output port for a specified duration*

Name:	**PulsBit**(*portn, mask1, mask2, time_to_puls*)		
Inputs:	*portn*	• Address of the output port to be modified	• 0x0000 – 0xFFFF
	mask1	• An 8-character string containing 1's and 0's representing a bit mask. All bits set to 1 are held high for the duration of *time_to_puls*.	• "00000000" – "11111111" V1 – V16 P1 – P16
	mask2	• An 8-character string containing 1's and 0's representing a bit mask. All bits set to 1 are held high for the duration of *time_to_puls*.	• "00000000" – "11111111" V1 – V16 P1 – P16
	time_to_puls	• Number of milliseconds to pause program execution while asserting the specified high bits	• CNCDriverTim er – 2147483 647 V1 – V16 P1 – P16

Purpose

Some operations of a CNC machine are time-based as opposed to operations that signal their completion via a status line. The *PulsBit( )* command provides a convenient way of executing an operation for a specified time period.

For example, after a part has been machined, it may be necessary to rinse it with water for 30 seconds before removing it from the CNC machine. The *PulsBit( )* command can be used to turn on the bit which controls the rinse cycle for the required period of time.

Function

The *PulsBit( )* command gives you the means to control CNC operations for a specified period of time. It turns on the designated bits of an output port for a specified duration.

The portn argument specifies the output port which contains the bit(s) that operate the control line(s) you are interested in.

mask1 can be set to either:

- The current value of the output port, BVn
- An absolute bit mask value (see description of *mask2* below)

You specify which ctrl lines should be pulsed on and off by constructing a bit mask, mask2. The mask2 argument is an 8-character string composed of 1's and 0's in ASCII text. This argument can consist of either a string literal enclosed in quotation marks (e.g. "*10000000*") or a parameter variable (*V1 – V16, P1 – P8*).

The `PulsBit( )` command performs a bitwise OR operation between `mask1` and `mask2.` It assigns the result to the output port for `time_to_puls` milliseconds. It then resets the port to its original value. The minimum TIME_TO_WAIT is the value of the parameter `CNCDriverTimer` found in CNC.INI.

The following table shows all the possible ways that `PulsBit( )` can affect a single bit position of an output port when mask1 = BVn:

PulsBit( )

Starting Value of an Output Port Bit (`mask1` = BVn)	Corresponding Bit in Bit Mask (`mask2`)	Control Line *During* Execution of `PulsBit( )`	Control Line *After* Execution of `PulsBit( )`
0	0	0 - Off	0 - Off
0	1	1 - On	0 - Off
1	0	1 - On	1 - On
1	1	1 - On	1 - On

Examples

```
PulsBit(PORT0, BV0, "10000000", 30000)
```

This example turns on control line # 7 that is connected to the PC's output port 0 for 30 seconds. The remaining bits in the port retain their current values during and after execution of `PulsBit( )` since their values in the bit mask are 0.

```
PulsBit(PORT1, BV1, V5, 100000)
```

The bits in output port 1 are ORed with the bit mask in variable `V5`. The result is written to output port 1 and the corresponding control lines are set on and off for 100 seconds. Afterwards, the original value of output port 1 is restored.

```
PulsBit(PORT1, BV1, P4, P7)
```

The bits in output port 1 are ORed with the bit mask in parameter variable `P4`. The result is written back to output port 1 and the corresponding control lines are set on and off for a period of `P7` milliseconds. Afterwards, the original value of output port 1 is restored.

SendMsg( )

SendMsg( )	*Sends a predefined message to another CIM entity*	
Name:	`SendMsg(`*`msg_to_send`*`)`	
Inputs:	*`msg_to_send`*	• Index to predefined messages stored in VC2_WM.DBF • `0 - 9999`

Purpose

In a CIM environment, a CNC machine must report its status to other CIM entities such as:

- The CIM Manager which handles production scheduling and tracking
- Devices which are dependent on this machine (e.g. the robot that tends the machine)
- The Graphic Production Module which displays the machine's status

The *SendMsg( )* command informs these entities when the CNC machine has completed processing a part, when there is a problem that causes an alarm condition, etc. This command uses a set of predefined messages to perform this function. Each message has an associated ID and destination device address.

You can generate real-time status messages by inserting the *SendMsg( )* command throughout a program.

You can add your own custom messages to the message file. For example, this capability is useful if you are programming a robot to tend this CNC machine. You could define a message to notify the robot when the machine is ready to receive a part and another message to inform the robot that the part is ready to be picked up. You would use the *SendMsg( )* command to send these messages.

Function

The `SendMsg(  )` command sends predefined real-time status messages to any CIM entity. The argument `msg_to_send` specifies the ID number of a message stored in the file VC2_WM.DBF. This ID number is sufficient to deliver the message because there is a destination address associated with each message in this file.

Examples

```
SendMsg(2582)
```

This statement finds the message with the ID number of 2582 in the file VC2_WM.DBF. It then sends this message to the destination device listed in this message record.

```
SendMsg(P4)
SendMsg(V12)
```

SetBit( )

SetBit( )	*Modifies the bits of the specified output port*	
Name:	`SetBit(`*`portn, mask1, bo, mask2`*`)`	
Inputs:	*`portn`*	• Address of the output port to be modified	`0x0000 - 0xFFFF`

	`mask1`	• An 8-character string containing 1's and 0's representing a bit mask (typically the current value of output *`portn`*, BVn)	`"00000000" - "11111111"`
			 `V1 - V16`
			 `P1 - P16`

	`bo`	• A single character designating the bitwise operation to be performed	`& - AND`
			 `| - OR`
			 `^ - XOR`
			 `~ - NOT`

	`mask2`	• An 8-character string containing 1's and 0's representing a bit mask	`"00000000" - "11111111"`
			 `V1 - V16`
			 `P1 - P16`

Purpose

The control lines of CNC machine are commonly mapped to bits in a PC's output port(s). When a bit is toggled on and off, it controls the corresponding function on the CNC machine. For example, suppose bit 3 is mapped to the door on the CNC machine. Setting bit 3 to one would close the door and setting it to zero would open the door.

By setting the appropriate bit(s) to the desired value, the *`SetBit( )`* command allows you to control a CNC machine.

Function

The *`SetBit( )`* command allows you to modify a set of bits in an output port in order to control the operation of a CNC machine.

The `portn` argument specifies the output port which contains the bit(s) that operate the control line(s) you are interested in setting. Each I/O port on a PC contains 8 bits.

`mask1` can be set to either:

```
 The current value of the output port, BVn
```
```
 An absolute bit mask value (see description of mask2 below)
```

Typically, `mask1` would be set to the system variable BVn, the current value of the output port. Using BVn allows you to set only the bits you are interested in while preserving the values of the rest.

Alternatively, you could set the port to an absolute value by specifying a bit mask for `mask1` instead of BVn. For example, to reset the port to a known value, you could specify the same bit mask value for `mask1` and `mask2` and use an OR operation between them. This would assign the value of the bit masks to the output port regardless of the port's previous value.

You can set the value of any group of bits in an output port by using the appropriate bit mask, mask2, and bitwise operator, bo. The mask2 argument is an 8-character string composed of 1's and 0's in ASCII text. This argument can consist of either a string literal enclosed in quotation marks (e.g. "*10000000*"), or a parameter variable (*V1 – V16*, *P1 – P8*).

The following truth tables show the results of using each of the four C Language bitwise operators available:

& - AND

Bit in *mask1*	Corresponding Bit in *mask2*	Result after SetBit( )
0	0	0
0	1	0
1	0	0
1	1	1

| - OR

Bit in *mask1*	Corresponding Bit in *mask2*	Result after SetBit( )
0	0	0
0	1	1
1	0	1
1	1	1

^ - XOR

Bit in *mask1*	Corresponding Bit in *mask2*	Result after SetBit( )
0	0	0
0	1	1
1	0	1
1	1	0

~ - NOT

Bit in *mask1*	Corresponding Bit in *mask2*	Result after SetBit( )
0	Not used	1
1	Not used	0

Examples

```
SetBit(PORT0, BV0, |, "10000000")
```

This example turns on control line # 7. This control line is connected to the PC's output port 0. The OR truth table indicates that when the mask contains a 1 in a given position, that bit will be set to 1 regardless of the bit's initial value in the output port.

```
SetBit(PORT1, BV1, &, V16)
```

The bits in output port 1 are ANDed with the bit mask in variable *V16*. The result is written back to output port 1 and the corresponding control lines are set on and off accordingly.

```
SetBit(PORT1, BV1, ^, P8)
```

The bits in output port 1 are XORed with the bit mask in parameter variable *P8*. The result is written back to output port 1 and the corresponding control lines are set on and off accordingly.

Wait( )

Wait( )	*Suspends program execution for a specified duration*	
Name:	**Wait**(*time_to_wait*)	
Inputs:	*time_to_wait* • Number of milliseconds to suspend program execution	• CNCDriverTimer - 2147483647, V1 - V16, P1 - P16

Purpose

After performing an operation on a CNC machine, it is sometimes desirable to pause for a while before continuing with the next operation. For example, it may be necessary to wait 2 minutes after a certain procedure to allow a part to cool down before a robot extracts it from the CNC machine. In this case the command *Wait(120000)* can provide the required delay before the CNC script program signals the CIM Manager that the part is ready.

Function

When the CNC Script Interpreter encounters the *Wait( )* command, it pauses for the indicated amount of time before executing the next program statement. The argument `time_to_wait` specifies the number of milliseconds the interpreter waits before resuming execution. This argument can be either an integer or a parameter variable (*V1 - V16*, *P1 - P16*). The minimum `time_to_wait` is the value of the parameter *CNCDriverTimer* found in CNC.INI.

Examples

```
Wait(2000)
```

This statement causes the CNC Script Interpreter to pause for 2 seconds.

```
Wait(V16)
```

```
Wait(P8)
```

The length of the pause resulting from each of these two statements depends on the values of variables *V16* and *P8.*

WaitBit( )

WaitBit( )	*Suspends execution until the specified bit(s) are set to one or until the specified bit pattern appears on an input port*

Name:	**WaitBit**(*portn*, *mask*, *time_to_wait*)		
Inputs:	*portn*	• Address of the output port to be modified	• 0x0000 – 0xFFFF
	mask	• An 8-character string containing 1's and 0's representing a bit mask	• "00000000" – "11111111", V1 – V16, P1 – P16
	time_to_wait	• Maximum number of milliseconds to suspend program execution while waiting for a bit pattern	• CNCDriverTimer – 2147483647, V1 – V16, P1 – P16

Purpose

After performing an operation on a CNC machine, it is frequently necessary to wait for a status line to signal that the operation was successfully performed. Status lines from a CNC machine are connected to an I/O board which maps each line to a bit in a PC input port.

By examining the value of these input port bits, it is possible to determine information about a machine such as:

Status Line	Example
An operation is completed.	Bit 0 Drilling in progress = 0 Hole drilled = 1
An alarm condition has occurred.	Bit 1 Normal temperature = 0 Machine overheated = 1
The position of a component on a CNC machine.	Bit 2 Door open = 0 Door closed = 1

The *WaitBit( )* command allows you to read a set of status lines in order to monitor the operation of a CNC machine.

Function

The WaitBit( ) command monitors an input port, `portn`. It compares the value of this port with the bit mask argument, `mask`. This command waits for the specified interval for one of the following conditions to be true:

- An exact match occurs between all 8 bits of the input port and the bit mask.

- A bit equal to 1 in *mask* matches up with the corresponding bit equal to 1 in the output port. For example, bits 7 and 5 below meet this condition:

- 10100000 ⇐ *mask*
  11110000 ⇐ BVn

When one of these conditions is true, the Command Interpreter prints in the Status window:

```
--- Condition is true ---
```

It then proceeds to execute the next command.

The *WaitBit( )* command monitors the port for the period of time specified in the argument `time_to_wait` (in milliseconds). This argument can be either an integer or a parameter variable (*V1 - V16, P1 - P16*). The minimum `time_to_wait` is the value of the parameter *CNCDriverTimer* found in CNC.INI.

If neither of the above conditions occurs during this interval, the command times out. The CNC Script Interpreter aborts program execution and generates the error message WM_CIMDDE_CNCERROR. The CNC Device Driver relays this error message back to the CIM Manager.

Examples

```
WaitBit(PORT1, "1000001", 2000)
```

This statement causes the CNC Script Interpreter to check port 1 for up to 2 seconds. If status lines 0 and 7 in this port are high, the condition is true and execution resumes with the next CLINT statement. If this match does not occur within 2 seconds, an error is generated.

```
WaitBit(PORT0, V3, P2)
```

The variables *PORT0* and *V3* contain the address of the input port and the bit mask respectively. Variable *V3* is assigned its value at initialization time from the file VC2_CNC.INI. The time out interval, *P2*, is specified at run time when this program is invoked.

WaitBitZ( )

WaitBitZ		*Suspends execution until the specified bit(s) are set to zero*	
Name:		*WaitBitZ(*portn, mask, time_to_wait*)*	
Inputs:	*portn*	• Address of the output port to be modified	• 0x0000 – 0xFFFF
	mask	• An 8-character string containing 1's and 0's representing a bit mask	• "00000000" – "11111111", V1 – V16, P1 – P16
	time_to_wait	• Maximum number of milliseconds to suspend program execution while waiting for a bit pattern	• CNCDriverTimer – 2147483647, V1 – V16, P1 – P16

Purpose

The *WaitBitZ( )* command allows you to read a set of status lines in order to monitor the operation of a CNC machine. See the *WaitBit( )* command above for details.

Function

The WaitBitZ( ) command monitors an input port, portn. It compares the value of this port with the bit mask argument, mask. This command waits for the specified interval for one of the following conditions to be true:

- An exact match occurs between all 8 bits of the input port and the bit mask.

A bit equal to 1 in mask matches up with the corresponding bit equal to 0 in the output port. For example, bits 5 and 7 below meet this condition:

- 10100000 ⇐ *mask*
  01011111 ⇐ BVn

When one of these conditions is true, the Command Interpreter prints in the Status window:

 --- **Condition is true ---**

It then proceeds to execute the next command.

The *WaitBitZ( )* command monitors the port for the period of time specified in the argument time_to_wait (in milliseconds). This argument can be either an integer or a parameter variable (*V1 – V16, P1 – P16*). The minimum time_to_wait is the value of the parameter *CNCDriverTimer* found in CNCVDn.INI.

If neither of the above conditions occurs during this interval, the command times out. The CNC Script Interpreter aborts program execution and generates the error message WM_CIMDDE_CNCERROR. The CNC Device Driver relays this error message back to the CIM Manager.

```
 WaitBitZ(PORT1, "10000001", 2000)
```

This statement causes the CNC Script Interpreter to check port 1 for up to 2 seconds. If status lines 0 and 7 in this port are low, the condition is true and execution resumes with the next CLINT statement. If this match does not occur within 2 seconds, an error is generated.

```
 WaitBitZ(PORT0, V3, P2)
```

The variables *PORT0* and *V3* contain the address of the input port and the bit mask respectively. Variable *V3* is assigned its value at initialization time from the file CNCVDn.INI. The time out interval, *P2*, is specified at run time when this program is invoked.

WaitFile( )

WaitFile( )		*Suspends execution until the specified file is created*	
Name:	WaitFile(*String, time_to_wait*)		
Inputs:	*String*	• Any valid MS-DOS file name.	• Any valid MS-DOS file name.
	time_to_wait	• Number of milliseconds to suspend program execution	• CNCDriverTimer - 2147483 647, V1 - V16, P1 - P16

Purpose

The command is generally used to get a status message in order to continue execution of the process after uploading a G-code program to a CNC machine.

Function

The *WaitFile( )* command waits until the specified file is created during a specified interval. If the specified file has not been created during the specified interval, the command times out. The CNC Script Interpreter aborts program execution and generates the error message WM_CIMDDE_CNCERROR. The CNC Device Driver relays this error message back to the CIM Manager.

Examples

```
 WaitFile(P1, V1)
 WaitFile(V1, P1)
 WaitFile(V1, V2)
 WaitFile(P1, P2)
 WaitFile("
 Windows XP: [default system drive (for example
 C:)]\Documents and Settings\All
 Users\Documents\Intelitek\OpenCIM\ WSn\filename
 Windows Vista/7: [default system drive (for example
 C:)]\Users\Public
 Documents\Intelitek\OpenCIM\WSn\filename"), wait time)
```

WaitString( )

WaitString( )		*Suspends execution until the specified string arrives from Network*	
Name:	WaitString(*String, time_to_wait*)		
Inputs:	*String*	• A string that specifies the beginning or end of an operation.	• Any string that contains an address in the OpenCIM environment.
	time_to_wait	• Number of milliseconds to suspend program execution	• CNCDriverTimer – 2147483647, V1 – V16, P1 – P16

Purpose

After performing an operation on a CNC machine, it is usually necessary to wait for a status message in order to confirm that the operation was performed successfully.

Function

The *WaitString( )* command waits for a string during a specified interval. If the CNC device driver doesn't receive the string during the specified interval, the command times out. The CNC Script Interpreter aborts program execution and generates the error message WM_CIMDDE_CNCERROR. The CNC Device Driver relays this error message back to the CIM Manager.

Examples

```
WaitString(V1, V2)
WaitString(V1, P1)
WaitString(P1, V1)
WaitString("Test is done", 10000)
```

SendString( )

SendString( )	*Sends a string to the specified OpenCIM device (e.g. to a robot)*	
Name:	SendString(*Address, String*)	
Inputs:	*Address* • Name of the workstation in the OpenCIM environment.	• Any of the OpenCIM predefined commands.
	String • String to be sent to be sent to the defined address	• Any string that contains an address in the OpenCIM environment.

