The CODE Product Family

Getting Started
With CODE
for Windows NT

Cimetrix®
INCORPORATED

6979 South High Tech Drive
Salt Lake City, UT 84047-3757
(801) 256-6500  FAX (801) 256-6510
This document was written to accompany CODE version 4.1. The information contained in this document is subject to change without notice. Slight variations may occur on computers from varying manufacturers. Every effort has been made to supply complete and accurate information. However, Cimetrix Inc. makes no warranty of any kind with regard to this manual, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose. Cimetrix shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this manual.

Copyright © 1990 - 1999 Cimetrix Inc.

This document is the property of Cimetrix Inc. and is protected by copyright. The information contained herein is proprietary and may not be copied, transferred, or disclosed to others without prior written authorization from Cimetrix Inc., Salt Lake City, Utah 84047-3757.

Portions of the CODE System are protected by US Patent 4,831,549.

Trademark Acknowledgments

Terms mentioned in this document that are known to be trademarks are listed below. Cimetrix Inc. cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark.

Cimetrix is a registered trademark of Cimetrix, Inc.
The Cimetrix logo, CODE, CIMServer, CIMulation, CIMControl, CIMTools, CODE API, CIMCal, CIMApp and CIMTeach are trademarks of Cimetrix Inc.
80386, 80486, and i486 are trademarks of Intel.
PostScript is a registered trademark of Adobe Systems, Inc.
Venture Com
PMAC
MEI
All other products are trademarks or registered trademarks of their respective companies.
# TABLE OF CONTENTS

## Introduction ................................................................................................................... 1-1
- About this Manual ........................................................................................................ 1-1

## Package Contents....................................................................................................... 1-2
- Software .................................................................................................................. 1-2
- Documentation ....................................................................................................... 1-2

## Installation ................................................................................................................... 2-1
- System Requirements .............................................................................................. 2-1
  - Installing CODE .................................................................................................. 2-2
    - Upgrading CODE Programs ................................................................................. 2-2
    - Installing CIMulation ......................................................................................... 2-2
      - Installing CIMulation from CD-ROM ............................................................. 2-3
      - Installing CIMulation from the Cimetrix FTP Site ............................................. 2-3
    - Installing CIMControl ...................................................................................... 2-3
      - Installing CIMControl from CD-ROM .......................................................... 2-3
      - Installing CIMControl from the Cimetrix FTP Site ........................................ 2-4
      - Installing the RTX Version of CIMControl ................................................... 2-4
  - License Codes .................................................................................................... 2-4
    - Acquiring License and Product Codes ............................................................ 2-4
    - Adding License and Product Codes ................................................................. 2-6
- Setting Up a New User Account .............................................................................. 2-7

## CODE Basics ............................................................................................................... 3-1
- What is CODE? ....................................................................................................... 3-1
- C .......................................................................................................................... 3-2
- The CIMServer .................................................................................................... 3-3
  - Connecting to the CIMServer .......................................................................... 3-4
- CIMTools ................................................................................................................. 3-4
  - Starting CIMTools ............................................................................................ 3-4
- Interfaces ................................................................................................................. 3-5
  - CIMulation Graphics Window ........................................................................... 3-5
    - The Tools Menu ............................................................................................. 3-6
    - The Tree Node Browser ................................................................................. 3-6
Quick Tutorial ................................................................. 4-1

The Model ........................................................................... 4-1
Building Your Model ......................................................... 4-2
Open a New Workcell ......................................................... 4-2
Creating Nodes ................................................................. 4-3
Assigning Attributes and Parameters to Nodes ................. 4-6
Joint Parameters ............................................................... 4-8
"Tcf" Parameters ............................................................... 4-12
Geometry Parameters ....................................................... 4-12
"Robot" Parameters ........................................................... 4-15
Saving Your Workcell ....................................................... 4-17
Running Your Robot Model Using CIMTools .................... 4-18

Programming Your Model .................................................. 5-1

Setting Up the Development Studio ................................. 5-1
New Development Environment ....................................... 5-1
Project Settings ............................................................... 5-2

The Program ........................................................................ 5-3
Checking for Errors in Your Program ............................... 5-5

Running Your Robot ......................................................... 5-5
Using the Teach Pendant .................................................... 5-5
Executing the Program ...................................................... 5-7

Conclusion ........................................................................... 5-7
Chapter 1

Introduction

About this Manual

Getting Started With CODE is the first manual in the CODE documentation set that you should read. It is designed to introduce you to CODE and to help you get CODE up and running.