Purpose

When a program is to be run in the ACL controller, a string (recognizable by the ACL controller) is sent to the ACL controller and the address tells the controller where to send the string.

Function

The *SendString( )* command sends predefined commands to any CIM entity.

Examples

```
SendString(P1, P2)
SendString(V1, V2)
SendString(V1, P1)
SendString("WS0 DDE ACL 25", "RUN SSCNC")
```

MSDOS( )

MSDOS( )	*Performs any MS-DOS command*	
Name:	MSDOS(*String*)	
Inputs:	*String* • Any valid MS-DOS command.	• Any valid MS-DOS command.

Purpose

The command provides an interface from OpenCIM to the MS-DOS operating system.

Function

The *MSDOS( )* command launches the MS-DOS command interpreter with the string specified at the function's input.

Examples

```
MSDOS(V1)
MSDOS(P1)
MSDOS("LOADER.BAT FILE.DAT")
```

MSWINDOWS( )

MSWINDOWS( )	*Launches any MS-WINDOWS executable file*	
Name:	MSWINDOWS(*String*)	
Inputs:	*String* • Any valid MS-Windows command line.	• Any valid MS-Windows command line.

Purpose

The command provides an interface from OpenCIM to the MS-WINDOWS operating system.

Function

The *MSWINDOWS( )* command launches the MS-WINDOWS executable file according to the string specified at the function's input.

Examples

```
MSWINDOWS(V1)

MSWINDOWS(P1)

MSWINDOWS("[default system drive (for example
C:)]\WINDOWS\NOTEPAD.EXE

Windows XP: [default system drive (for example
C:)]\Documents and Settings\All
Users\Documents\Intelitek\OpenCIM\ WSn\filename

Windows Vista/7: [default system drive (for example
C:)]\Users\Public
Documents\Intelitek\OpenCIM\WSn\filename")
```

ABORT( )

ABORT( )	*Unconditionally aborts the current CNC Device Driver program*
Name:	ABORT( )

Purpose

Using this command is the only way to abort a CNC Device Driver program without the CIM Manager.

Function

The *ABORT( )* command unconditionally aborts the current CNC Device Driver program.

Examples

```
ABORT( )
```

Robot and CNC Interface

When a robot inserts a part into a CNC machine, it must coordinate its movements with the operation of the machine using a CNC synchronization mechanism. This mechanism is used when a robot sends a command message to a CNC machine telling it to perform a certain function. The robot then waits for a response. Only after the CNC responds does the robot's ACL program continue execution.

The following scenario describes how a typical robot and CNC machine interact when the robot inserts and removes a part from the machine:

- The robot waits while the CNC machine opens its door and vise.
- The robot enters the machine. It holds the part in place while the machine clamps the part in its vise.
- The robot exits the machine and signals the CIM Manager that the CNC machine is ready to be activated.
- The robot waits outside the machine until it receives a signal from the CIM Manager that the machine is finished.
- The CNC opens its door and waits until the robot has grasped the part before it opens its vise.
- The robot removes the part and takes it to the next location.

While the CIM Manager could conceivably coordinate the above interaction between a robot and a CNC machine, it is more efficient for the ACL controller to do this. This section describes how to write ACL programs that communicate with script programs found in the CNC device driver. (You should already be familiar with OpenCIM ACL programming before continuing. See ACL Programming for OpenCIM for more information.)

The CNC synchronization mechanism is actually a specific case of how you can send command and status messages to perform OpenCIM operations.

It is possible to have two robots attached to a single ACL controller. In this case, you would use the ACL system file, CIMSYSM instead of CIMSYS. You would also need to adjust the sample code presented in this section to distinguish between the two robots. These changes are beyond the scope of this discussion.

The following diagram illustrates the sequence of commands that are executed when a robot inserts a part into a CNC machine. The commands shown in step 2 below are generated by the robot's PUT program for the CNC machine.

Command Sequence:

①. When the production plan calls for machining a part, the CIM Manager sends a single command telling the tending robot to insert the part in the CNC machine.

②. The PUT program in the robot's ACL controller sends a series of commands to open and close the door and the vise on the CNC machine so the robot can insert the part.

③. When the robot signals that the part is in place, the CIM Manager sends an activate command to the CNC machine.

Figure 11-8: Sequence of Commands when a Robot Tends a CNC Machine

The following table describes the dialog that occurs between a robot and its CNC machine. This table shows the sequence of CNC commands generated by the PUT and GET programs that insert and remove a part from a machine.

Robot Action	Robot Request	CNC Response
PUT Part into CNC Machine		
Robot brings part to entrance of CNC machine and waits for door to open.	**Open Door**	*Door Open*
Robot enters CNC machine with part and waits for vise to open.	**Open Vise**	*Vise Open*
Robot places part in vise and waits for vise to close.	**Close Vise**	*Vise Closed*
Robot releases part, withdraws from CNC, and waits for door to close.	**Close Door**	*Door Closed*
Robot signals CIM Manager that part is ready to be machined. The CIM Manager turns on the CNC machine.		
(The robot does NOT directly turn on the CNC machine by sending an activate command. Instead, the CIM Manager sends the activate CNC command message to the machine so that it can take care of downloading the G-code needed to process this part.)		
GET Part from CNC Machine		
Robot moves to entrance of CNC and waits for door to open.	**Open Door**	*Door Open*
Robot enters CNC, grasps part, and waits for vise to open.	**Open Vise**	*Vise Open*
Robot removes part from CNC machine.		

Writing Scorbase Programs for the CNC Machine

This section describes briefly how to write Scorbase programs for the CNC machine and displays various examples. It includes the following:

- Receiving Strings from an OpenCIM Device Driver
- Receiving Notifications from the CNC Machine and Sending Messages to the CNC Device Driver

Receiving Strings from an OpenCIM Device Driver

The Scorbase software automatically recognizes and executes the following two string formats, for example:

- **SET PRGNM=301:** In this string format, PRGNM is the name of the Scorbase variable and this statement is equivalent to the Set Variable PRGNM = 301 instruction in the Scorbase program. This string is generally defined in the **Program** column of the CIM Machine Definition utility in the CIM Manager application.

- **Run STRTM:** In this string format, STRTM is the Scorbase subroutine name and the string is equivalent to the Call STRTM instruction in the Scorbase program. The Scorbase program should contain the STRTM subroutine, as shown in the following example:

```
Set Subroutine STRTM

        Print to Screen: STRTM.PRGNM = 'PRGNM'

        Call Subroutine SCRIPT.OPENPORT (2)

        Call Subroutine SCRIPT.SENDFILE(2, PRGNM)

        Call Subroutine SCRIPT.CLOSEPORT(2)

Return from Subroutine
```

The STRTM subroutine is executed in parallel to the Pick and Place operation of the robot

Notes

The STRTM subroutine does not contain axis control Scorbase commands.

The Run STRTM string is generally defined in the CNC device driver script file.

Receiving Notifications from the CNC Machine and Sending Messages to the CNC Device Driver

The USB controller's digital input can be connected to the CNC machine's digital output. The CNC machine's G-Code program may change the status of the digital output that is connected to USB controller.

The Scorbase program can recognize changes to the status of the digital input and then run a user defined subroutine (END_LATHE23), which can be executed in parallel to the Pick and Place operation of the robot, as shown in the following example:

Set Subroutine PUT023

      Print to Screen: PUT PART TO CNC MACHINE CHUCK (CNC22)

          ...

          ...

          ...

      Enable Input Interrupt 1

      On Input Interrupt 1 On Run Subroutine END_LATHE23

      Send Message $Finish to MANAGER ID TASK_ID

    Send Message $End to MANAGER ID TASK_ID

Return from Subroutine

Set Subroutine END_LATHE23

      Send Message ENDLATHE to Device Driver ID 23

      Disable Input Interrupt 1

Return from Subroutine

*Pay attention to the **Enable/Disable Input Interrupt** instructions to ensure system safety.*

Note

Writing ACL Programs for the CNC Machine

The CNC synchronization mechanism is used when a robot must order a CNC machine to perform an operation and wait until the machine signals that it has completed the operation. The previous table shows the sequence of events in a set of PUT and GET programs that communicate with a CNC machine. The sample ACL code shown in this section demonstrates how to write these programs and implement the CNC synchronization mechanism.

You must code this synchronization mechanism into each of the following programs. Note the order in which programs are activated as shown in the figure below.

ACL Controller	The PUT and GET programs which tell the robot how to insert and remove parts from a CNC machine.
CNC Device Driver	CNC script programs (executed by the CNC device driver) which are activated by a robot request. For example: OPEN DOOR, CLOSE DOOR, OPEN VISE, CLOSE VISE
ACL Controller	A group of short ACL programs which set a semaphore variable that announces that the CNC machine has completed the requested operation (e.g. DROPN - door open, DRCLS - door closed, VCOPN - vise open, VCCLS - vise closed).
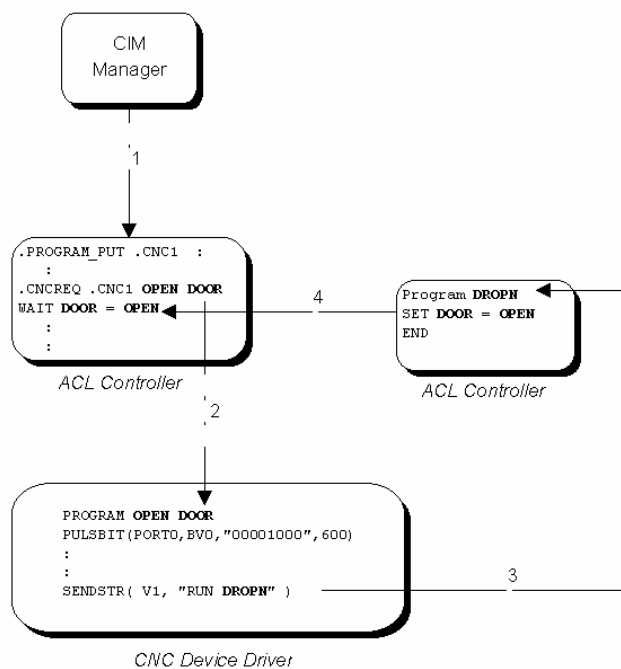
Figure 11-9: Sequence of Program Activation During CNC Synchronization

Sequence of Program Activation

1. `RUN GTxxx` `RUN PT001`	Pick-and-place command from the CIM Manager runs the ACL program PT001 to insert a part into the CNC machine. (The constant .CNC1 equals 001, the device ID of the machine.)

2. `.CNCREQ`
   `.CNC1 OPEN`
   `DOOR`
	Activates CNC script program OPEN DOOR in CNC device driver #1.

3. SENDSTR(V1, "RUN DROPN")	Sends an acknowledgment by running the ACL program DROPN. This line sends a run command to the ACL device driver whose address is specified in variable V1.
4. SET DOOR = OPEN	Causes the program PT001 to resume.

The following ACL statement sends a command to a CNC machine:

```
.CNCREQ    .CNC1 OPEN DOOR
```

The macro and symbolic constant in this statement expand to:

```
PRINTLN "%CNCREQ 001 OPEN DOOR   "
```

Each item in this statement is explained below:

ACL Statement that Requests a CNC Operation

PRINTLN	This ACL command sends a message to the robot's ACL device driver via a serial connection between the ACL controller and the Station Manager PC.
%CNCREQ	This string signals the ACL device driver that the rest of this message is a command intended for a CNC machine. The ACL device driver interprets and formats the message accordingly.
001	The device ID of the CNC machine that the robot is tending; used as part of the destination address.
OPEN DOOR	This string is the name of the script program which the CNC device driver will execute. The robot waits until this script program sends a response message indicating that it has finished performing the requested CNC operation. You can check the name of CNC script programs by looking at the Command List on the CNC Control Panel.

When an ACL program needs to activate an operation on another device and wait for an acknowledgment, it can use the CNC synchronization mechanism. The sample ACL code below demonstrates an efficient way to implement this mechanism:

```
          :
          :
SET DOOR = 0          The global variable DOOR represents the status of the door on
                      the CNC machine. It can have one of the following values:

                         0 - Door status is about to change. CNC request is pending.
                         OPEN - Door is open.
                         CLOSE - Door is closed.

                      The variable DOOR is used as a semaphore to signal that the
                      CNC machine has completed the requested operation (i.e. when
                      a response has been received from the CNC machine). This
                      variable is reset to 0 here to indicate that a CNC request is
                      pending.

.CNCREQ .CNC1 OPEN DOOR

                      This line uses an ACL macro (explained above) to send the
                      command message "OPEN DOOR" to CNC machine #1.

WAIT DOOR = OPEN      This line suspends execution of this ACL program until an
                      acknowledgment is received from the CNC machine.
                      (Note that this line does NOT assign the value of OPEN to the
          :           variable DOOR. The string DOOR = OPEN is a logical
          :           condition which must be satisfied before the WAIT statement
                      allows execution to continue.)
```

In this example, the CNC machine signals that the door is open by sending a command to the ACL device driver to run a short ACL program, DROPN. This program sets the global variable DOOR equal to OPEN. This assignment satisfies the WAIT condition and execution continues.

☞
Tip

The WAIT statement is more efficient than using a polling loop to continually check the value of the variable DOOR. Unlike a loop, this statement suspends execution of the program until the condition is satisfied. Suspending execution allows other active ACL programs to run faster.

The following is a set of sample ACL programs: GET, PUT, DROPN, DRCLS, VCOPN, and VCCLS. These programs insert and remove a part from a machine using the CNC synchronization mechanism (the CNC synchronization code is highlighted):

```
.PROGRAM_GET .CNC1
.OPEN
.FAST
MOVED     CIM[260]
MOVED     CIMB[260]
MOVED     CIM[260]
JAW 50
        SET        DOOR = 0
        .CNCREQ    .CNC1 OPEN DOOR
        WAIT   DOOR = OPEN
.MEDIOM
MOVED  CIM[261]
SPLINE CIM 262 263
.MEDIOM
MOVED     CIM[288]
.SLOW
MOVED     CIM[289]
.CLOSE
        SET        VISE = 0
        .CNCREQ    .CNC1 OPEN VISE
        WAIT       VISE = OPEN
.MEDIOM
MOVED     CIM[288]
.MEDIOM
MOVED     CIM[262]
.FAST
.START
MOVED     CIM[261]
MOVED     CIM[260]
.END_GET
```

```
PROGRAM   DROPN
SET       DOOR  = OPEN
END
```

```
PROGRAM   DRCLS
SET       DOOR  = CLOSE
END
```

```
PROGRAM   VCOPN
SET       VISE  = OPEN
END
```

```
PROGRAM   VCCLS
SET       VISE  = CLOSE
END
.PROGRAM_PUT .CNC1
.SYNC
.FAST
MOVELD    CIM[260]
 IF PART = 2
   ORIF PART =4
   GOSUB GIJIN
 ENDIF
MOVED     CIM[260]
MOVED     CIMB[260]
        SET        DOOR = 0;
        .CNCREQ    .CNC1 OPEN DOOR
        WAIT       DOOR = OPEN
.MEDIOM
MOVED  CIM[261]
SPLINE CIM 262 263
        SET        VISE = 0
```

```
        .CNCREQ   .CNC1 OPEN VISE
        WAIT      VISE = OPEN
.MEDIOM
MOVED     CIM[268]
.SLOW
MOVED     CIM[269]
JAW 50
.MEDIOM
MOVED     CIM[270]
.SLOW
.CLOSE
MOVED     CIM[271]
        SET VISE = 0
        .CNCREQ   .CNC1 CLOSE VISE
        WAIT      VISE = CLOSE
.MEDIOM
MOVED     CIM[270]
MOVED     CIM[262]
.FAST
MOVED     CIM[261]
MOVED     CIM[260]
MOVED     CIMB[1]
        SET       DOOR = 1
        .CNCREQ   .CNC1 CLOSE DOOR
        WAIT      DOOR = CLOSE

.FINISH
.OPEN
.END
.END_PUT
```

Using CNC Script Programs for ACL Program Response

When a CNC script program has completed an operation requested by an ACL program, it must send back an acknowledgment to the ACL program. The following script statement sends this acknowledgment and is explained below:

```
SENDSTR (V1, "RUN DROPN")
```

SENDSTR	Sends the acknowledgment message to the robot's ACL device driver using the OpenCIM network.
V1	A script parameter variable containing the destination address of the robot which is to receive this acknowledgment. This variable is assigned in a parameter file (e.g. CNC1.INI) when the CNC device driver starts up.
"RUN DROPN"	A command to run the ACL program DROPN which sets a semaphore variable indicating that the CNC door is open. This ACL program name corresponds to the operation being performed. For example: DRCLS - door closed VCOPN - vise open

The following sample CNC script program shows how this statement appears at the end of the script after the requested operation has completed:

```
PROGRAM OPEN DOOR
PULSBIT(PORT0,BV0,"00001000",600)
DRAW( --- OPENING THE DOOR --- )
WAITBIT( PORT0, "00000010", 20000 )
DRAW( --- DOOR IS NOW OPEN --- )
SENDSTR( V1, "RUN DROPN" )
END
```

Figure 11-10: Sample CNC Script Program that Sends an Acknowledgment to a Robot

Writing Portable CNC Script Programs

If you have multiple CNC machines which use the same commands, you can reuse the same CNC script file (e.g. CNC_SCR.DBF) to control these machines. Follow the example below which shows how to write portable CNC scripts using a parameter variable to specify a destination address:

Portable:	SENDSTR (V1, "DROPN")
Not Portable:	SENDSTR(WS3 ACL43, "DROPN")

Use a text editor to assign V1 in a parameter file (e.g. CNC1.INI) as follows:

```
:
:
V1 = ACL43
:
:
```

The elements in this example are explained below:

WS3   The name of the Station Manager PC (as found in the file SETUP.CIM) running the CNC device driver. Substitute the appropriate workstation number for the 3 in this example.

ACL   The type of communication channel used to send messages to this device. Use ACL for any device attached to an ACL controller.

43   The device ID that indicates which device driver is to receive this message. Substitute the device ID of the robot which tends this machine for the 43 in this example.

Adjustments for Single Robots Tending Two Machines

If a single robot is tending two CNC machines, make the following adjustments to the sample programs shown in this section:

- In the ACL controller, use two separate global variables, DOOR1 and DOOR2, for each machine (instead of the single variable DOOR).
- Use two separate ACL programs, DRON1 and DRON2, to set these variables (instead of the single program DROPN).

The OPEN DOOR script programs on the CNC machines would call their associated ACL program, DRON1 or DRON2.

ACL Controller Backup and Restore

Because there is always a chance that system files could be altered or destroyed, we recommend keeping backup files of your OpenCIM system. Should your OpenCIM system crash and need to be replaced, the backup files can be used to restore the system. These procedures should only be performed by the system supervisor.

The Backup procedure involves three stages:

1. Back up the ACL controllers to the Station Manager PCs
2. Back up the Station Manager PCs to the CIM Manager PC
3. Back up the CIM Manager PC.

Note

Do not perform Backup procedures while the OpenCIM is running because currently running programs may be aborted and data files may be in an unstable state.

Always keep robot positions, ACL programs and parameters on disk.

Back up and restore the entire system regularly to ensure good backup at all times.

How to Back Up an ACL Controller

An entire backup of the ACL controller includes all robot parameters, positions and programs.

To back up the controller, do the following:

❶
❷
❸
Procedure

Backing Up the ACL Controllers

1. From the ATS main screen on the Station Manager PC, press [Shift + F10] for Backup. The Backup Manager screen appears.

2. Make sure the Backup Directory field is correct before proceeding.

3. Select "Parameter" in Backup/Restore.

4. Type `Parameter` in the File Name field and press [Enter].

5. Press [F3] for backup; a warning message appears stating that `All running programs will be aborted. Are you sure (Y/N)?`

6. Type `Y`; asterisks appear and move across the bottom of the screen representing a time bar. When the time bar lapses, `Done` appears on the screen confirming that the backup copy has been completed successfully.

7. Press [Enter] until the cursor selects "Positions" in Backup/Restore.

8. Type `Position` in the File Name field and press [Enter].

9. Repeat steps 5 and 6 for the backup of Positions.

10. Press [Enter] until the cursor selects "Programs" in Backup/Restore.

11. Type `Programs` in the File Name field and press [Enter].

12. Repeat steps 5 and 6 for the backup of Programs.

13. Press [Enter] until the cursor selects "All" in Backup/Restore.

14. Type `All` in the File Name field and press [Enter].

15. Repeat steps 5 and 6 for the backup of All.

16. Verify that all the backup files you copied can be found in the WS*n* Robot*n* directory. The backup file extension is .CBU (e.g. POSITION.CBU).

Optimization Enhancement Using Open Source

The Optimization Manager of OpenCIM enables users to choose any combination of algorithms and weights for the machine queues, using the algorithms supplied in the installation.

Using Open Source in OpenCIM, the advanced programmer can add new algorithms to the existing list. The new algorithms are automatically added to the OpenCIM Optimization Manager. The user can then choose the new algorithms from the Algorithms drop down list in the Optimization Manager and test them by running the OpenCIM Manager. The new algorithm can use any of the parameters that are included in the OpenCIM database.

For detailed instructions on adding new algorithms refer to the ReadMe file that is provided with the source files of the QueAlgDef project.

Experimenting with Production Strategies Using the A-Plan

The A-Plan is a table of sequential instructions which the CIM Manager executes in order to produce the products being ordered. It is created when you submit an order. You can also edit this table manually with a z editor.

Just as you can compile and run a program without ever examining the associated assembly code, so you can submit an order and run the OpenCIM production line without dealing with the A-Plan. However, advanced users may want to understand the mechanics of OpenCIM production in order to optimize certain critical areas. This section explains the structure of the
A-Plan and how to modify it.

The A-Plan is based on processes and operations that have been set up in the Part Definition table. For each ordered part, the A-Plan lists the procedures required to produce that part.

The relationship between the Part Definition table and the A-Plan is similar to the relationship between source code written in a high-level programming language and the resulting assembly language output after compilation. Dealing with the source code is the easiest way to understand and change a program. However, dealing with the assembly language output might be appropriate for advanced users who need to optimize certain critical areas or who want to understand more fully what is happening at the hardware level. Similarly, the Part Definition table provides an overview of the CIM production process while the A-Plan lists the underlying details.

In addition to user-defined processes, the A-Plan includes the intervening steps required to move parts from machine to machine and from station to station. When an order is submitted, the Order Entry module automatically creates the A-Plan by combining the appropriate material handling commands (described below) with the user-defined processes from the Part Definition table.

A-Plan Commands

The following discussion of the A-Plan assumes that:

- The default storage device is a single ASRS used to house empty templates, raw materials, and finished parts.
- The default assembly device is a jig machine which is used to hold parts that are in the process of being put together by a robot.

The list below shows each command that can appear in the A-Plan table. Related commands are grouped together. These groups are labeled in the Purpose column.

Purpose	A-Plan Command	Description
Loop for Producing Multiple Parts of the Same Type	MAKE	Defines the beginning of a loop used to produce the number of products ordered.
	NEXT	Marks the point at which production of the next ordered part begins. Note that production of the next part may begin before the current part has finished.
Template Commands	DELIVER	Tells the PLC to stop a specific template at a station.
	FREE	Releases an empty template so it can be returned to the ASRS.

Purpose	A-Plan Command	Description
Assemble Two Parts	BASE	Commands a robot to place the first part to be assembled in a jig. The CIM Manager waits for all subparts that belong to this assembly to arrive before placing the base part in the jig.
	PACK	Commands a robot to place a subsequent part to be assembled onto the base part in the jig. The CIM Manager waits for the Base command to finish before it starts executing a Pack command.
	ENDPACK	Signals the end of an assembly operation.
Storage Commands	GET	Reserves a part that is being stored in the ASRS.
	STORE	Sends a part to the ASRS (or other storage location) to be stored.
	RENAME	Assigns the name shown in the Part field to a finished part. This command appears in the A-Plan immediately after the last user-defined process from a Part Definition table has been performed.
Robot Commands	PLACE	Commands a robot to move a part or template from one location at a station to another (i.e. a pick-and-place operation). This command is used to insert parts into devices such as a laser scan meter or robot vision system. However, it is NOT used when placing a part in an assembly jig (see *Base/Pack* above) or a CNC machine (see *Load* below).
	LOAD	Uses a robot to insert a part into a CNC machine and downloads the appropriate G-code to the machine if required. The Load command is similar to the Place command with the added feature that it ensures the required G-code is downloaded..
	UNLOAD	Removes a part from a CNC machine. The Unload command is similar to the Place command.
Comment Line	NOP	This line is ignored. It can be used for adding comments or blank lines to the A-Plan.
User-Defined Process	*Process Name*	Executes the process as defined in the Machine Process table. Each user-defined process that appears in a Part Definition table for this product also appears in the A-Plan table.

The following A-Plan commands shown in detail are representative of how to interpret the other commands.

MACRO	
Format	MAKE <ttl qty> <initial qty> <subsqnt qty> <priority type> <priority #>
Description	Defines the beginning of a loop used to produce the quantity ordered for each line in the Order table.
	SUBPART — Name of the product being ordered. This name is made unique by a suffix which identifies the position of this part in the Part Definition tree.
	Target — Sequential number that corresponds to a line number in the Order table. This number is incremented by one for each occurrence of a Make command in the A-Plan table.
	<ttl qty> — Total number of products to be produced.
	<initial qty> — Number of parts produced in parallel when production of this part first begins.
	<subsqnt qty> — Number of parts produced in parallel after the initial quantity has been completed.
	<priority type> — Priority method used to determine which part order gets produced first. Valid methods are: P - Produce orders with the highest priority first. D - Try to finish all parts by their Due Time. B - Consider both priority and due time when determining the sequence of production.
	<priority #> — Priority of this order (1 - 9). A priority of 1 is most urgent, 9 is least urgent. When *<priority type>* is set to either P or B, the CIM Manager uses *<priority #>* to determine the sequence in which to produce orders.
Example	MAKE COVERED_BOX/1.1 1 1,1,1,P,1
Note	See the NEXT command.

GET

Format   `GET  <subpart> <storage location>`

Description   Reserves a part in a storage location that is needed to produce the current order. The default storage location is the ASRS. Parts can also be stored in a storage rack, a part feeder, or some other designated location. If no part is available, a warning message appears on the CIM Manager screen and a Wait status symbol on the Production screen.

`<subpart>`   The subpart in a storage location that is being reserved.

`<storage location>`   The place in the CIM cell from where you want to order the part. If this field is omitted, the ASRS is assumed.

Example   GET BOX ASRS1

Note   See also STORE and RENAME.

You can view/edit a table of A-Plan commands that is created when you submit an order. Parts that have a 1 in the # column can be produced in parallel (except for a part associated with an ONFAIL process). The commands in this table are executed from top to bottom.

Processes that have a blank entry in their Subpart column operate on the last subpart listed previously. For example, in the figure below, processes 3 and 4 operate on the BOX subpart shown in line 2. Process 6 operates on COVER/1.1 shown in line 5.

Robot pick-and-place operations are not shown in the A-Plan table. The CIM Manager implicitly performs these operations when it needs to move a part from one station location to another.

Figure 11-11: A-Plan Table

You can track the current status of production by watching the Program View screen on the CIM Manager PC. This screen shows the commands that the CIM Manager executes to produce an order.
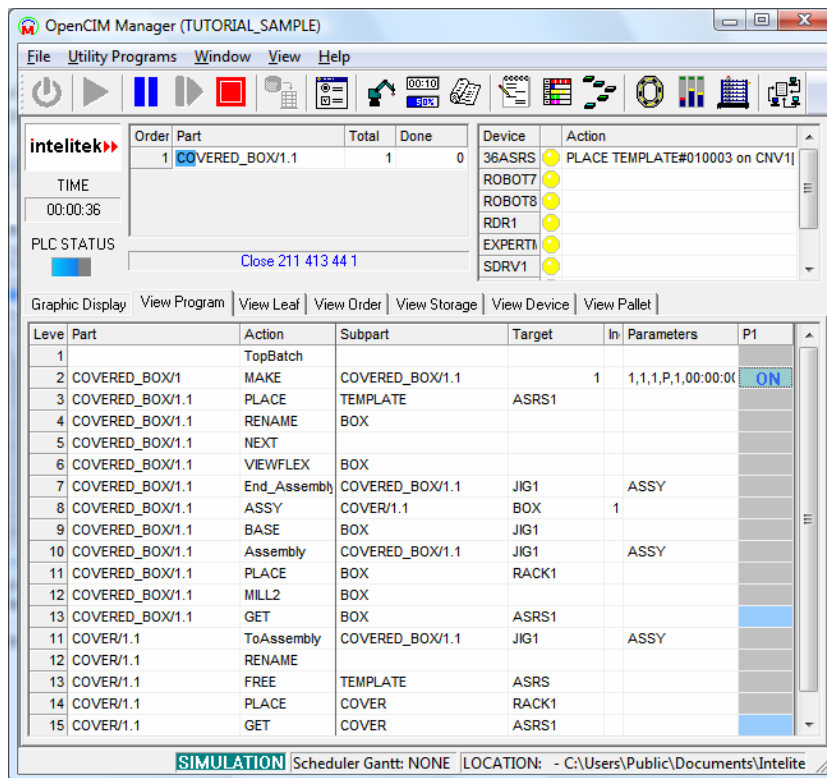
Figure 11-12: Program View Showing A-Plan Used to Produce an Order

The elements of the Program View screen are described in detail in Chapter 6, Operating CIM Manager.

12

Inside OpenCIM

This chapter describes various OpenCIM administration procedures for the advanced user. It includes the following sections

- **OpenCIM Loader: DDLoader.EXE**, describes the OpenCIM Loader that is used to start up all device drivers belonging to the same workstation on each Station Manager PC.

- **OpenCIM Directory Structure**, describes default OpenCIM directory structure of the CIM folders and files that is automatically created in the installation procedure.

- **OpenCIM Database Structure**, displays lists of tables consisting of the files contained in the OpenCIM database

- **Software Backup**, describes the OpenCIM backup procedures.

OpenCIM Loader: DDLoader.EXE

The OpenCIM Loader automates the start-up of the Device Drivers under Windows. The Loader runs on each workstation PC in order to automatically start up the Device Drivers on these PCs:

OpenCIM Device Drivers Started by the Loader	
Station Manager PCs	• Each device driver running on this PC

You can set up the programs you want the Loader to run in an INI file that you specify on the Loader's command line. (In order to display the Path and Command Line columns of the Loader, select Show All Columns from the File drop-down menu.) Clicking on the Loader's icon on the Program Manager screen would thus start up all the programs listed in this INI file. The Loader should be used to start up all device drivers belonging to the same workstation on each Station Manager PC.
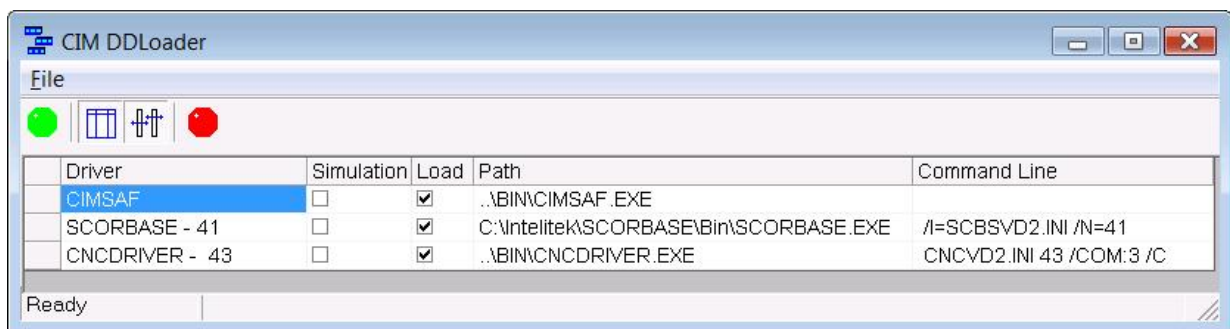
Figure 12-1: CIM Device Loader Sample

The name of this INI file typically corresponds to the station number as follows:

Workstation Naming Conventions	
WS1.INI	Settings for workstation No. 1.
WS2.INI M	Settings for Workstation No. 2

The Loader looks through the INI file for the section entitled *[Loading]* and runs the programs (maximum eight) specified. Use a text editor to edit the command line for each program in this INI file.

The following example of an INI file shows the command lines invoked by the Loader

```
[General]
CimSetupPath = ..\CIMCELL\SETUP\SETUP.CIM
M
[Loading]
Load1=..\BIN\BCRDRIVER.EXE BCRVD1.INI 13 /COM:2 /C

Load2=[default system drive (for example
C:)]\Intelitek\Scorbase\Bin\Scorbase.EXE /I=SCBSVD1.INI
/N=11

Loader Command Lines
```

Each device (or controller) that is connected to a Station Manager PC requires a separate device driver running on that PC. It is possible to have a controller attached to a Station Manager PC that supports multiple devices (e.g. a robot and a bar code reader attached to a Scorbase controller). In this case, only one device driver is required (e.g. a Scorbase device driver). In this section, the term device can also refer to a controller with multiple devices.

The list below shows the name of the program that corresponds to each OpenCIM device driver:

- ACL device driver: `ACLDriver.EXE`
- CNC device driver: `CNCDriver.EXE`
- Laser Scan Meter device driver: `LSMDriver.EXE`
- ROBOTVISIONpro device driver: `RVPDriver.EXE`
- PLC device driver: `PLCDriver.EXE`
- ULS device driver   `ULSDriver.EXE`
- ViewFlex device driver   `ViewFlex.EXE`
- Scorbase for Controller-USB Device Driver   `Scorbase.EXE`
- Scorbase NXC100 and XtraDrive
- Scorbase for Controller-USB PRO controller
- Hydraulic device driver   `HYDDriver.exe`
- Pneumatic device driver   `PNEUDriver.exe`
- Process device driver   `ProcDriver.exe`
- BCR device driver   `BCRDriver.exe`

The table below describes the command line switches you can specify when you invoke a device driver or program module. These switches let you specify the device, data files, and control mode that apply to a program.

All device drivers share the following command line format:

`xxx.EXE IniFile DevID /COM:n [/Simulation]`

(where `xxx` is the type of device driver).

`IniFile`	The name of the initialization file containing parameter definitions for this particular device. These parameter definitions override global values set in the file OPENCIM.INI
`DevID`	The unique ID number of the device as defined in the Setup program. This number is used to identify the device when communicating with the CIM Manager or with another CIM device (e.g. a CNC machine being tended by this robot).
	If more than one device is attached to a controller, you can specify the ID of any one of these devices (e.g. the robot's ID in the case of an ACL controller).
`COM:n`	The RS232 port on the Station Manager PC that the device driver uses to communicate with the device. Com ports 1 - 4 are supported. Com parameters (baud rate, parity, etc.) can be assigned in the initialization file `IniFile`, in the section for this device. For example:
	`[CNCDriverDefinitions]`
	A port value of 0 (zero) indicates that the device driver is to operate in Manual mode.
`/Simulation`	This optional switch starts up the device driver in Simulation mode. In this mode, the device driver emulates a robot by automatically

generating status messages in response to command messages.