If you have not yet installed CODE, begin with Chapter 1 and read through the entire manual. However, if CODE has already been installed, skip the installation information and continue with the rest of the information in the manual.

NOTE: It is assumed that you are familiar with your computer and its operating system. If you need additional assistance, consult your computer’s documentation or your system administrator.

After completing this manual, you should be able to do the following:

• Install the CODE product
• Load and run a workcell in simulation
• Use the integrated viewing tools in CIMulation
• Model a very simple robot with CIMTools
• Know where to find more information about CODE
Package Contents

Before installing CODE, please check to see that the package contains the following:

Software

The CODE software package should come in one of two media formats, either a CD or a self-extracting executable file from the Cimetrix FTP site, both of which contain the CODE executable files, Visual C++ Project files, and associated include files and libraries.

Documentation

The documentation shipped with CODE should include the following manuals and online help:

- Getting Started With CODE (this manual)
- CIMTools Reference
- CODE Applications Programming
- CIMTeach Reference Manual
- CIMApp Reference Manual
- Customizing CODE
- Cimetrix IDE Ready Help

The Cimetrix documentation is created in .PDF format and is built into the self-extracting executable file found in <installation directory>\Cimetrix\doc on both the CD and the Cimetrix FTP site. The Cimetrix API library is also found in helpfile(.hlp) format in this same directory.
Chapter 2

Installation

System Requirements

In order to install and run CODE, the following minimum requirements must be met:

- Intel x86/Pentium with Windows NT 4.0
- 16 MB RAM
- 30 MB free hard disk space
- One Ethernet network adapter card
- CD-ROM Drive (to run CODE from the installation CD)
- TCP/IP networking protocol installed and configured on your computer. To check to see if TCP/IP is installed on your computer:

1. Open the network control panel.
2. An entry labeled "TCP/IP protocol" will be present in the Installed Network Software list box if TCP/IP has been installed.

NOTE: If TCP/IP is not installed on your computer, please refer to your Windows NT installation guide or consult your System Administrator for more information.
Installing CODE

The instructions that follow explain the basic steps for upgrading or installing CIMulation or CIMControl for the first time from the CD or the Cimetrix FTP site. Carefully note the following guidelines before installing CODE products:

- All CODE products will normally be installed into the same directory
- The default installation directory will be C:\Cimetrix
- You must have write permission for any directory into which CODE will be installed.
- You must install the software from the administrator account or from an account which has installation privileges.

Upgrading CODE Programs

If you are upgrading to CODE 4.1 from a previous version of CODE, you must first uninstall all previously installed CODE programs before installing the new version of CODE.

**Warning!** If you have CODE 3.6.1 programs installed, you should **not** uninstall CODEUtils 3.6.1 or it will remove core DLLs that will cause your operating system to cease functioning.

To uninstall CODE:

1. Click **My Computer** from your desktop and select **Control Panel**.
2. This will display the **Control Panel** box. Select **Add/Remove Programs**.
3. The **Add/Remove Programs Properties** list box will be displayed.
4. Select all CODE applications (one at a time) and click **Add/Remove**.
5. The selected applications will be removed.

**NOTE:** Make sure you uninstall all previously installed versions of CODE.

After CODE has been properly uninstalled, you can then install the upgrade from CD or the Cimetrix FTP site as discussed in the following sections.

Installing CIMulation

You may install CIMulation either from a CD-ROM or from a self-extracting executable downloaded from the Cimetrix FTP site.
Installing CIMulation from CD-ROM

To install CIMulation using the installation CD:

1. Insert the installation CD into the CD-ROM drive (usually the D: drive)

2. Select the Start | Run program, which reads:

   D:\CIMulation4.1.0

Follow the Installation Wizard’s instructions, which will lead you through the installation by using a number of graphical dialog boxes requesting acceptance of the License Agreement and the default installation directory. When the program begins copying files to the hard disk, a status bar will show the percent of the completed installation.