You can invoke a device driver in any one of the following ways:

- Use the OpenCIM Loader to automatically start up a set of device drivers listed in the `[Loading]` section of an INI file.
- In the Windows Program Manager, click on an icon which is defined to run the device driver.
- In the Windows Program Manager, select File, Run and enter the device driver command line.

OpenCIM Directory Structure

Each PC on which the OpenCIM software has been installed has the same directory structures.
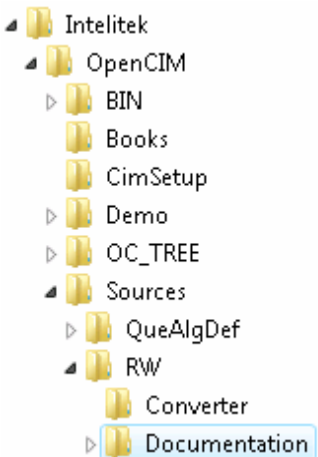
The OpenCIM software and the projects files are installed in separate locations. The file structures of the two are detailed below.

Installation Directory

The CIM software is installed in the programs files directory (for example: [default system drive (for example C:)]\Program Files\Intelitek\OpenCIM\).

Some of the subdirectories are described below:

Subdirectory	Description
Bin	OpenCIM software files
Books	OpenCIM software book
Sources	
QueAlgDef	Queue algorithm source dll to which you can add your own algorithms for the machine queue.
RW	Documentation on how to develop custom parts.
Converter	Application for converting a graphic module from Autocad or 3D studio to RWX.
Documentation	Documentation detailing how to work with the graphic modules.

Figure 12-2: OpenCIM Installation Directory Structure

Projects Directory

CIM projects are installed in the public user directory:

- Windows XP: [default system drive (for example C:)]\Documents and Settings\All Users\Documents\Intelitek\OpenCIM\
- Windows Vista/7: [default system drive (for example C:)]\Users\Public\Documents\Intelitek\OpenCIM\

This directory contains the projects folders.

Project Folder: Contains all project folders and a lib directory which includes preconfigured setup files for common machines, robots and stations.

WorkStation's Folders: In every project folder you can find subfolders with data that is relevant only to a specific station (e.g. WS1, WS2). The project folders can be located on the CIM Manager PC (if you want to centralize the system data) or on that station PC. It is recommended to use the first structure, where the data is kept in a single location as it facilitates backup.

The figure below shows the OpenCIM projects directory structure (path listing) before any OpenCIM system is created using the Virtual CIM Setup.
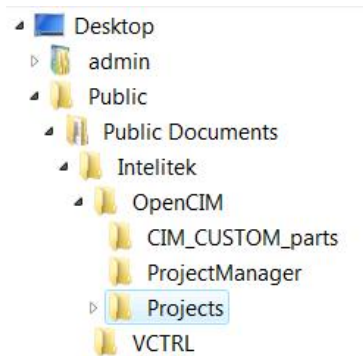
Figure 12-3: OpenCIM Projects Directory Structure

CIM CUSTOM_PARTS: contains all the RWX files for parts provided by Intelitek. For information on modifying these parts and creating new ones, refer to Chapter 7: CIM Manager Utility Programs.

For every cell created by the Virtual CIM Setup, a subdirectory, which contains the following subdirectories, is created:

Subdirectory	Description
DATA	Contains data files that change during the normal operation of the CIM.
DOC	Contains all documentation relevant to this particular system.
LOG	Contains a log of the system.
SETUP	Contains all the files that change when setting up the CIM cell.
WS0	Workstation Manager.
WS*n*	Contains all data files that are unique to this station.
VC2_WM.DBF	OpenCIM network messages file.
CIMCELL.INI VC2.INI CIMCELL.O2B CIMCELL.O2C CIMCELL.O2P	These files contain graphic information of the cell.

The WSn subdirectories represent each of the workstations in the CIM cell.

The subdirectory is created in the CIM Manager for each specific WSn. and is then shared via the network with the Station Manager. A subdirectory is created for each machine at that workstation.

A system with three workstations (WS1, WS2, WS3) may be set up as follows, for example:

- The WS1 subdirectory contains all INI files and DBF files that are unique to workstation #1. ROBOT 1 contains all of the robot programs specific to workstation #1.
- The WS2 subdirectory contains all INI files and DBF files that are unique to workstation #2. ROBOT 2 contains all of the robot programs specific to workstation #2. GCODE contains all the G-code programs necessary for the CNC machine.
- The WS3 subdirectory contains all INI files and DBF files that are unique to workstation #3.

The table below lists the files found in a typical OpenCIM system, grouped by subdirectory. The subdirectory generically called CIMCELL represents a typical CIM cell.

File Type	Description
Projects Directory\LIB\ACL	Subdirectory containing a library of generic ACL source code, utility programs, and parameter files.
CIMSYS.DMC	Source code of standard ACL system programs for a single robot attached to an ACL controller.
CIMSYSM.DMC	Source code of standard ACL system programs for multiple robots attached to the same ACL controller.
Projects Directory\LIB\ACL\ATS	
DOWNLOAD.EXE	ACLoff-line downloader utility program that sends DNL files to an ACL controller.
SEND.EXE	Utility program to send a command to an ACL controller.
TERM_ACL.EXE	ATS terminal emulation program used to interact with an ACL controller.
SETUP.DIR	A data file used by the TERM_ACL program.
ASRSB.PRB ASRSSQ.PRB BELT.PRB LSB100CM.PRB LSB150CM.PRB LSB20.PRB NOCONECG.PRB NOCONECT.PRB ROTARY_B.PRB SERVOG_C.PRB XYTAB.PRB SCC_BELT.PRB	ACL parameter files.
SETUP.PAR	Location of the Robot parameters.
ATS.BAT	Batch file to load the TERM_ACL.EXE.
TERM.MAC	
ONOFF.CBU	On/Off program for the controller.
PAR.CBU	File containing parameters for the controller.
PAR14.CBU	File containing parameters for Robot ER 14.
PAR9.CBU	File containing parameters for Robot ER 9.
PARMK2.CBU	File containing parameters for Robot MK2.

File Type	Description
BIN	Subdirectory containing OpenCIM program files (EXE, DLL, HLP, ICO, TRK, VBX).
C4DLL.DLL	Object code files containing a shared library of Windows programs.
DBCREATER.DLL	
QueAlgDef.dll	Machine queue algorithms Dll
VBC4DLL.DLL	
OpenCIM_Ver4.5.chm	Help file
ACLDRIVER.EXE	The ACL device driver.
APLAN.EXE	The A-Plan program.
BCRDriver.exe	The bar-code device driver
CHECKCOMMUNICATION. EXE	Utility program for testing TCP/IP communications.
CIMREPORT.EXE	Report Generator program
CIMSAF.EXE	OpenCIM safety device driver
CIMSETUP.EXE	Virtual CIM Setup program
CIMSIMUL.EXE	Graphic Display Program Loader
CNCDRIVER.EXE	CNC device driver
DBTOOL.EXE	Edits database files
DDLOADER.EXE	Loader program for device drivers
HYDDriver.exe	Hydraulic device driver
LSMDRIVER.EXE	Laser Scan Meter device driver
MACHINEDEFINITION. EXE	Machine Definition module
MANAGER.EXE	CIM Manager module
MRP.EXE	MRP module
Optimization.exe	Optimization Utility
PARTDEFINITION.EXE	Part Definition module
Performance.exe	Performance Utility
PLCDRIVER.EXE	PLC device driver
PNEUDriver.exe	Pneumatic device driver
ProcDriver.exe	Process device driver
ProjectManager.exe	Project Manager Program
RFIDDriver.exe	RFID Device Driver
RVPDRIVER.EXE	ROBOTVISIONPro Device Driver
SCHEDULER.EXE	Scheduler Module
SCRIPTER.EXE	Activates Graphic Display commands
STORAGEMANAGER.EXE	Storage manager module
ULSDRIVER.EXE	Laser Engraver Device Driver
CIMCELL	Generic name for a subdirectory containing all data for one particular CIM cell, as created by the Virtual CIM Setup.

File Type	Description
CIMCELL / DATA	Subdirectory containing data files used by the OpenCIM system. (*.DBF is a file in dBASE format).
ORDER.CFG	Internal configuration parameters used by the MRP.
APLAN.DBF	A-Plan commands generated by last order submitted.
BACK.DBF	The REF.BAT batch file uses this data file to restore STORAGE.DBF.
CSTORDER.DBF	Details of the customer order.
CUSTOMER.DBF	List of customers.
CIMREP.DBF	Report format specifications.
LEAFPART.DBF	Internal information used by the CIM Manager to display active production operations in Leaf View.
MACHINE.DBF	List of machines as defined in the Machine Definition.
OPT_MQUEUE.DBF	List of which algorithm to use for each machine queue.
ORDER.DBF	Records that appear in the Order table.
PART_DEF.DBF	Records that appear on the Part Definition screen.
PART_PRC.DBF	Records that appear in the Part Process table.
PERF_SUMM.DBF	System performance summary.
PROCESS.DBF	Records that appear in the Machine Process table in the Machine Definition module.
Performance.dbf	Machine performance summary.
REPORT.DBF	Records that appear in the Report table.
PROCESS.DBF	Machine process sumarry.
SCHEDULER.BDF	Information used by the scheduler.
PURCHASE.DBF	List of item parts for purchase.
STORAGE.DBF	Contents of all storage devices.
SCHEDULER.DBF	Machine task schedule summary.
SUPPLIER.DBF	List of Suppliers.
TEMPLATE.DBF.DBF	Conveyor template definitions as assigned in the Storage Definition module.
ANALYSIS.RPT APLAN.RPT ASRS.RPT ASRSSUM.rpt LOCATION.RPT MACHINE.RPT OPTIMIZATION.rpt ORDER.RPT PART.RPT partprocess.rpt performance.rpt PROCESS.RPT PURCHASE.RPT PurchaseNew.rpt	Listing of files for the Report Generator. Each file is named according to the report and/or application it applies to.

File Type	Description
SUBPART.RPT	
MACH_MAC.CDX P_DEF_AP.CDX PROC_AP.CDX PDEF_PAR.CDX PROC_PAR.CDX P_DEF_OR.CDX MACH4TAG.CDX P_PRC_AP.CDX ORDER_AP.CDX PART4TAG.CDX PARTD_AS.CDX	dBASE index files automatically generated by the programs which update the associated DBF files.
STR_TEMPLATE.CDX	
CIMCELL **/ LOG**	Subdirectory containing LOG files for the system.
CIMCELL **/ SETUP**	Subdirectory containing installation and configuration related files.
MAP.INI	Data file that associates PC network file names with workstation numbers. Also used to assign multiple devices to a single device driver.
SETUP.CIM	Configuration data file; contains all devices and their associated parameters.
DEVICE.DMC	Assignment of device names to ACL program numbers.
OPENCIM.INI	Default device driver parameters.
FDR1.INI	File detailing the files used by each storage device.
RACK1.INI	File containing the configuration of the rack.
PNEUST1FDR1	File containing the configuration of the pneumatic station.
CONPALET.INI	File containing the minimum number of empty pallets running in the conveyor.
CIMCELL **/ WS0**	Subdirectory containing files for the CIM Manager PC.
CIM.LOG	OpenCIM network messages file for this station.
CIM.PRT	File containing a listing of all messages (LOG file).
WS0.INI	Parameters passed into the CIM Manager at initialization time.
CIM.PNP	Pick-and-place file.
CIM.LOG	File containing the leaf view (LOG file).
ERROR.LOG	Error graphic file
OLMT.TXT	List of Messages sent to Simulation (Log file).
CIMCELL **/ WS***n*	Subdirectory containing files for a typical workstation.
ACLVD1.INI	ACL Virtual driver.
VC2_QC.INI	Format settings for quality control report.
BCRVD1.INI	BarCode Virtual driver.
SCBSVD1.INI	Scorbase Virtual driver

File Type	Description
PNEUVD1.INI	Pneumatic Station Virtual driver
$ACL_011.PRT	File containing ACL device driver protocol.
CNCVD1.INI	CNC Virtual driver.
ULSVD1.INI	ULS Virtual driver
HYDVD1.INI	Hydraulic Station Virtual driver
VFVD1.INI	ViewFlex Virtual driver
PROCVD1.INI	Process Virtual driver
$LSM_009.PRT	File containing LSM device driver protocol.
LSMVD1.INI	LSM Virtual device driver INI file.
$PLC_001.PRT	File containing PLC device driver protocol.
PLCVD1.INI	PLC virtual device driver.
WS1.INI	Typically used to pass command line parameters to the Loader program used to start each device driver at this station in Real mode.
ACL_011.PNP	Pick-and-place file.

Every machine, robot or station directory in a workstation contains a number of preconfigured setup files.

The following is an example of a robot directory which contains files for the Robots in the stations.

CIMCELL **/ WS*n*** **/ ROBOT*n***	Subdirectory containing files for the ACL controller at Station *n*.
REPORT.DLD	Log file showing the commands executed during the last ACL download (see the *ACLoff-line* manual).
WS2.DNL TRASH1.DNL BFFR1.DNL RDR1.DNL RACK1.DNL CNV1.DNL EPILOG1.DNL PROLOG1.DNL FDR1.DNL ASMBUF.DNL ASRS.DNL	Listing of all the download robot programs for this station.
RVPCOM2V.QCL RVPCOMOV.QCL RVPCOM1V.QCL RVPCOM.QCL	Listing of all download Quality Control robot programs.
SETUP.DIR	Data file used by the TERM_ACL program.
CIMCELL / WS*n* / ROBOT*n*	Subdirectory containing files for controllers operated by Scorbase software
STN.WS	

File Type	Description
STN.SBP	
STN.PNT	
LIB	Subdirectory containing preconfigured setup files for common machines, ACL programs, etc. (library of ACL programs).
	Located in the projects directory
LIB / ACL	Subdirectory containing a library of unique ACL source code, utility programs, and parameter files.
Projects Directory\LIB\ACL	Subdirectory containing a library of generic ACL source code, utility programs, and parameter files.
CIMSYS.DMC	Source code of standard ACL system programs for a single robot attached to an ACL controller.
CIMSYSM.DMC	Source code of standard ACL system programs for multiple robots attached to the same ACL controller.
Projects Directory\LIB\ACL\ATS	
DOWNLOAD.EXE	ACLoff-line downloader utility program that sends DNL files to an ACL controller.
SEND.EXE	Utility program to send a command to an ACL controller.
TERM_ACL.EXE	ATS terminal emulation program used to interact with an ACL controller.
SETUP.DIR	A data file used by the TERM_ACL program.
ASRSB.PRB ASRSSQ.PRB BELT.PRB LSB100CM.PRB LSB150CM.PRB LSB20.PRB NOCONECG.PRB NOCONECT.PRB ROTARY_B.PRB SERVOG_C.PRB XYTAB.PRB SCC_BELT.PRB	ACL parameter files.
SETUP.PAR	Location of the Robot parameters.
ATS.BAT	Batch file to load the TERM_ACL.EXE.
TERM.MAC	
ONOFF.CBU	On/Off program for the controller.
PAR.CBU	File containing parameters for the controller.
PAR14.CBU	File containing parameters for Robot ER 14.
PAR9.CBU	File containing parameters for Robot ER IX.
PARMK2.CBU	File containing parameters for Robot MK2.
LIB/ Hydraulic	Subdirectory containing hydraulic station files.
Load.pgm	
Press.pgm	

File Type	Description
`Unload.pgm`	
LIB/ QC	Subdirectory containing examples of Quality Control INI files.
`VC2_QC.INI`	
LIB/ SCR	Subdirectory containing all the CNC script and BATCH file examples.
`SCRLAB.DBF`	
`SCRILAB.DBF`	
`SCRTIL.DBF`	
`SCRSITIL.DBF`	
`SCRPRJ.DBF`	
`VC2_WM.EXP`	
`CNC_L.LAB`	
`EXAMPLE.DBF`	
`CNC_L.BAT`	
`CNC_SCR.DBF`	
`CNCSCRSI.DBF`	
`LIB/ ViewFlex`	Subdirectory containing all the ViewFlex Scripts
`CHECK_V.bas`	
`FIND_4_PINS.bas`	
`FIND_4_SCREWS.bas`	
`Lower_Left_Pin.mod`	
`Lower_Left_Screw.mod`	
`Lower_Right_Pin.mod`	
`Lower_Right_Screw.mod`	
`Upper_Left.mod`	
`Upper_Left_Pin.mod`	
`Upper_Left_Screw.mod`	
`Upper_Right_Pin.mod`	
`LIB/ Vuniq40`	Subdirectory containing required files to work with with Vuniq.

MAP.INI

The setup file MAP.INI performs the following functions:

- Contains the TCP/IP configuration of the manager and all device drivers in an OpenCIM cell. This information is required by each OpenCIM in order to enable communication with other OpenCIM entities.
- Allows two or more devices to use the same device driver to send and receive messages

A typical MAP.INI has the following format:

```
[Redirect]
53=51
```

```
23=21
24=21
[COMMMANAGER]
RemoteIP=200.1.1.1
RemotePort=700


[COMMACL11]
RemoteIP=200.1.1.1
RemotePort=711

[COMMLSM13]
RemoteIP=200.1.1.1
RemotePort=713

[COMMACL21]
RemoteIP=200.1.1.1
RemotePort=721


[COMMACL24]
RemoteIP=200.1.1.1
RemotePort=724
```

When a device driver starts, it must access MAP.INI in order to be able to communicate with other device drivers and the CIM Manager. On each Station Manager PC the file path to MAP.INI is contained in the parameter variable *CimMapPath* in the *[Networking]* section of the INI file for this station.

Note

*To guarantee proper operation of OpenCIM, be sure that the file MAP.INI is a*ccessible to all device drivers*. To be sure that the MAP.INI file is accessible, do the following:*

- *Open any device driver*
- *Check if the MAP.INI file is updated by editing the TCP/IP configurations of the DD opened*

You can use any ASCII text editor (e.g. Windows Notepad) to modify MAP.INI. The two types of entries in this file are described below.

Since only one device can be assigned to a device driver on a command line, an ID entry in MAP.INI is used to assign other devices to this device driver. For example, a bar code reader (device 13) and a robot (device 11) may be connected to the same ACL controller and thus share the same ACL device driver for passing OpenCIM messages. The following command line assigns the robot to the ACL device driver:

```
ACLDriver.EXE ACLVD1.INI 11,13 /COM:2
```

The following MAP.INI entry allows the bar code reader to share the use of this device driver:

```
[REDIRECT]
13=11
```

| 13 | A device that is sharing the use of a device driver. |
| 11 | The primary device that is assigned to a device driver on the command line that invokes the device driver. |

SETUP.CIM

SETUP.CIM is an ASCII file which defines all devices found in the OpenCIM system. The file is located in the path *Projects Directory*\CIMCELL*SETUP*. The fields in this file are separated by a space.

The SETUP.CIM file for the CIMLAB4 Virtual CIM is shown below.

```
5
51 1 ER5P ROBOT1 R 1 0 0 0 0  0
52 2 ER9 ROBOT2 R 1 0 0 0 0  0
53 3 ER5P ROBOT3 R 1 0 0 0 0  0
54 4 ER7 ROBOT4 R 1 0 0 0 0  0
0 0 PLANE PLANE 0. 0. 0.
1 2 CNV1 CNV1 C 4 0 0 0 4  ROBOT1 0 ROBOT2 0 ROBOT3 0 ROBOT4 0 0
55 4 ERXY XYJIG J 1 0 0 0 1 ROBOT4 0 0
2 1 RNDAS RNDAS1 A 54 0 0 0 1  ROBOT1 0 0
3 1 READER RDR1 Y 1 0 0 0 1  ROBOT1 0 0
4 2 MILL_S MILL1 M 1 0 0 0 1  ROBOT2 0 0
5 2 M2AS BFFR1 B 2 0 0 0 1  ROBOT2 0 0
6 2 M2AS BFFR2 B 2 0 0 0 1  ROBOT2 0 0
7 3 LATH_S LATHE1 M 1 0 0 0 1  ROBOT3 0 0
8 3 M2AS BFFR3 B 2 0 0 0 1  ROBOT3 0 0
9 4 M2AS BFFR4 B 2 0 0 0 1  ROBOT4 0 0
10 4 FEEDER FDR1 F 1 201 0 20 1  ROBOT4 0 0
11 4 RACK RACK1 K 9 101 0 0 1  ROBOT4 0 0
12 4 RACK RACK2 K 2 102 0 0 1  ROBOT4 0 0
13 4 TRASH TRASH1 X 1 0 0 0 1  ROBOT4 0 0
14 4 JIG JIG1 J 1 0 0 0 1  ROBOT4 0 0
17 4 VISION VSN1 V 1 0 0 0 1  XYJIG 0 0
16 4 SCREWD SDRV1 D 1 0 0 0 1  XYJIG 0 0
51 1 ER5P ROBOT1 R 1 0 0 0 0  0
52 2 ER9 ROBOT2 R 1 0 0 0 0  0
53 3 ER5P ROBOT3 R 1 0 0 0 0  0
54 4 ER7 ROBOT4 R 1 0 0 0 0  0
60 0 SPARE    CONPALET O 16 1   0   0   0 0
60 0 SPARE    CONPALET O 16 1   0   0   0 0
```

Fld#	Field Name	Type #	Width	Remarks
1	Device (Object/Location) ID (Do not skip numbers, make all IDs sequential)	Numeric	3	
2	Station Number	Numeric	2	
3	Physical Name (for graphic display)	Character	20	
4	Logical Name	Character	20	
5	Type: A: ASRS M: Machine B: Buffer O: Conpallet C: Conveyor P: Part D: Device (e.g. screwdriver) Q: Quality Control R: Robot F: Feeder H: Reserved S: User Defined I: Station U: Undefined J: Jig V: Vision K: Rack X: Trash L: Laser Scan Meter Y: Barcode or RFID Z: AGV	Character	1	
6	Capacity	Numeric	2	
7	Subtype (type of objects it can hold) or Connectivity (used with Feeder, Rack, Conpallet and Buffer)	Numeric	3	
8	Location ID (for graphic display)	Numeric	3	
9	Location Position (for graphic display)	Numeric	2	
10	Number of Robots (that can access this device)	Numeric	2	
11	Robot's Name	Character	20	Multiple field
12	Robot's Position	Numeric	2	Multiple field
13	Steady Flag	Logical	1	

The following is an example of a SETUP.CIM file.

```
1
11 1 SQRAS SQRAS1 R 1 0 0 0 0  0
211 1 ASRS ASRS1 A 72 0 0 0 1 SQRAS1 0 0
21 2 ER9 ROBOT5 R 1 0 0 0 0  0
31 3 MK3 ROBOT3 R 1 0 0 0 0  0
41 4 ER14 ROBOT4 R 1 0 0 0 0  0
0 0 PLANE PLANE 0. 0. 0.
1 1 CNV1 CNV1 C 4 0 0 0 4  SQRAS1 0 ROBOT5 0 ROBOT3 0 ROBOT4 0 0
12 1 READER RDR1 Y 1 0 0 0 1  SQRAS1 0 0
24 2 PCMILL PCMILL1 M 1 0 0 0 1  ROBOT5 0 0
23 2 PCTURN PCTURN1 M 1 0 0 0 1  ROBOT5 0 0
33 3 WELDST WELDST1 D 1 0 0 0 1  ROBOT3 0 0
49 4 CMM CMM1 Q 1 0 0 0 1  ROBOT4 0 0
47 4 VISION VSN1 V 1 0 0 0 1  ROBOT4 0 0
44 4 RACK RACK1 K 1 101 0 0 1  ROBOT4 0 0
45 4 RACK RACK2 K 1 102 0 0 1  ROBOT4 0 0
43 4 JIG JIG1 J 1 0 0 0 1  ROBOT4 0 0
48 4 TRASH TRASH1 X 1 0 0 0 1  ROBOT4 0 0
22 2 M2AS BFFR1 B 2 0 0 0 1  ROBOT5 0 0
32 3 M2AS BFFR2 B 2 0 0 0 1  ROBOT3 0 0
42 4 M2AS BFFR3 B 2 0 0 0 1  ROBOT4 0 0
46 4 FEEDER FDR1 F 1 103 0 0 1  ROBOT4 0 0
50 0 SPARE CONPALET O 16 1   0  0  0 0
```

DEVICE.DMC

The DEVICE.DMC file in the SETUP directory defines numbers for the ACL logical name of every device in the system.

The DEVICE.DMC file for the CIMLAB4 Virtual CIM is shown below.

```
#IFNDEF _DEVICE_DMC
#DEFINE _DEVICE_DMC
    #DEFINE CNV1            001
    #DEFINE RNDAS1          002
    #DEFINE BFFR1           005
    #DEFINE BFFR2           006
    #DEFINE BFFR3           008
    #DEFINE BFFR4           009
    #DEFINE FDR1            010
    #DEFINE RACK1           011
    #DEFINE RACK2           012
    #DEFINE TRASH1          013
    #DEFINE ROBOT1          051
    #DEFINE ROBOT2          052
    #DEFINE ROBOT3          053
    #DEFINE ROBOT4          054
    #DEFINE ROBOT5          055
    #DEFINE MILL1           004
    #DEFINE LATHE1          007
    #DEFINE JIG1            014
    #DEFINE SDRV1           016
    #DEFINE VSN1            015
    #DEFINE RDR1            003
#ENDIF
```

For an explanation of this file and how to edit it, refer to Writing ACL Source Code in Chapter 11, OpenCIM Programming.

INI Files

OpenCIM uses a set of INI files to set configuration parameters for the following programs:

- Each OpenCIM device driver
- The CIM Manager
- The OpenCIM Loader
- The Storage Definition module

OpenCIM parameter settings for the above programs are stored in a set of text files: *.INI. These files use the same structure as standard Windows INI files such as WIN.INI. These programs read their respective INI files at initialization time. If you make a change to a program's INI file after it has started, you must end the program and restart it for the change to take effect.

Default values for all devices are stored in the file OPENCIM.INI. Settings for individual devices can be made in a local INI file which is specified on the device driver loader's command line. If the same parameter appears in both OPENCIM.INI and a device driver's local INI file, the local setting takes precedence. All parameters must appear in either OPENCIM.INI or in a local INI file.

The (OPENCIM32\CIMLAB4\SETUP\) OPENCIM.INI file for the CIMLAB4 setup is shown below.

```
[General]
CimSetupPath=..\SETUP\SETUP.CIM
CimSetupDir=..\SETUP
CimLibDir=..\LIB
CimDataDir=..\DATA
CimWorkDir=..\data
CimReportDir=..\DATA

[Networking]
CimMapPath=..\SETUP\MAP.INI
Timer=300
AttemptsCount=3
PassCount=3
OffDelay=3
EchoFilter=1124

[Simulation]
PCPLC=20,7,15

[ASRS1]
NumberOfRows=6
NumberOfCols=6
NumberOfGrids=2
FirstGridWithMinIndex=Bottom
LocationMinIndexGrid=LeftBottom
DirectionIncIndexGrid=Right

[WndStatus]
PROGDEF=0
ORDERDER=0
STRGDEF=0
DEVICEDEF=0
LOGDEF=0
PALLETEDEF=0
LEAFDEF=0
EVENTDEF=0
MANAGERDEF=0
HYSTORYDEF=0

[CIMMODES]
TRACKINGMODE=0
```

```
UPDATEDURATION=0
SENDTOGRAPH=0
SIMULATION=1
CIMSPEED=1

[DDFileName]
CNV1=..\CIMLAB4\WS1\PLCVD1.INI
SQRAS1=..\CIMLAB4\WS1\ACLVD1.INI
RDR1=..\CIMLAB4\WS1\LSMVD1.INI
ROBOT5=..\CIMLAB4\WS2\ACLVD5.INI
PCTURN1=..\CIMLAB4\WS2\CNCVD1.INI
PCMILL1=..\CIMLAB4\WS2\ CNCVD2.INI
ROBOT3=..\CIMLAB4\WS3\ACLVD3.INI
ROBOT4=..\CIMLAB4\WS4\ACLVD4.INI
VSN1=..\CIMLAB4\WS4\RVPVD1.INI
CMM1=..\CIMLAB4\WS4\LSMVD2.INI
```

You can use some combination of the parameter file strategies listed below when deciding how to organize INI files for your system:

Put all default parameter settings in OPENCIM.INI. Then write a separate INI file for each device driver that contains only those parameters that deviate from the default.

Have a separate INI file for each device driver containing all the parameters for that device. Use OPENCIM.INI only for global parameters.

Have a separate INI file for each Station Manager PC (e.g. WS1.INI) which contains the parameters for all devices at that station. These files would also contain the command lines used by the Loader to start each device driver at a station. Use OPENCIM.INI only for global parameters.

To set up or edit an INI file, use a text editor (e.g. Windows Notepad) that can save files in ASCII format. The file is divided into sections with the title of each section enclosed in brackets, for example [Networking]. A program searches for the sections that apply to it and reads the associated parameter settings. Keep the following considerations in mind when editing an INI file:

Only lines that begin with valid parameter names are read (i.e. only parameters that belong in that section), all other lines are ignored. If you misspell a parameter name, it will be ignored.

A leading space at the beginning of a line will cause that line to be ignored.

Do not insert the same parameter more than once in a section (there is no guarantee as to which value will be used).

You can insert blank lines in an INI file to group related parameters and to set off sections.

You can create a comment line by inserting an extra character at the beginning of a line (by convention the semicolon, ;). For example, if you want to try a new text color but also keep the original value for reference, you could do the following:

```
;MainWindowTextColors  = 0,255,0
 MainWindowTextColors  = 0,255,255
```

The table below lists all OpenCIM parameters that you can set. Note that some of the values in the table are internal system parameters which should only be changed under the direction of Intelitek technical support. The table is divided into the following sections:

INI Parameter	Description
Default Settings (OPENCIM.INI)	
`[General]`	The parameters in this section define the OpenCIM directory structure on the central server PC.
`CimLibDir=..\LIB`	Location of the original OpenCIM data and program files. These data files can be copied to the SETUP directory to restore the system to its original state.
`CimDataDir = ..\DATA`	Location of data files which define the OpenCIM configuration (e.g. machine processes, part definitions, etc.).
`CimWorkDir = ..\DATA`	Location of data files, including log files, which are updated during OpenCIM production.
`CimReportDir = ..\DATA`	Location of report output. Other software applications can interface here to pick up OpenCIM production data.
`[Networking]`	
`CimMapPath = ..\SETUP\MAP.INI`	The location of the OpenCIM map file which contains: • The name of all workstation PCs. • A list of all devices which share the use of a device driver (e.g. a robot, and an automatic screwdriver all connected to the same ACL controller).
`Timer = 1000`	The frequency at which a program checks its mailslot for messages to transmit or receive. The CIM Manager must check its mailslot more frequently than the programs with which it communicates (device drivers or Storage Definition module). CIM Manager: 200 - 600 ms Device drivers: 400 - 1000 ms Reserved for Intelitek technical support use only.

INI Parameter	Description
PassCount = 3	Number of Timer intervals to wait before a retry is sent. For example:
	`Timer = 1000, PassCount = 3` → delay before retry is 3000 ms
	Reserved for Intelitek technical support use only.
OffDelay = 3	Number of milliseconds a device driver waits for confirmation from the CIM Manager that it has received a shutdown notification message. A value of 0 (zero) causes the device driver not to send a shutdown message.

ASRS Device Driver Settings

```
[Structure]

NumberOfRow = 6

NumberOfCols  = 6

NumberOfGrids = 2

FirstGridWithMinIndex = Bottom
LocationMinIndexGrid = LeftUp
DirectionIncIndexGrid = Right
```

ACL Device Driver Settings

```
[General]
```

```
CimSetupPath =..\CIMCELL\SETUP\SETUP.CIM
```
Tells the device driver where to find the following important OpenCIM setup files:

- OPENCIM.INI: Contains default parameter settings for all device drivers. The file OPENCIM.INI must appear in the same directory as the setup file named in this setting.

- SETUP.CIM: Lists all physical devices and their IDs. The name SETUP.CIM is the default name for this file.

This parameter setting must appear in each local INI file. It is not needed in OPENCIM.INI.

```
[ACLDriverDefinitions]
```

```
ACLDriverPromptNum = 3
```
Number of times the device driver sends a query to an ACL controller to invoke the controller's command prompt, ">". Reserved for Intelitek technical support use only.

```
BaudRate = 9600
Parity   = None
DataBits = 8
StopBits = 1
XonXoff  = Yes
```
These are the standard RS232 settings for communicating with an ACL controller. They should not be changed since they match the fixed settings in the controller.

```
MainWindowBkgndColors = 0,0,0
MainWindowTextColors  = 0,255,0
```
MainWindowBkgndColors = 0,0,0
MainWindowTextColors = 0,255,0
Color settings for the background and text colors in the device driver's Status window. When a Station Manager PC is running several device drivers simultaneously, these parameters allow you to set each one to a different color in order to distinguish between them.

The three numbers represent a color using the RGB color scheme (Red, Green, Blue). Values range from 0–255. A 0 (zero) indicates the absence of red, green, or blue, respectively, and 255 signifies a primary color at full intensity. For example:

Green = 0,255,0
White = 255,255,255
Black = 0,0,0

Do not set the text color to the same value as the background color. This combination would cause the text to become invisible.

CNC Device Driver Settings

```
[General]

CimSetupPath= ..\CIMCELL\SETUP\SETUP.CIM
```
See this parameter in the ACL section above.

```
[CNCDriverDefinitions]
```

```
CNCScriptFilePath = CNC_SCR.DBF
CNCScriptDebugFilePath  = CNC_SCRS.DBF
```
These files contain the CNC script programs which appear in the CNC Command List on the Control Panel.

This first parameter specifies a file that is used when the CNC device driver is operating in Real Mode. The second parameter is an alternate file used when the device driver is operating in Simulation or Manual mode.

The file name is mandatory. If no path is specified, the current working directory is used as defined by the device driver's /C command line switch.

```
MainWindowBkgndColors = 0,0,0
MainWindowTextColors = 255,255,255
```
See these parameters in the ACL section above.

```
; --- CNC Variables ---
```

```
V1 = ACL21
V2 =
V3 =
V4 =
V5 =
V6 =
M
V16 =
```
Parameter variables used by CNC script programs. These variables are used to write portable CNC script programs. For further information, see "CNC Device Driver" in Chapter 8.

```
BV0 = 0
BV1 = 0
```
These 8-bit values, which range from 0-255, are assumed to be the initial state of the control lines of a CNC machine when the system is turned on.

```
PORT0 = 0x500
PORT1 = 0X501
```
I/O port addresses on a Station Manager PC that are mapped to the status and control lines of a CNC machine using a special interface card. These values should match the jumper settings on the I/O card.

```
CNCDriverDebuggerTimer = 100
```
Refresh interval (in milliseconds) for display of input port values on the CNC Control Panel.