After installation is complete, the program will ask you if you would like to install the license codes. You must install these licenses before the Cimetrix products will run! A later section describes the process for setting up license codes. Finally, you must restart your computer to enable your software. If you do not, the Cimetrix software will not function properly.

Installing CIMulation from the Cimetrix FTP Site

It is possible to download CIMulation from the Cimetrix FTP site as a self-extracting executable. This type of install is used for emergency purposes only. For further information, please see your sales representative.

Installing CIMControl

Installing CIMControl from CD-ROM

To install CIMControl using the installation CD:

1. Insert the installation CD into the CD-ROM drive (usually the D: drive)

2. Select the Start | Run program, which reads:

   D:\CIMControl4.1.0

Follow the Installation Wizard’s instructions, which will lead you through the installation by using a number of graphical dialog boxes requesting acceptance of the License Agreement and the default installation directory.

In addition, the default setting is Typical, which installs CIMControl for the following configurations:

- MEI/XMP card support on Windows NT
NOTE: Windows NT also supports MEI/DSP and PMAC cards.

- PCL722
- Pamux and DeviceNet I/O cards support on Windows NT

When the installation begins, a status bar will show what percent of the process is complete. The names of the files that are being copied to your hard disk will appear above the status bar. Installation may take several minutes.

After installation is complete, the program will ask you if you would like to install the license codes. **You must install these licenses before the Cimetrix products will run!** A later section describes the process for setting up license codes. Finally, you must restart your computer to enable the Cimetrix software. If you do not, then it will not function properly.

### Installing CIMControl from the Cimetrix FTP Site

It is possible to download CIMulation from the Cimetrix FTP site as a self-extracting executable. This type of install is used for emergency purposes only. For further information, please see your sales representative.

**Installing the RTX Version of CIMControl**

RTX is VentureCom’s Real time extension product for Windows NT. To install this version of CIMControl, select **Custom** in the installation process, then select **Real time extension** on the following screen and continue with the installation.

### License Codes

Authorization to use Cimetrix software is given through license codes which are distributed by Cimetrix support. The following sections show how to acquire and add these codes so that you can run Cimetrix applications.

### Acquiring License and Product Codes

Before you can run CODE applications, you must acquire the necessary license codes from Cimetrix.

1. Click on **Start | Programs | Cimetrix | License Manager**. The following Window will appear on your screen:
The **Security ID** information shows two things:

- The name of the computer on which you are running the license manager (in the caption it is named SCRIPT).
- The security ID number (e.g. 0000c0b50ddb, but your ID number will be different). This number is a 12-digit code which uniquely identifies your computer. You will need this number when contacting Cimetrix in order to get licenses for your CODE products.

2. Contact Cimetrix Customer Support by any of the following methods to get the license codes:

   - **Main Telephone:** (801) 256-6500
   - **Support Telephone:** (800) 344-7292
   - **Fax:** (801) 256-6510
   - **E-mail:** support@cimetrix.com

You will be required to provide the support representative with the security ID number obtained from the Cimetrix License Manager as shown in Figure 2-1. The Product Code (each product has a code that is specific to that product) and License Code will be sent immediately. Any product or license codes you receive should be recorded here and kept in a secure location:

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Product Code</th>
<th>Product License</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Cimetrix License Manager Interface](image)

**Figure 2-1: Cimetrix License Manager Interface**
Adding License and Product Codes

Once you have received the license codes and product codes from Cimetrix, as shown in the above section, the codes must be must be added to the Cimetrix License Manager.

1. Click on the Add button of the Cimetrix License Manager. The following window will appear:

![Figure 2-2: The Add License Box](image)

2. In the Product field, enter the type of product that you want the license to enable, for example, CIMControl, CIMLicense, or CIMulation. Note that if you already have a license installed and you select the product, then the name and product code will automatically be filled in and greyed out in the corresponding areas.

   **NOTE:** The options listed in this field are licenses that have been installed already. If you have never installed any licenses before, the only option available will be New Product.

3. Enter the name of the product in the Name field, for example, CIMControl 4.0.1.

4. Enter the product code. This 8-digit code is generated by Cimetrix and is used to identify which product is being used. Note that this code is not case sensitive.