Reserved for Intelitek technical support use only.

```
CNCDriverTimer = 10
```
This is an internal timer (in milliseconds) used during communications.

Reserved for Intelitek technical support use only.

```
CNCIdleProcTimer = 10
```
Polling interval (in milliseconds) for checking the alarm status of a CNC machine.

Reserved for Intelitek technical support use only.

`CNCIdle = CNCIDLE(port0, "00000010")`	This setting checks for an alarm condition on the specified input port. If the value of the input port matches this mask, the CNC device driver sends the following alarm message to the CIM Manager: WM_CIMDDE_CNCERROR with the error value = CNCALARM

`BaudRate = 9600`
`Parity   = None`
`DataBits = 8`
`StopBits = 1`
`XonXoff  = No`
	RS232 parameters used by the CNC device driver when it downloads G-code to a CNC machine. You must set these parameters to match the RS232 settings on the CNC machine.

`Loader =Project directory`
`\WSN\taskload.bat`
`TaskLoadedMark = TASK.CNC`
	Include the `Loader` parameter when you want to use your own program to download G-code to a CNC machine (instead of using the device driver's built-in downloader).
	
	The `Loader` file name must be a batch file. The last command in this file should create the flag file specified in `TaskLoadedMark`. This flag file signals that the download is complete. The device driver automatically deletes this flag file each time it invokes the `Loader` batch file.

LSM Device Driver Settings

`[General]`	
`CimSetupPath== ..\CIMCELL\SETUP\SETUP.CIM`	See this parameter in the ACL section above.
`[LSMDriverDefinitions]`	
`QCReport = No`	Enables/disables creation of a log file for capturing results from this quality control device. If set to No, all other QC report parameters below are ignored.
`QCReportTemplateFile = VC2_QC.INI`	Name (including path) of the file that defines the format of the quality control log file.
`QCReportFileName =`	Name (including path) of the quality control log file itself.
`QCReportFileMarker =`	A flag file (including path) which indicates that the quality control log file has been updated. A user application can delete this flag file after processing the log file. The next time the flag file appears, the application knows that there is new log file data to process.
`QCReportFileDeleteOnStart =`	This switch controls whether a new quality control log file will be created each time this device driver starts up. If Yes, the previous log file is deleted. If No, results are appended to the existing log file.

`SimulationFailPercent = 20`	When the device driver is operating in simulation mode, this value determines what percentage of the simulated quality control tests are randomly reported as failures. This setting provides the default value that appears in the Fail % box on the QC Control Panel. The range is 0–100 (0 = test always passes, 100 = test always fails).

`BaudRate=9600`
`Parity=None`
`DataBits=8`
`StopBits=1`
`XonXoff=No`
	RS232 parameters used by the QC device driver when it communicates with the controller for the quality control device. You must set these parameters to match the RS232 settings on the device's controller.

`MainWindowBkgndColors=40,150,100` `MainWindowTextColors=100,50,200`	See these parameters in the ACL section above.

RFID Device Driver Settings

`[General]` `CimSetupPath=..\CIMCELL\SETUP\SETUP.CIM`	See these parameters in the ACL section above.
`[Networking]` `CimMapPath=..\CIMCELL\SETUP\MAP.INI`	Contains the TCP/IP configuration of the manager and all device drivers in an OpenCIM cell.

`[RFIDDriverDefinitions]`
`BaudRate=9600`
`Parity=None`
`DataBits=8`
`StopBits=1`
`XonXoff=No`
	RS232 parameters used by the RFID device driver for communication with the RFID reader.

ROBOTVISIONpro Device Driver Settings

`[General]`	
`CimSetupPath== ..\CIMCELL\SETUP\SETUP.CIM`	See this parameter in the ACL section above.
`[RVPDriverDefinitions]`	
`Frame=1`	Frame number as defined on the Frame Definition screen in the Setup menu of the ROBOTVISIONpro software. See the ROBOTVISIONpro documentation for details.
`Snap=No`	For older versions (prior to v2.3) of the ROBOTVISIONpro software, set this value to Yes. For v2.3 and later, set it to No. For best results, use only v2.3 and later.

`BaudRate=9600`
`Parity=None`
`DataBits=8`
`StopBits=1`
`XonXoff=No`
	RS232 parameters used by the QC device driver when it communicates with the controller for the quality control device. You must set these parameters to match the RS232 settings on the device's controller.

`MainWindowBkgndColors=150,150,150` `MainWindowTextColors=255,255,255`	See these parameters in the ACL section above.
`SimulationFailPercent=50`	See this parameter in the Laser Scan Meter section above.

| ```
QCReport=Yes
QCReportTemplateFile=VC2_QC.INI
QCReportFileName=
QCReportFileMarker=
QCReportFileDeleteOnStart=
``` | See these parameters in the Laser Scan Meter section above. |

ViewFlex Device Driver Settings

```
[General]

ScriptPath= project
directory\cimcell\ws3
```

ULS Device Driver Settings

```
[General]

CimSetupPath=..\CIMCELL\SETUP\SETUP.CIM

[Networking]

CimMapPath=..\CIMCELL\SETUP\MAP.INI

[CNCDriverDefinitions]
```

`BaudRate = 9600`
`Parity   = None`
`DataBits = 8`
`StopBits = 1`
`XonXoff  = No`
	RS232 parameters used by the CNC device driver when it downloads G-code to a CNC machine. You must set these parameters to match the RS232 settings on the CNC machine.

```
; --- CNC Variables ---
```

`V1 = ACL71`
`V2 =`
`V3 =`
`V4 =`
`V5 =`
`V6 =`
`M`
`V16 =`
	Parameter variables used by CNC script programs. These variables are used to write portable CNC script programs. For further information, see "ULS Device Driver" in Chapter 8.

`BV0 = 0` `BV1 = 0`	These 8-bit values, which range from 0-255, are assumed to be the initial state of the control lines of a CNC machine when the system is turned on.
`PORT0 = 0x500` `PORT1 = 0X501`	I/O port addresses on a Station Manager PC that are mapped to the status and control lines of a CNC machine using a special interface card. These values should match the jumper settings on the I/O card.

```
[DEBUG]