5. Enter the license code. This 12-digit number is generated by Cimetrix and authorizes use of the software. Note that this code is not case sensitive. Note that a single license can incorporate multiple licenses. For example, if you purchased 10 copies of CIMulation, you might setup all of the licenses on one computer. A single 12-digit license code would in this case could incorporate all 10 CIMulation licenses.
6. Click on the **OK**, **Apply**, or **Cancel** button.

   **OK**  Accepts the current license code and closes the *Add License* box.

   **Apply**  Accepts the current license code and leaves the *Add License* box open to add more license codes.

   **Cancel**  Closes the *Add License* box.

7. The License Manager checks to see if the information entered is valid. You will be notified immediately if the information is incorrect.

   **NOTE:** If you select **Apply** and want to add more licenses, repeat steps 2-6 for each additional product license.

---

### Setting Up a New User Account

To set up a new user account that allows you to run Cimetrix software:

1. Click on **Start** | **Program** | **Cimetrix** | **New User**.

   **NOTE:** If you already have an account set up, you will be immediately notified.

2. The `<installation directory>`\CimetrixInit file will be copied to your home directory. Note that the `<installation directory>` default is C:\Cimetrix. This must be done prior to running the CODE software.

   **NOTE:** If you do not have a home directory, please see your Systems Administrator. The home directory is setup in the Windows NT User Manager program.
There are some basics about CODE that must be understood before proceeding to the tutorial on building workcells in the next section. These elements include:

- a definition of CODE
- a definition of the CIMServer and how to connect to it
- a definition of the default user interface, CIMTools, and how to start it up
- a review of the Graphics Window Interface
- a review of the Tree Node Browser (CIMTools) interface

What is CODE?
CODE is an acronym which stands for Cimetrix Open Development Environment. It is made up of a suite of software tools which aid in the conceptualization, design, simulation, testing, debugging and control of an automated workcell. This suite consists of the following:

**CIMulation**
Allows you to simulate automated manufacturing tasks using your PC computer. Also known as CIMServer.

**CIMControl**
Allows you to run the actual (non-simulated) manufacturing equipment using your PC computer. Also known as CIMServer.

**CODE API**
This is a library of Application Programming Interface functions (a named block of code that can be called within a program) which
relates to motion control (for Cimetrix specifically). For more information on CODE APIs, see CODE API Reference Manual- Volume 1, 2, or 3.

**CIMTools**

Allows you to create a simulated workcell from the CIMTools interface. You can then view it in motion by using the Teach Pendant and check for collision possibilities. For more information, see CIMTools Reference Manual for Windows NT.

**Common Terms**

Many of the terms commonly found in the Cimetrix documentation may not be familiar to the first time user. Some of the common terms found in constructing a workcell include:

**Workcell**

A representation of the physical model. The workcell is made up of a database (nodes) that represents various parts located in positions that relate to each other and/or to the world. These "nodes" contain geometric information of the various components and can be accessed in both simulation and online modes.

**Node**

A placeholder or frame that graphically represents anything in the workcell. Each node frame has position and orientation relative to a parent node frame. Each node frame may have attributes and children node frames.

**Tree**

Hierarchy set up much the same as an actual tree, wherein a single node serves as the trunk, while other nodes act much the same as branches or twigs. These nodes are set up very similar to the parent-child relationship, so they are denoted "parent" or "child" nodes.

**World Node**

The highest node in the tree hierarchy. It is a node that is common to all other nodes in the workcell. It is always defined--an empty workcell only includes the world node definition.

**Attributes**

Properties of a node that determine how it will behave in a workcell. A few attributes are described as follows:

**Robot**

User can assign properties of a robot, such as speed, controller types and kinematics.
Joint
This attribute allows the user to set speeds and limits, and define whether the joint node is rotational (moving around its center axis) or translational (moving along an axis). Because of the tree hierarchy, a node with this attribute must be a descendant of the robot node.

Geometry
A node with this attribute allows you to assign a type to it, such as box or cylinder, and a size. The workcell can function without geometry nodes, but it is easier to avoid collisions if the user can see accurate dimensions for the components in the workcell model.

Pose
The position and orientation of a node frame in space.

Position and Orientation
This specifies the relationship of a node to its parent. Position refers to the x-, y-, and z-coordinates. Orientation refers to its rotation around the x-, y-, and z-axes.