Protocol=YES
```

PLC Device Driver Settings

`[General]`
`CimSetupPath=..\CIMCELL\SETUP\SETUP.CIM`
`[PLCDriverDefinitions]`
	See this parameter in the ACL section above.

`Type = OMRON`	The type of PLC being used. This value determines what communications protocol is used between the device driver and the PLC.
`SimulationStations = 3,6,1,2,4,5,7`	In Simulation or Manual mode, this parameter sets the order in which the stations appear.
`SimulationPallets = 8`	In Simulation mode or Manual mode, this parameter specifies the number of pallets traveling on the simulated conveyor.

`SimulationPosPerStation = 3,3,3,3,3,3,3`	In Simulation or Manual mode, this parameter specifies the distance between each station as measured in pallet lengths, i.e. the number of pallets that would fit on the simulated conveyor between two stations. The position of each number here corresponds to the order of stations as listed in the parameter `SimulationStations`. For example, the distance between stations 4 and 5 below is 8 pallets long:
	SimulationStations = 3,6,1,2,4,5,7
	SimulationPosPerStation = 3,5,7,4,8,3,5
`SimulationDirection = L`	In Simulation or Manual mode, this parameter specifies the direction in which the simulated conveyor travels.
	L = Clockwise R = Counter-clockwise

`BaudRate = 9600`
`Parity   = Even`
`DataBits = 7`
`StopBits = 2`
`XonXoff  = No`
	RS232 parameters used by this device driver when it communicates with the PLC. You must set these parameters to match the RS232 settings on the PLC.

`MainWindowBkgndColors = 150,0,170` `MainWindowTextColors  = 0,0,0`	See these parameters in the ACL section above.

VC2_WM.DBF

The VC2_WM.DBF (Virtual Controller Series 2, Windows Messages Database File) contains the following fields:

Field	Description
DDE_CHNNL	The type of device driver that sends this message. This value must be one of the following (in lower case): • dde_acl • dde_cnc • dde_plc • dde_cim • dde_rvp • dde_olmt • dde_asrs • dde_lsm
WM_INPUT	The ID of the message to be sent.
NAME_WS	Name of destination workstation that is to receive this message (as specified in the file MAP.INI).
NAME_DDE	The type of device that receives this message (e.g. dde_cim for the CIM Manager).
WM_	This number identifies the type of message being sent.
ID_DEVICE	Device ID of the receiver as specified in the file SETUP.CIM. If a device does not appear in SETUP.CIM, this value will be 0 (e.g. the CIM Manager, the Graphic Tracking Module, an ASRS).
NOTE	A description of this message (free text).

Note
The term DDE stands for MS-Windows Dynamic Data Exchange. OpenCIM device drivers do not currently use DDE but this term remains in the above field names for backward compatibility.

The following table shows the standard OpenCIM messages contained in VC2_WM.DBF:

External CIM Message	Message Description
2225	Robot Start from ACL
2226	Robot Finish from ACL
2230	Robot End from ACL
2335	Pallet Stop from PLC
2336	Pallet Pass from PLC
2337	Error from PLC
2338	Arrive Free Pallet from PLC
2229	Error from ACL
2339	Complex Pass Message from PLC to OLMT
2580	CNC End from CNC
2576	CNC Error from CNC
2581	CNC Start from CNC

External CIM Message	Message Description
2582	CNC Finish from CNC
2138	ACL to CNC Request
2137	ACL to CNC String
2195	QC Result
2358	Device is ready
2583	CNC Task is loaded
2359	Device is unavailable

OpenCIM Database Structure

The OpenCIM database is compatible with the Xbase database management programs (e.g. dBASE, FoxPro and Clipper).

The OpenCIM database consists of the following files:

File	Description
PART_DEF.DBF PART_PRC.DBF	Created by the Part Definition program.
MACHINE.DBF PROCESS.DBF	Created by the Machine Definition program.
TEMPLATE.DBF	Created by the Storage Definition program.
STORAGE.DBF	Created by the CIM Manager and is maintained by the Storage Definition program.
ORDER.DBF	Created by the Order Entry program.
APLAN.DBF	Created by the Order Entry program through the APLAN.EXE program.
CIMREP.DBF LEAFPART.DBF	Created by the CIM Manager.
SCHEDULER.DBF	Created by the CIM Scheduler program
PERFORMANCE.DBF	Created by the performance program
OPT_MQUEUE.DBF	Created by the Optimization program
PURCHASE.DBF CUSTOMER.DBF SUPPLIER.DBF	Created by the Order Entry program

The following tables describe the structure of the files that compose the OpenCIM database.

Fld #	Field Name	Type	Width
PART_DEF.DBF File Structure			
1	PART	Character	20
2	DESCRIPT	Character	40
3	ID	Numeric	4
4	TEMPLATES	Character	15
5	RACKS	Character	15
6	SETUPTIME	Numeric	8
7	LEADTIME	Numeric	8
8	PRODUCT	Logical	1
9	SUPLIED	Logical	1
10	PHANTOM	Logical	1
11	COST	Numeric	6
12	SUPLIER	Character	30

Fld #	Field Name	Type	Width
13	SUPTIME	Numeric	8
14	PCTLOST	Numeric	2
15	MINORDER	Numeric	4
16	CAPACITY	Numeric	2
17	CATNUMBER	Character	30
18	SAFSTOCK	Character	5

PART_PRC.DBF File Structure

1	PART	Character	20
2	SEQNO	Numeric	2
3	SUBPART	Character	20
4	PROCESS	Character	20
5	PARAMETERS	Character	30
6	ORDERING	Logical	1

MACHINE.DBF File Structure

1	SERVER	Character	20
2	COST	Numeric	6
3	NBUFFER	Numeric	1
4	NRACK	Numeric	1
5	NCONVEYOR	Numeric	1
6	NMAX	Numeric	2
7	QUETYPE	Character	4
8	QUEVEC1	Numeric	4
9	QUEVEC2	Numeric	4
10	QUEVEC3	Numeric	4
11	QUEVEC4	Numeric	4
12	QUEVEC5	Numeric	4
13	QUEVEC6	Numeric	4
14	NPRELOAD	Numeric	2
15	PRG1	Character	80
16	DATE1	Date	8
17	PRG2	Character	80
18	DATE2	Date	8
19	PRG3	Character	80
20	DATE3	Date	8
21	PRG4	Character	80

Fld #	Field Name	Type	Width
22	DATE4	Date	8
23	PRG5	Character	80
24	DATE5	Date	8
25	LASTLOADED	Numeric	1

PROCESS.DBF File

1	SERVER	Character	20
2	ACTIONYPE	Character	12
3	PROCESS	Character	20
4	FILE	Character	80
5	PROGRAM	Character	20
6	ARGCNT	Character	20
7	PARAMETERS	Character	40
8	FAILPRCNT	Numeric	2
9	DURATION	Character	8
10	NEED ROBOT	Character	3

TEMPLATE.DBF File Structure

1	BAR_CODE	Character	6

STORAGE.DBF File Structure

1	SERVERID	Numeric	4
2	SERVER	Character	20
3	INDEX	Numeric	3
4	TYPE	Character	1
5	SUBTYPE	Numeric	3
6	STATUS	Numeric	1
7	PARTID	Numeric	4
8	PARTNAME	Character	20
9	PARTPOS	Numeric	4
10	TEMPLTID	Numeric	4
11	TEMPLATE	Character	20
12	TEMPLTPOS	Numeric	4

ORDER.DBF File Structure

1	SEQNO	Numeric	2
2	PART	Character	20
3	ITEMS	Numeric	3
4	FIRSTDO	Numeric	

Fld #	Field Name	Type	Width
5	NEXTDO	Numeric	2
6	PRIORITY	Numeric	2
7	TARGET	Character	20
8	NOTE	Character	40
9	DUEDATE	Date	8
10	DUETIME	Character	8
11	DONE	Numeric	3
12	FAIL	Numeric	3
13	INPROCESS	Numeric	3
14	EXPDATE	Date	8
15	EXTIME	Character	8

APLAN.DBF File Structure

Fld #	Field Name	Type	Width
1	PART	Character	20
2	SEQNO	Numeric	2
3	PROCESS	Character	20
4	SUBPART	Character	20
5	TARGET	Character	20
6	INDEX	Character	20
7	DURATION	Character	8
8	PARAMETER	Character	80

CStOrder.DBF File Structure

Fld #	Field Name	Type	Width
1	CUSTNAME	Character	20
2	PART	Character	20
3	ITEMS	Numeric	3
4	DONE	Numeric	3
5	PRIORITY	Numeric	2
6	DUEDATE	Date	10
7	DUEPERIOD	Character	8
8	MANINDEX	Numeric	8

Customer.DBF File Structure

Fld #	Field Name	Type	Width
1	CUSTNAME	Character	20
2	DESC	Character	80
3	ADDRESS	Character	40
4	PHONE	Character	25
5	FAX	Character	25

Fld #	Field Name	Type	Width
6	EMAIL	Character	25

Orderlist.DBF File Structure

1	SEQNO		2
2	PCONTROL	Character	1
3	DESC	Character	80

PURCHASE.DBF File Structure

1	SUPPLIER	Character	20
2	PART	Character	20
3	SPART	Character	20
4	ITEMS	Numeric	3
5	COST	Numeric	6
6	DUEDATE	Numeric	6
7	SENDDATE	Numeric	6

SUPPLIER.DBF File Structure

1	SUPPLIER	Character	20
2	DESC	Character	80
3	ADDRESS	Character	40
4	PHONE	Character	25
5	FAX	Character	25
6	EMAIL	Character	25

REPORT.DBF File Structure

1	NAME	Character	20
2	REPORTNAME	Character	120
3	DESTINATION	Character	30
4	NOTE	Character	80

SCHEDULER.DBF File Structure

1	PART	Character	20
2	PROCESS	Character	20
3	MACHINE	Character	20
4	ORDERNUMB	Numeric	3
5	PLNSTART	Character	8
6	PLNFINISH	Character	8
7	PLDURATION	Character	8
8	ACTSTART	Character	8
9	ACTFINISH	Character	8

Fld #	Field Name	Type	Width
10	ACTDURATION	Character	8
11	STATUS	Numeric	2

LEAFPART.DBF File Structure

1	ID	Numeric	2
2	NAME	Character	20
3	ORDER_NO	Numeric	3
4	IDLIST	Character	80
5	ACTION_POS	Numeric	3
6	ACTION_SUB	Numeric	3
7	ACTION_TYPE	Numeric	3
8	STATUS	Numeric	1

CIMREP.DBF File Structure

1	CODE	Character	20
2	ORDER	Character	20
3	ORDERSEQ	Character	20
4	PART	Character	20
5	PARTID	Numeric	4
6	DEVICE	Character	20
7	DEVICEID	Numeric	4
8	ACTION	Character	20
9	ACTIONID	Numeric	4
10	SUBPART	Character	20
11	SUBPARTID	Numeric	4
12	WHICYH	Character	20
13	INDEX	Numeric	3
14	STATION	Character	20
15	TIME	Character	8
16	DATE	Date	10
17	DURATION	Character	8

PERFORMANCE.DBF File Structure

1	SERVER	Character	30
2	ORDERID	Character	5
3	TOTALTIME	Character	17
4	PROCTIME	Character	17
5	EFFICIENCY	Character	10

Fld #	Field Name	Type	Width
6	MAXQUELEN	Character	4
7	COST	Character	15
8	SETUPS	Character	4
9	FAILPRCNT	Character	10
10	NOTE	Character	80

OPT_MQUEUE.DBF Structure

1	SERVER	Character	30
2	ALGNAME	Character	30
3	ALGWEIGHT	Character	3
4	ACTIVE	Character	3
5	FIN2FIN	Character	
6	Note	Character	80

Application to Report File Cross Reference

The following table links the specific OpenCIM application to the database file name, the report name and the report template file name.

Application	Database File Name	Report Name	Report Template File Name
Part Definition	PART_DEF.DBF PART_PRC.DBF	Part Report PRocess Report Product Tree	part.rpt proces.rpt partprocess.rpt
Machine Definition	MACHINE.DBF	Machine Definition Report	machine.rpt
Storage Definition (ASRS)	STORAGE.DBF	AS/RS Report	ASRS.rpt
CIM Manager (Created by)	CIMREP.DBF	Analysis Report	Analysis.rpt
CIM Manager and ASRS	STORAGE.DBF	Location Status Report	Location.rpt
MRP	APLAN.DBF	Aplan Report	Aplan.rpt
	ORDER.DBF	Order Report	order.rpt
	PURCHASE.DBF	Purchase report	Purchase.rpt
Performance	Performance*.DBF	Performance Report	performance.rpt
Optimization	OPT_MQUEUE*.DBF	Optimization Settings Report	Optimization.rpt
Report Generator	PART_DEF.DBF		
		Part Definition	Part.rpt

PART_PRC.DBF	Sub. part	Subpart.rpt
PART_PRC.DBF	Process	Process.rpt
MACHINE.DBF	Machine Definition	Machine.rpt
STORAGE.DBF	Storage Definition	asrs.rpt
PURCHASE.DBF	Purchase	purchase.rpt
ORDER.DBF	Manufacturing Order	order.rpt
APLAN.DBF	Aplan	aplan.rpt
CIMREP.DBF	Analysis	analysis.rpt
STORAGE.DBF	Location	location.rpt

Software Backup

Since system files could be altered or destroyed, it is recommended that you keep backup files of your OpenCIM system. The backup files can be used to restore the system if necessary.

The backup procedure involves three stages:

Back up the ACL controllers to the Station Manager Pcs (detailed in Chapter 8).

Back up the Station manager PCs to the CIM Manager PC.

Back up the CIM Manager PC to a backup disk.

❧

Notes

These procedures should be performed by the system supervisor only.

Do not perform Backup procedures while the OpenCIM is running because currently running programs may be aborted and data files may be in an unstable state.

Always keep robot positions, ACL programs and parameters on disk.

Backup and restore the entire system regularly to ensure good backup at all times.

The following procedure backs up a CIM cell directory.

❶
❷
❸
Procedure
Backing up the
OpenCIM System

1. Back up the ACL controllers to a backup folder in each WS*n* folder in the Manager PC.

2. Place all relevant data of a *CIM cell* under its folder in the Manager PC.

3. Compress the CIM cell folder using Winzip.

4. Make copies of the compressed folder to your backup disk

It is recommended that you also make separate backup files of the following items:

Robot Points	Robot point coordinates are associated with the station PC connected to an ACL controller. These points can change whenever a new product is defined, an existing product definition changes, or a robot (or any other device) is moved.
Gcode and process programs files	These files contain the process program of the CNC machines or/and other processing machines.
QC script files	These files contain the Quality Control procedures.

13

Errors and Troubleshooting

This chapter describes how to handle device errors, troubleshooting and so on. It includes the following sections:

- **Device Error Handling**, describes the device error window that appears whenever a problem is detected with a specific machine or a robot as well the information referring to occurring problems.
- **Troubleshooting**, describes various trouble shooting procedures enabling you to identify the problems.
- **Error Messages**, lists the OpenCIM error messages.
- **Contacting Technical Support**, describes how to contact Intelitek or your local distributor for assistance and displays the problem report form.

Device Error Handling

The Device Error screen appears whenever a problem is detected with a specific machine or a robot. This screen allows you to determine how to deal with an error without having to reset the entire CIM.

Figure 13-1: Device Error Screen

Device errors can be caused by various factors, including:

- Robot collision
- Device breakdown
- Defective part which does not properly fit into a machine or robot
- Machine running out of supplies for a given process

The Device Error screen gives you complete information about what was happening at the time that the problem occurred. It identifies the current part that was being processed (current process) when the error occurred. You can then choose what to do in order to recover from the error.

This screen is divided into the following sections:

- Where the problem occurred (top)
- What the problem is (middle)
- How to proceed (bottom)

Where the Problem Occurred

The following table describes information referring to a problem that occurred in OpenCIM. This includes the station where the problem occurred, the part that was being processed when the problem occurred and so on.

Station	The name of the workstation PC where the problem occurred (e.g. `WS 03`).
Part	The ID of the current part that was being processed when the error occurred.
Device	The name of the robot or machine that experienced the problem.

Next Process	The name of the process that was to be performed on the Next Machine (described below). By examining the Process table for the current part, you can determine exactly where the production process was interrupted (select the current part on the Part Definition form).
Order ID	The entry in the Order table that was interrupted by the error.
Next Machine	The name of the next machine that was to process the current part. This information is especially useful when a robot error occurs. The Next Machine field tells you where the robot was supposed to deliver the current part when the error occurred.
Action	The CIM production command (A-plan action) that was being carried out when the error occurred.

What the Problem Is

All fields described in this section are optional. For a given error message, only those fields for which information is available will display.

Error Message	The text of the error message generated by the control program (e.g. robot program, G-code, etc.) that was running when the error occurred.
Error No.	The error code returned by the control program.
Program Name	The name of the program that was running when the error occurred.
Program ID	The ID number of the control program.
Program Line #	The line in the control program that generated the error.
Source Location	The place where the current part resided prior to the error. This field consists of a location ID followed by an index number if appropriate (e.g. a slot number in a rack).
Target Location	The place where the current part was to go next if the error had not occurred. This field consists of a location ID followed by an index number if appropriate (e.g. a slot number in a rack).

How to Proceed

In order to continue operation, two tasks must be accomplished:

- The part should be placed where the next process assumes the part is to be found.
- The proper messages must be sent to the CIM Manager so that it can activate the next process.

In most cases, the safest and easiest way to accomplish these tasks is to:

1. Remove the source of the problem.
2. Cause the device to repeat the operation. This is done by sending the device the appropriate command from the device driver. Then the CIM Manager ignores the reported error because the problem for the device has already been corrected.

Ignore	Process successfully completed; ignore the error and resume production.
	If the current process was successfully completed (with or without help from the operator), clicking the Ignore button causes the CIM Manager to ignore the cause of the error and proceed with normal processing of the current part.

 Caution

Before you select Ignore, make sure that:

- *The current part has not been damaged as a result of the error.*
- *The cause of the error will not recur.*
- *The current part is in the proper position to be handled by the next process.*

Retry   Reserved for future use.

Fail   Reserved for future use.

How to Recover a Failed Device

If a device fails to operate (for example a CNC machine), the CIM-Device Error! screen appears. To recover:

❶
❷
❸
Procedure

Recovering a Failed Device

1. Go to the device that has failed (CNC, Robot, QC or PLC) and abort all programs.
2. Find the problem and correct it.
3. Return to the CIM Manger PC and select "Ignore" in the CIM Device Error! screen. The manager assumes that the last operation was completed and continues on to the next operation.

   Example 1: While running the application, a robot goes into impact protection and the CIM-Device Error! screen appears on your CIM Manger PC.

1. Refer to the Robot user manual and take corresponding actions to abort the current program.
2. If appropriate (depending on specific robot controller type), initialize the controller.
3. Verify that the robot is in free space and then type Run Homes. Wait until the robot has finished homing.
4. Return to the CIM Manger PC and select "Ignore" in the CIM Device Error! screen.
5. According to the type of controller, try to run the last submitted Pick and Place sequence.

   Example 2: While running the application, the CNC fails and the CIM-Device Error! screen appears on your CIM Manger PC.

1. Go to the CNC device driver.
2. Find the problem in the CNC machine and correct it.
3. Load the machine again (manually) with its supplied part.
4. Prepare the machine for Cycle Start.
5. Return to the CIM Manger PC and select "Ignore" in the CIM Device Error! screen
6. Using the CNC device driver, return to the last operation (normally Operate0). Wait until the CNC finishes its G-code.
7. Return manually to the last OpenCIM operation using that device driver.

Troubleshooting

If the installation and startup procedures detailed in your system user manual were closely followed, your OpenCIM system will give you reliable service. If a problem should occur, the first step in the troubleshooting procedure is to identify the problem and its source.

If you encounter problems accessing the Web Viewer, verify that the Internet Explorer Security Settings are set to the default settings.

The OpenCIM system has been designed to simplify troubleshooting procedures by using the CIM-Device Error! dialog box.

When troubleshooting, pay careful attention to the following general warnings:

Warning

- Have all personnel remain clear of the robot envelope, CNC machines, Quality Control machines and all other equipment when power is applied. The problem may be intermittent and sudden unexpected robot or equipment motion could result in injury.

- Have someone ready to operate an emergency "Stop" switch in case it becomes necessary to shut off power to the robot's CNC machines, QC machines, etc.

- Never reach into a machine or robot to actuate a switch because unexpected machine or robot motion could occur, causing injury.

- Remove all electrical power at the main, and turn off all switches before checking electrical connections or any inputs/outputs which could cause robot or machine motion.

There are several cases of alteration that can occur to the OpenCIM programs, including extreme environmental conditions, electromagnetic interference, improper grounding, improper wiring connection and unauthorized tampering. If you suspect the memory of the PC has been altered, scan the disk with the appropriate utility.

Problem	Solution
17. If you receive one of the following messages: General Protection Error... Assertion Failed... An Error has occurred in your application...	• Reset your PC, and perform a disk scan.
18. OpenCIM tells you that it has received an unknown message.	• This is not a real problem. This occurs when someone clicks the mouse on one of the OpenCIM device drivers and activates a procedure.
19. The system is running but the robot Device Driver is not responding to CIM Manager.	ACL: • Verify that the ACL controller is in CON mode. • Verify that ACL Controller-A is in Motors ON. • Try to run the failed command again from the control panel of the ACL device driver. • Verify that your MAP.INI is correct. • Review the robot and controller user manuals, and check if the robot system is operational and is communicating with OpenCIM. Scorbase: • Verify that you have a single Network Connection in the System Network Connections Window. • Verify that your MAP.INI is correct. • Verify that you are working in StandAlone mode. • Verify that your TCP/IP status is enabled. To do so: • Click the TCP/IP icon on the toolbar and select **Enable TCP/IP** if it is disabled.
20. While the system is running, a pallet stops in the wrong destination or does not stop in the correct position.	• Turn the PLC off and then turn it on again. Verify for each pallet, that it stops and releases at each station. • Start the PLC device driver. Place only one pallet on the conveyor and follow the report on the PLC device driver control panel. Verify that the correct pallet ID is reported for each station as the pallet passes. Repeat the same test for each pallet. • Using the control panel, deliver one of the pallets to one of the stations and release it (refer to "The PLC Device Driver" for more details on operating the PLC device driver).

Problem	Solution
21. The system is running but the PLC is not responding.	• Verify that the PLC is on. • Verify that you are able to operate the PLC from the control panel. • Verify that your MAP.INI is correct.
22. The system is running but the RVP is not responding.	• Verify that your MAP.INI is correct. • Verify that the RVP is in the automatic mode and you received the prompt >.
23. The system is running but the CNC does not respond.	• Try to operate the CNC machine from own board. • Verify that the CNC is in the mode designated by your system user's manual. • Try to operate the CNC machine from the CNC device driver. • Verify that your MAP.INI is correct.
24. You try to run the system and the yellow Wait message appears (after waiting there is still no change).	• Verify that your MAP.INI is correct. • Check the OpenCIM Debug dialog box. If "Error 8" is displayed then reboot your PC.
25. One of the OpenCIM applications is unable to locate one of its source files in the setup directory.	• Using the File Manager, verify that the project directory (on the main PC) is a shared directory. • Verify that your PC is connected to the main PC according to your identification in all your local INI files.
26. The barcode fails a good template.	• If your barcode is operated by an ACL controller, verify that in your Part Definition form you entered the following: PROCESS column : READC PARAMETERS column: $TEMPLATETYPE
27. The CIM Manager is running, but gets stuck after the CNC process.	• In the Machine Definition form, verify that the field "List of Preloaded Programs" is not empty.
28. The ACL driver can not establish communication with the robot.	• Verify that the ACL controller is on. • Verify that the motors are on. • Verify that no other application is using the same COM port (e.g. ATS, ACL Off line). • Exit Windows and start the ATS using the correct COM port. If the problem still exists, refer to the ATS manual. • Start Windows and then start the ACL device driver. • If the problem still exists, exit the device driver and verify the COM port in the ACL.INI file.

Problem	Solution
29. The following communication error message appears while operating the robot from a PC with Scorbase: *No communication between Controller-USB and computer*. The Power LED is orange.	• Change to On-line mode. • Make sure the connecting cable is properly connected to the Controller-USB and to the computer. • If the problem persists, replace the USB cable. • Reinstall the USB driver. Refer to the section "USB Driver Installation" below.
30. A device driver is unable to open an RS232 port in order to communicate with its device, and displays the following message in the Control Mode box: `Cannot Open Com:`*`n`*	This error message indicates that the device driver could not open the serial port on the Station Manager PC. Possible causes include: • The port is in use by another application. • The port number is invalid. • One of the serial port parameters is invalid.
31. The ViewFlex device driver can not establish communication with the Manager	• Verify that your MAP.INI is correct. • Verify that script file directory is defined correctly in the VFVD.INI file. • Verify that Viewflex device driver is in On line mode.

Error Messages

Several errors shown in the list below are related to setup problems. The Virtual CIM Setup stores its information in the file SETUP.CIM.

Code	Description and Solution
`9001`	Undefined Part Set up this part using the Part Definition module.
`9002`	Internal Error • Call Intelitek technical support.
`9003`	A `Start Operation` message was received when no operation was requested.
`9004`	A `Finish Operation` message was received when no operation was requested.
`9005`	An `End Operation` message was received when no operation was requested. • Check the ACL program or CNC or Scorbase script associated with the current process that may be sending an erroneous Start, Finish, or End message. OR • Someone has manually triggered a Start, Finish, or End message by running a program from the Control Panel of either the ACL or CNC device driver.
`9006`	Unrecognized device, location, or part. • Define the unrecognized device or location using the Setup module. OR • Define an unrecognized part using the Part Definition module.