Clients
Programs which perform their function by sending requests to the CIMServer.

CIMServer
CIMulation or CIMControl (See The CIMServer section in this chapter).

The CIMServer
The CIMServer, or server, is the central control program for CODE, and performs the following basic tasks:

- Receives requests from other programs to perform operations.
- Builds and maintains a model of the workcell.
- Supplies motion planning and trajectory generation

There are two types of CIMServers:

<table>
<thead>
<tr>
<th>CIMulation</th>
<th>Allows you to simulate an automated manufacturing task using your PC computer. Basically, it means you can build a graphical representation of your system before building the actual robot.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIMControl</td>
<td>Allows you to run (from your PC) the actual manufacturing equipment after you have built it (i.e. it provides servo control and I/O operations).</td>
</tr>
</tbody>
</table>
Connecting to the CIMServer

After you have properly installed CODE, run the license manager to install the license, and rebooted your system, you are ready to start the CIMServer. In this example, CIMulation is used, however, the use of CIMControl is directly analogous. To connect to the CIMServer:

1. Login to your system.

2. Start CIMulation by selecting Start | Programs | Cimetrix | CIMulation, or type the following into a DOS command line window:

   cimulation -server myServerName

3. Wait for the server to completely start. The Graphics Window will appear with a default workcell when the server is ready.

CIMTools

The information contained herein regarding CIMTools just covers the basics of the software package. For more information on the full capabilities of CIMTools, please see CIMTools Reference Manual for Windows NT.

Starting CIMTools

After connecting to the CIMServer, you must start CIMTools in order to create your workcell (we will create a workcell in the tutorial in the next section).

To start CIMTools:

1. Select Start | Programs | Cimetrix | CIMTools, or type the following command into a DOS command line window:

   cimtools -server myServerName

Either of these operations will start CIMTools.
Interfaces

When building a workcell, you will see and work with two different graphical user interfaces (GUI) simultaneously. The first interface that you will see after connecting to the CIMServer is the CIMulation interface, also known as the Graphics Window. The second interface is displayed when CIMTools is opened, and is known as the Tree Node Browser. Note that the CIMulation and CIMTools GUI are separate processes. CIMTools uses the CODE API functions.

CIMulation Graphics Window

The CIMulation Graphics Window will be displayed when you connect to the CIMServer.

Figure 3-1: CIMulation Graphics Window
This window gives shape to your workcell. As you create and give characteristics to the nodes that make up your robot in the Tree Node Browser, geometric figures may be created in the Graphics Window. The two windows work together simultaneously.

**The Tools Menu**

The following menu is available when viewing a workcell in the Graphics Window:

- Sets graphic back to default view
- Click and drag model in the Graphics Window
- Click and rotate model in the Graphics Window
- Zooms out
- Selects desired nodes in the window
- Enlarges a selected part of the graphic
- Moves graphic left
- Moves graphic right
- Moves graphic down
- Moves graphic up
- Rotates graphic toward back
- Rotates graphic toward front
- Rotates graphic toward left
- Rotates graphic toward right
- Rotates graphic downside-up to left
- Rotates graphic upside-down to right
- Zooms graphic out
- Zooms graphic in

You can hide or show this menu by clicking **Tools** and **Hide** or **Show** from the menu bar in the Graphics Window.

**The Tree Node Browser**

From the main CIMTools interface, also known as the Tree Node Browser (or Browser for short) you can accomplish the following tasks:

- create nodes
- give attributes to nodes such as:
  - robot
  - joint
  - geometry
- setup your automated workcell
- check for collisions
- delete nodes
These operations, in addition to many others, will enable you to setup an automated workcell.

From this Browser you can also open previously created workcells, save new workcells, cut and paste nodes or work on node branches. You also get a general sense of node relations one to another and to the world node (for more information on this node, see the section entitled Common Terms in this manual).
Chapter 4

Quick Tutorial

The information given to this point in the manual has shown what you will see when installing CODE, connecting to the CIMServer and opening CIMTools. So, what’s next? The tutorial in this section will teach you how to build a workcell and control it with a program (Visual C++ console application).

The Model

The model we will be building in this tutorial will be a basic four-axis model like the following:

Figure 4-1: Tutorial Model
Building Your Model

Before you can build a model, you must first connect to the CIMServer and open CIMTools (see Connecting to the CIMServer and Starting CIMTools).

After connecting to the CIMServer and starting CIMTools, the following sections must be followed in order to build your model.

Open a New Workcell

When you connect to CIMServer for the first time, the default workcell, Peg.w, will be displayed (the Peg.w file is displayed in Figure 3). To create a new workcell:

1. Click the New Workcell icon or select File | New in CIMTools (a.k.a. the Tree Node Browser). A dialog box asking if you want to save the open workcell will be displayed. Click Yes to save before opening a new workcell, No to open a new workcell without saving the open workcell, or Cancel to perform neither action.

2. Opening a new workcell will display the following in the Tree Node Browser:

![Figure 4-2: New Workcell](image)
Notice the **world** node displayed. This node is the main ancestor of all other nodes we will create in the workcell. It is the default node that connects the rest of your nodes.

Also, when you open a new workcell, CIMulation (the Graphics Window) should become a blank slate as shown below.

![New Graphics Window](image)

**Figure 4-3: New Graphics Window**

**Creating Nodes**

As was previously stated in the *Common Terms* section of this manual, a node is a placeholder that graphically represent basically anything in the workcell. Nodes are represented in the Graphics Window with a triad which looks like this:

![Triad](image)

This triad represents a specific node.
To create nodes for this workcell:

1. Make sure the **world** node is selected in the Tree Node Browser (it will be highlighted in blue if it is).

2. Select **Edit | Add Node**, and the **New Node** box will be displayed.

   ![New Node Box](image)

   **Figure 4-4: The New Node Box**

3. Type “**Robot**” in the **Enter Name** text field.

4. Click **Apply**.

   It is important to remember that you must always create a **Robot** node in an automated workcell for each mechanism. It does not always have to be the child of **world** as shown here. The name is arbitrary. However, this node must be inserted into the Tree Node Browser when you are creating a system using servo controllers like the one we are creating here.

   Notice that the **Robot** node will appear as a "child" of **world** because **world** was highlighted when you created **Robot**. Therefore, remember that the node being created will be a child of whichever node you have highlighted in the Tree Node Browser.

With that in mind, go ahead and create the following nodes:

- **Axis1** as a child of **Robot**
- **Base** as a child of **Robot**
• **Axis2** as a child of **Axis1**
• **Arm1** as a child of **Axis1**
• **Axis3** as a child of **Axis2**
• **Arm2** as a child of **Axis2**
• **Axis4** as a child of **Axis3**
• **Arm3** as a child of **Axis3**
• **Tcf** as a child of **Axis4**
• **Arm4** as a child of **Axis4**

When you are done, the Tree Node Browser should appear as so:

![Browser with Nodes Created](image)

*Figure 4-5: Browser with Nodes Created*
NOTE: Nodes in the Browser are relative to one another, so if you adjust the position or orientation of a parent node, the child nodes will appear to move with it. A child’s position and orientation relative to its parent remain constant when the parent is adjusted. Take a teapot, for example, with the handle being the parent, and the actual teapot being the child. If you move the handle, the teapot will move relative to the handle. The teapot’s relationship to the handle does not change.

Assigning Attributes and Parameters to Nodes

After creating the nodes being used in this workcell, we may assign attributes to those nodes, or in other words, give additional properties to the nodes. What types of nodes are they? Robot, joint, geometry? Assigning attributes also means giving properties to the nodes. Note that a node is not required to have any additional attributes. Frequently, a node will be used as a target frame for motion, or a parent node for organizing other nodes.

To assign attributes to the nodes:

1. Go to the Tree Node Browser and click on the node you want to assign attributes to. In this workcell, the first node we will assign attributes to will be the Robot node. Therefore, click on Robot in the Browser.

   NOTE: The World node does not need any attributes assigned to it.

2. Click on the Node Inspector icon, and the Node Inspector will be displayed:
3. On the Attributes tab, you will see a number of attributes. Click on Robot. This will assign the Robot attributes to the node named Robot. The purpose of this operation is to identify that this robot uses servo control. Each robot is independently servo-controlled when running with CIMControl.