`9007`	Cannot perform this process. Either the process definition or device definition is missing. • Define the unrecognized process using the Machine Definition module. OR • Define the unrecognized device using the Setup module.
`9008`	Start message received from an unrecognized device.
`9009`	Finish message received from an unrecognized device.
`9010`	End message received from an unrecognized device.
`9011`	Error message received from an unrecognized device. • Someone has manually triggered a Start, Finish, End or Error message by running a program from the Control Panel of either the ACL or CNC device driver. OR • Check if the ACL program or CNC script that sent the message is using an invalid device ID ($ID). OR • Incorrect assignment of a device to a device driver in the file VC2.MAP. OR • The device ID on the device driver command line is incorrect. OR • Define the unrecognized device using the Setup module. OR
`9012`	This location has not been assigned to a robot. • Use the Setup program to make this assignment.
`9013`	The file SETUP.CIM is missing from the working directory.

Code	Description and Solution
	• Copy a backup version of this file to the working directory. OR
	• Run the Setup module to create a new setup file from scratch.
9014	Unable to transfer this part to its next destination.
	• No path has been defined between the part's current location and its next destination. Use the Setup module to link these two locations. OR
	• Internal Error. Call Intelitek technical support.
9015	Cannot move part because its destination location is already occupied.
	• Internal Error. Call Intelitek technical support.
9016	The robot has received a command to continue an operation that it has not started.
	• Add the Move command to the Part Definition table to have the robot grab the part first.
9017	Invalid Storage Device. A request was received to retrieve a part from a location that is not a storage device.
	• Use the Setup module to define the target device as a storage device. OR
	• Use the Part Definition module to change the target device to be a valid storage device.
9018	This part is not available to the current process.
	Check the Part Definition table.
9019	A quality control result was received when no QC test was requested.
	• Check if an ACL program or CNC script is sending an incorrect message. OR
	• Someone has manually triggered a quality control result by running a program from the Control Panel of either the ACL or CNC device driver.
9020	Reserved for future use.
9021	An unexpected status message was received. There was no corresponding command message sent.
	• Check if an ACL program has assigned an invalid value to the variable $ID (the task ID). OR
	• Check if a CNC program has assigned an invalid value to the variable $ID. OR
	• Device driver internal error. Call Intelitek technical support.
9022	Reserved for future use.
9023	Invalid storage index.
	• Use the Setup module to increase the value of the Capacity field for this storage device. OR
	• Use the Part Definition module to ensure that the storage index specified in the Parameter field of the Part Definition table is within the range of the Capacity field for this device.
9024	Part is not available at this storage location.

Code	Description and Solution
	• Use the Storage Definition module to update the storage contents. OR • Abort this order if there are not enough parts to complete it.
9025	Invalid location index for a machine. • Use the Setup module to increase this machine's part capacity. OR • Internal Error. Call Intelitek technical support.
9026	Undefined process. • Add this process to a suitable machine using the Machine Definition module. OR • Modify the Part Definition table to use a valid process.
9027	No template buffer has been defined for this station. • Use the Setup module to add a buffer.
9028	Process cannot be performed because the machine is not defined in the file SETUP.CIM. • Use the Part Definition module to specify a different process in the Part Definition table. OR • Use the Setup module to define this machine.
9029	Inconsistent value in the inventory database file STORAGE.DBF. • Rebuild the storage data by adding the "/INIT" switch to the CIM Manager command line (CIM.EXE). OR • Check the CIM Manager's INI file (usually OPENCIM.INI) to ensure that the parameter CimDataDir in the [General] section is the same as that in the INI file for the Storage Definition module. • If you want to restore a known good copy of the file STORAGE.DBF, click on the Refresh Storage icon on the Program Manager screen (or manually copy this file from a backup).
9030	Machine not defined in file SETUP.CIM. • Use the Machine Definition module to set up this machine.
9031	The requested G-code task has not been assigned to a CNC machine. • Use the Machine Definition module to assign this task to this CNC machine.
9032	Reserved for future use.
9033	Reserved for future use.
9034	Unexpected ONFAIL process. • Use the Part Definition module to edit the Part Definition table so that ONFAIL only appears immediately after a quality control process.
9035	No ONFAIL process immediately after a quality control test. • Use the Part Definition module to edit the Part Definition table so that ONFAIL appears immediately after this quality control process.
9036	Reserved for future use.

Code	Description and Solution
9037	Error in A-Plan Place command. • Internal Error. Call Intlitek technical support.
9038	Error in A-Plan Next command. • Internal Error. Call Intelitek technical support.
9039	Reserved for future use.
9040	The parameter `ConPallet` (maximum # of pallets) is not defined in the file SETUP.CIM. • Add `ConPallet` assignment to SETUP.CIM.
9041	Cannot start the CIM Manager because it is already running on this PC. • Switch to the window in which the CIM Manager is running.
9042	Reserved for future use.
9043	Invalid status message. Received an unexpected quality control result from a non-QC device. • Check an ACL program or CNC script that might be sending an incorrect message.
9044	Reserved for future use.
9045	Invalid status message. A bar code result was received when no operation was requested. Result ignored. • Someone has manually triggered a bar code result by running a bar code program from an ACL Control Panel. OR • Check an ACL program or CNC script that might be sending an incorrect message.
9046	Machine queue overflow - Too many parts are waiting to use this machine. Use the Order Entry module to decrease the initial quantity ordered for parts that use this machine.
9047	Reserved for future use.
9048	No status message was received after a command was sent because this device driver was reset. • Select how you would like to continue from the options shown on the Device Error screen.
9049	No status message was received after a command was sent because this device driver is not running. • Select how you would like to continue from the options shown on the Device Error screen.
9050	DBF handler error - Request to use an inactive field. • Internal Error. Call Intelitek technical support.
9051	DBF handler error - Record number less than 1. • Internal Error. Call Intelitek technical support.

Contacting Technical Support

If you need to contact Intelitek or your local distributor for assistance, please fill out a photocopy of the *Problem Report Form* below and fax it to us. Contact information can be found at http://www.intelitek.com/contact.asp

To Intelitek - Technical Support Department		Fax:
From: _____		Date: _____
Company: _____	Fax: _____	Tel #: _____

Problem Report Form

Product name: _____

Installed at:_____

Serial numbers of all relevant Intelitek elements: _____

_____

Date of purchase or invoice number: _____

Version number and date of every Intelitek software used (the information appears on the
first screen of every software supplied, and through the ACL command VER regarding
EPROMs):

    software: _____   version: _____   date: _____

    software: _____   version: _____   date: _____

    software: _____   version: _____   date: _____

Detailed descriptions of the problem, including (but not only) all the steps which led up to
the problem arising, since the start-up of the system (attach more pages if necessary):

_____

_____

Error messages as they appear on the display (PC screen, TP LCD, PLC LEDs, etc.):

_____

_____

List all changes introduced to the system since the last time the system worked properly:

_____

_____

For robots: List of accessories and I/Os connected to controller and type of gripper attached
to arm: _____

_____

Attach a printout of all the control parameters.

For computers: Computer type, DOS version and manufacturer:_____

List of cards added (LAN, modem, etc): _____

Attach a printout of the AUTOEXEC.BAT and CONFIG.SYS file.

14

Glossary

This chapter contains the various abbreviations and terminology used in OpenCIM. It includes the following sections:

- **Abbreviations**, contains a list of the acronyms used in OpenCIM, as well as their descriptions.
- **Terminology**, contains a list of the OpenCIM terminology, as well as their descriptions.

Abbreviations

Abbreviation	Explanation
ACK	Acknowledge
ACL	Advanced Control Language
AGV	Autonomous Guided Vehicle
ASRS	Automated Storage and Retrieval System
ATS	Advanced Terminal Software
BMP	BitMaP (the file extension representing the Windows native bitmap picture format)
bps	bits per second
CIM	Computer Integrated Manufacturing
CNC	Computer Numerically Controlled
DBF	Database File (in dBASE format)
DD	Device Driver
FIFO	First In, First Out
FMS	Flexible Manufacturing System
GT	Get part (robot operation)
LAN	Local Area Network
LIFO	Last In, First Out
LSM	Laser Scan Meter
MRP	Material Resource Planning
NACK	Negative Acknowledgment
PLC	Programmable Logic Controller
PP	Pick-and-Place (robot operation)
PT	Put Part (robot operation)
QC	Quality Control
RV	Robot Vision
TCP/IP	Transmission Control Protocol - Internet Protocol
WMF	Windows Meta Format (the file extension representing the Windows native vector picture format)
WS	Workstation

Terminology

Term	Explanation
ACL	Robotic programming language used to control robots and peripheral equipment attached to Intelitek's ACL controllers (Advanced Control Language).
ACL Controller	A multitasking computer used to direct the operations of a robot(s) and peripheral devices in real-time.
ASRS	A robotic storage device used to store and dispense parts in a CIM cell.
Assembly	A part which has been put together from two or more subparts.
ATS	A PC based terminal emulation program used to program an ACL controller (Advanced Terminal Software).
Baud Rate	An RS232 parameter specifying the speed of the serial connection.
Bill of Materials	A structured list of all the materials or parts needed to produce a particular finished product or subpart.
Buffer	A buffer is a tray designed to hold a template when it is removed from the conveyor. It is attached to the outer rim of the conveyor at a station.
CIM Manager	The central control program of OpenCIM. This program directs production in the CIM cell using a variety of communication networks. It also allows the user to set up and define CIM elements.
Com Port	See *RS232*.
Control Program	A program which manages the operation of a CIM device such as a robot (ACL program), a CNC machine (G-code), a camera (ROBOTVISIONpro program), etc. A control program communicates with the OpenCIM system via a device driver at a Station Manager PC. The computer which executes a control program can reside in: A separate controller unit (e.g. an ACL controller) In the device itself (e.g. a CNC machine with embedded controller) A separate PC controlling the device (e.g. a PC attached to a ROBOTVISIONpro camera)
DBF	A file extension indicating "Data Base File" (i.e. in dBASE format).
Device Driver	A program which knows how to communicate with a given piece of equipment that is connected to a PC. It translates commands from other programs into a format understood by the device. It also translates information coming from the device into a format understood by other programs. OpenCIM uses device drivers to communicate with robot controllers, CNC machines, and quality control devices.

Term	Explanation
Download	The act of sending a file(s) from one computer system to another.
Feeder	A device which dispenses parts at station (typically to a robot).
FMS	Flexible Manufacturing System; refers to either a CIM cell or a station in a CIM cell.
Free Movement Zone	A region approximately ½ meter above the work surface in which the robot can move freely and quickly between locations without encountering any obstacles. See also *Pick-and-Place*.
G-Code	A program that directs the operation of a CNC machine. See also *Control Program*.
Group A, B	Used in the context of programming robot positions using ACL. A group refers to a set of axes of movement that apply to a device (e.g. robot, X-Y table). The device can move along all axes in its group simultaneously.
GT	The name of a generic ACL program used to direct a robot to pick up a part at a designated location.
Home a Robot	A procedure used to reset a robot to known starting position.
INI File	A text file containing settings for various OpenCIM parameters. Parameters are grouped into sections. The structure of OpenCIM INI files is similar to that of other standard Windows INI files such as WIN.INI.
I/O (Input / Output)	A low voltage connection used for binary signaling between devices (i.e. on or off). An input is used to read the status of a device. An output is used to turn a designated operation on or off.
Load	An operation which uses a robot to insert a part into a CNC machine.
Loader	A program which automates the start-up of the OpenCIM system under Windows based on command line parameters found in an INI file.
Machine	A CIM device (other than a robot) which performs production processes (e.g. CNC machine, laser scan meter, etc.).
Machine Tending	A device (e.g. a robot) which delivers and retrieves parts from a machine.
Material	See *Part*.
Order	Instructs the CIM system which part(s) to produce and in what quantity.
Pallet	A tray which travels on the conveyor and is designed to carry a template.

Term	Explanation
Part	An entity which moves between stations and machines according to a predefined path, or process. Three types of parts can be defined: supplied, phantom, and final product.
Part Family	A group of parts which are handled the same way by a robot (i.e. the same ACL program can be used to *pick-and-place* parts in the same family).
Pick-and-Place	The primary robot function which involves taking a part from one location (source) and placing it at another location (destination). The pick-and-place strategy minimizes the number of ACL programs required to move parts between two locations at a station. Each location has a GET and PUT program associated with it. The GET program "picks" up a part from the location. The PUT program "places" a part at this location. All GET and PUT programs for a robot are designed to work together to transfer a part from any location to any other location.
PLC	A device having several electrical inputs and outputs. A PLC switches its outputs on and off in response to the state of its inputs and the programming of its embedded computer. In the OpenCIM system, a PLC is used to control the conveyor.
Points	See Position.
Position	The path a robot follows is made up of a set of predefined points. Each point along this path is called a robot position. The coordinates of each point are "taught" by using a *teach pendant* or by running a special ACL program while leading the robot "by the nose" and recording each stopping point along the path.
Process	A production activity (e.g. lathing, milling, assembly, QC check, etc.) performed by a machine on a part.
Processed Material	A part which results from the processing of a raw material.
Product	Something that is manufactured by the CIM cell. The CIM begins production in response to orders placed for products.
PT	The name of a generic ACL program used to direct a robot to place a part at a designated location.
Quality Control	Any process used to check whether a part is satisfactory or not.
Rack	A set of storage compartments used at some stations to store parts either before or after they are processed at that station. Each type of rack is assigned an ID number. Each compartment in a rack is identified by a unique number.
Raw Material	See Supplied Part.

Term	Explanation
RS232	A common, low speed communication protocol which allows a wide variety of devices to communicate with each other (typically in the range of 300 - 19,200 bps). On a PC, RS232 ports are referred to as COM1 - COM4.
Robot	A device that moves parts from place to place at a station. Some robots are also capable of assembling parts.
Robot Vision	A quality control device which optically scans a part to determine if it is satisfactory.
Serial Port	See *RS232*.
Slidebase	A peripheral device which enlarges the working envelope of a robot by allowing it to move along a rail. The slide base gives the robot an additional degree of freedom.
Station	A location adjacent to the CIM conveyor which contains production and/or storage equipment.
Subpart	A part which undergoes some sort of processing in order to be included in a higher level part.
Supplied Part	A part which is the starting point for making a product. This part (or material) is purchased and inserted into a CIM storage location. It will later be processed by the CIM cell.
Template	Plastic trays which can hold various types of parts. They allow parts to be transported on the conveyor.
Unload	An operation which uses a robot to remove a part from a CNC machine.
Working Envelope	The entire area in which a robot can reach.
Xbase	Any database management program that is compatible with the dBASE standard for file formats and commands.

15

Intelitek Software Licensing

The software is protected by a licensing agreement.

There are 2 license types: permanent and subscription (rent license).

If you purchase a permanent license you will receive a CD key from Intelitek.

If you purchase a subscription license you will receive a Subscription CD key from Intelitek.

If the software is installed without a license, you can use the fully operational software for a 14-day evaluation period. To continue using the software after this period it must be unlocked.

The following sections contain detailed instructions on how to use the software license:

- Register your software and receive a PC-specific unlock code for each license purchased.
- Protect your license.
- Transfer a license from one PC to another PC for permanent license type only.
- Return the permanent license to Intelitek, so you can retrieve it later.
- Frequently asked questions.

To register your Intelitek software follow these easy steps:

1. Install the software.

To register the software on Windows Vista/7, click on the Project Manager icon and select Run as Administrator.

The first time you run the software, the Intelitek Software License dialog box appears.

Figure 15-1: Registration Dialog Box

2. Register using the method most convenient for you. We recommend registering online, which is automatic.

To register a subscription license, select **Subscription License**, for the subscription CD key fields to appear.

Figure 15-2: Subscription CD Key Fields

Registration Methods

The following descriptions of the different registration methods are relevant for both permanent and subscription licensing.

The Intelitek Software License dialog box allows for three methods of registration:

- Automatically from the Intelitek Website.
- By E-mail
- Manual Unlock

Automatically from the Intelitek Website.

Note *You must have internet access available on the host PC.*

1. In the Intelitek Software License dialog box, select **Online**. Enter your CD key in the appropriate fields. Your CD key can be found on the back of the CD case. It was also e-mailed to you or sent to your team's main contact (for FIRST teams).
2. Click **Unlock**.

The software will automatically connect to Intelitek's website. An unlock code will automatically be installed on your PC and a message will appear saying that the software is now licensed.

By E-mail

1. In the Intelitek Software License dialog box, select E-mail as your registration preference. Enter your CD key in the field below. Your CD key should be found on the back of your CD case, and was either E-mailed to you or sent to your team's main contact (for FIRST teams).
2. If E-mail is available on the host PC, a new email containing the required details will open.

Figure 15-3: Intelitek Software License E-mail

Fill in all the requested user information, and send the message. The licensing service will send back an unlock code, at which point you will complete a manual unlock (the third registration option described below). **This process is not automated and can take up to several days.**

3. If you have E-mail service, but not on the same computer on which the software is installed, a Notepad window containing all required details will open. Fill in all the requested user information, and then transfer the text/file to your email program. Be sure you have included your PC specific code, and CD key in the email!

Send to: info@intelitek.com

Subject: Intelitek Software License

Note: To ensure successful processing, use this exact subject line and do not edit the automatically generated headings in the message. You may add your personal information and comments into the body of the message.

Once you receive the unlock code, enter it in the Registration dialog box and select Unlock.

Manual Unlock

Manual Unlock registers your software without a CD Key by matching an Unlock Code to your PC Specific Code. You can obtain an Unlock Code from Intelitek if you register by email, or directly from another user after a license transfer.

1. In the Intelitek Software License dialog box, select Manual.
2. Enter the 8 digit alphanumeric Unlock Code in the field below. Click Unlock.

If the Unlock Code matches your PC Specific Code, your software will be licensed.

This completes the software registration procedure.

Permanent Licenses

Protect your permanent license

Every unlock code is unique. It will become invalid (and cause the software to stop functioning) when you change a physical component in your PC (e.g., hard disk, network card, CPU), format your disk, or install a new operating system.

1. If you want to upgrade your PC and keep your software operational, **you must first transfer the license (unlock code) to another PC.**

2. Once you have upgraded your system, reinstall the software (if necessary) and transfer the license back.

If you do not have a PC available for the temporary transfer operation, **return the license to Intelitek.** You will be able to retrieve the license by following the standard procedure for obtaining the unlock code.

Transfer a permanent license from one PC (source) to another PC (target)

1. On the target PC install the software and get the PC-specific code from the Registration dialog box.

2. On the source PC, open the Registration dialog box. Enter the PC-specific code of the target PC and Select Transfer. The software on the source PC will generate a new unlock code for the target PC and will remove the license from the source PC.

3. On the target PC, enter the new unlock code in the Registration dialog box.

Return a permanent license to Intelitek, so you can retrieve it later

Use this procedure when you need to remove a software license and do not have a target PC available.

1. From the Registration dialog box, select Remove the License and click Remove. The software will generate a unique Remove code.

2. Send the Remove code and your CD key to Intelitek using one of the methods described above (email, website, fax/phone). We will confirm the codes and update our licensing registration records.

3. When you are ready to retrieve your license, install the software (if necessary) and follow the instructions for obtaining an unlock code.

4. The installation procedure generates a **PC-specific code**. This code is found in the Registration dialog box.

To receive the unlock code for the software you installed, you must send Intelitek both the CD key and the PC-specific code. The Registration dialog box provides several methods for obtaining the unlock code.

FAQs – Frequently Asked Questions

What is a CD key?

This is the code on a label on the CD. It allows Intelitek to track software that has been purchased.

What do I do if I do not have a CD key?

When prompted to enter the CD key during the software installation, enter the word "evaluation". This will allow you to install the software for a trial period.

What is a PC-specific code?

This is a code generated by the software. It is unique for each PC and each installation of the software. This code allows Intelitek to generate the unlock code for the PC on which you installed the software. The PC-specific code is displayed in the Registration dialog box.

What is an unlock code?

This is a code that allows you to use the software after the evaluation period expires. You need to send your CD key and PC-specific code to Intelitek. We will reply with the unlock code for the software you purchased.

How do I install and register the software on more than one PC?

Repeat the procedure for obtaining an unlock code as many times as necessary.

Alternately, install the software on all PCs and make a note of the PC-specific code generated on each PC. You can then send us one email or fax listing all the PC-specific codes. You will receive unlock codes for each PC. (Note: this will be handled manually by our technical support and may take several days).

Why should I give you my personal details when I request the unlock code?

This will allow us to keep you informed about products, upgrades and services available for your system and software. It will also allow us to help you in case of a lost license.

How can I recover the unlock code after a disk crash or other system failure?

Once you have restored and reactivated your PC, reinstall the software. If it resumes operation in Evaluation mode, follow the procedure for obtaining an unlock code. Include a comment explaining why you need a new unlock code. (Note: this will be handled manually by our technical support and may take several days).

How can I extend the evaluation period?

To continue using the software after the evaluation period has expired, either a subscription (rent license) or a permanent license must be purchased.