Although we have already assigned the Robot attribute to the Robot node, there are still ten more nodes that need attributes assigned to them. By following the steps listed above for each node, assign the following attributes to each node:

- **Axis1** node- **Joint** attribute
- **Axis2** node- **Joint** attribute
- **Axis3** node- **Joint** attribute
- **Axis4** node- **Joint** attribute
- **Tcf** node- **Tcf** attribute
- **Base** node- **Geometry** attribute
- **Arm1** node- **Geometry** attribute
- **Arm2** node- **Geometry** attribute
**Arm3** node- **Geometry** attribute

**Arm4** node- **Geometry** attribute

A good way of telling that your nodes have been assigned attributes is to look at the circle next to the name of the node in the Tree Node Browser. **Robot** nodes have a red circle, **Geometry** nodes have a green circle, and **Joint** nodes have a blue circle.

Once you have created the nodes in the Tree Node Browser and the corresponding Triads in the Graphics Window, it is time to give graphical representations and position to the nodes in the Graphics Window. What is the robot going to look like?

**NOTE:** As we create the characteristics for the node, make sure the Graphics Window is open next to the Tree Node Browser so you can see the robot take shape, and so you can make changes as necessary.

**Joint Parameters**

After assigning the appropriate attributes to the nodes, we must set the desired parameters to them. The first parameters we will set are those for the joints. The joints in this tutorial include **Axis1**, **Axis2**, **Axis3** and **Axis4**. The joint parameters set here are used to move the robot in a specified angle and direction.

"**Axis1**" Parameters

1. Select **Axis1** in the Tree Node Browser by clicking on it.

2. Open the Node Inspector, and click on the **Joint** tab.
3. Set the following parameters:

**Joint type:** Rotational (rotates about an axis)

**Dependency:** Independent

**Max value:** 180

**Min value:** -180

Now that you have set the Joint parameters of your first node, you must position the node in a specified location on the screen. Previously, the Joint tab was used to set the joints at either Rotational or Translational in order to rotate or move the joint in a certain way. The Pose tab is used to position that node in a certain location on the screen. This is one of the benefits of using the Graphics Window. You can actually see how the joint will move or rotate and where it is positioned on the screen.

In order to position a node on the screen, open the Node Inspector. You will see the second tab labeled Pose. Click on the Pose tab to open the following panel (make sure the desired node is highlighted in the Browser):
GETTING STARTED WITH CODE

Figure 4-8: Node Inspector Pose Panel

The \( X \), \( Y \), and \( Z \) buttons are used to position the node in a certain location on the screen. If you play with the buttons, each one moves the triad in a different direction. To set the position of the \textbf{Axis1} node, first click on \textbf{Axis1} in the Browser. You will see the node name in the Relative to: field change from \textit{world} to \textit{Robot}. The purpose of the Relative to: field is to show how the node you selected is relative to other nodes in the workcell. The Use Parent box, when checked, will automatically select the parent of the node you have selected in the Browser. If this box is unchecked, you can select any node from the Browser to be entered as the Relative to node. Nevertheless, leave this box checked and set the following parameters for the \textbf{Axis1} node:

\textbf{Relative to: Robot}

\begin{align*}
X & : 0.0 \\
Y & : 0.0 \\
Z & : 1000
\end{align*}

The \textbf{Axis} and \textbf{Angle} fields in the Pose panel should be set to default, but just for good measure, make sure they are set to:

\begin{align*}
\textbf{Axis: Self X} \\
\textbf{Angle: 0}
\end{align*}
The parameters for the other three joints, Axis2, Axis3, and Axis4 are set up just the same as Axis1, so only the parameters for each are listed hereafter. Use the instructions for Axis1 to give the other Joint nodes their attributes.

"Axis2" Parameters
From the Joint panel, set the following parameters:

Joint type: Rotational

Dependency: Independent

Max value: 180

Min value: -180

From the Pose panel, set the following parameters:

Relative to: Axis1

X: 500

Y: 0

Z: 0

"Axis3" Parameters
From the Joint panel, set the following parameters:

Joint type: Translational (moves along a line instead of rotating around an axis)

Dependency: Independent

Max value: 0

Min value: -200

From the Pose panel, set the following parameters:

Relative to: Axis2

X: 500

Y: 0

Z: 0
"Axis4" Parameters
From the Joint panel, set the following parameters:

Joint type: Rotational
Dependency: Independent
Max value: 180
Min value: -180

From the Pose panel, set the following parameters:

Relative to: Axis3
X: 0
Y: 0
Z: -25

TCF Parameters
For this tutorial, we have added a TCF node. TCF stands for Tool Control Frame, and is used when adding a tool type to your robot. Some of the more common tools used in a robot include a gripper, a vacuum or a camera. Though this tutorial does not contain a tool, we have added a TCF node in order to explain what a TCF is and how it is created.

From the Pose panel, set the following parameters:

Relative to: Axis4
X: 0
Y: 0
Z: -50

Geometry Parameters
In this workcell, there are five nodes which require the geometry attribute, Base, Arm1, Arm2, Arm3 and Arm4. In order to set the Geometry parameters to a Geometry node, first go to the Browser and select the desired node. Then open the Node Inspector and click on the Geometry tab, which will display the following panel:
"Base" Parameters
1. Select **Base** by clicking on the node name in the Browser.

2. Open the Node Inspector and click on the **Geometry** tab.

3. Under **Geometry Settings**, click the down-arrow and select **Cylinder**.

Four boxes will appear below the down arrow. Enter the following information into these fields:

- **Rad**: 100
- **Len**: 1000
- **Res**: 8
- **Tol**: Will fill in automatically when **Res** amount is entered.

4. Click on **Standard Colors** and select the desired color of **Base**.

"Arm1" Parameters
Setting the parameters for **Arm1** or any of the other three arms is no different than the parameters set for **Base**, except for the shape and entry of dimensions, of course.

1. Select **Arm1** in the Browser and open the **Geometry** tab in the Node Inspector.
2. Click on the down-arrow and select **Box**.

Three boxes, **L1**, **L2** and **L3** will appear on the left side of the panel. The amounts entered into these boxes will change the dimension of the shape in the Graphics Window. Enter the following amounts:

**L1**: 800

**L2**: 200

**L3**: 350

3. Click on the **Pose** tab and enter the following amounts:

**X**: 300

**Y**: 0

**Z**: 0

This will position the **Arm1** shape on top of the cylinder created in **Base**.

4. Click on **Standard Colors** and select the desired color of **Arm1**.

**"Arm2" Parameters**

1. Select **Arm2** in the Browser and open the **Geometry** tab in the Node Inspector.

2. Click on the down-arrow and select **Box**.

Enter the following parameters:

**L1**: 600

**L2**: 200

**L3**: 250

3. Click on the **Pose** tab and enter the following amounts:

**X**: 300

**Y**: 0

**Z**: 0

4. Click on **Standard Colors** and select the desired color of **Arm2**.
"Arm3" Parameters
1. Select Arm3 in the Browser and open the Geometry tab in the Node Inspector.

2. Click on the down-arrow and select Cylinder.

Enter the following parameters:

Rad: 50
Len: 200
Res: 8

Tol: Will fill in automatically when Res amount is entered.

3. Click on Standard Colors and select the desired color of Arm3.

"Arm4" Parameters
1. Select Arm3 in the Browser and open the Geometry tab in the Node Inspector.

2. Click on the down-arrow and select Cylinder.

Enter the following parameters:

Rad: 60
Len: 25
Res: 8

Tol: Will fill in automatically when Res amount is entered.

3. Click on Standard Colors and select the desired color of Arm4.

Robot Parameters
The first node we will to give parameters to is Robot. If you look at the Browser, you will see that this node is labeled directly under world, though it does not have to be as previously stated. In order to give the Robot node the proper attributes:

1. Click on Robot in the Browser

2. Open the Node Inspector, and click on the Robot tab.
3. Add the following parameters:

**Control:** Inverse Kinematics

**Control Type:** Auto DH

**Value:** 4009

**Soln#:** -1

**NOTE:** If you get the following error message:

```
No DH match was found.
Set Robot to numerical inverse kinematics or add its custom solution to ROBLINE.
```

then you have set the parameters wrong in the nodes, so go back to the **Pose** and **Joint** panels in the Node Inspector and make sure the parameters we have listed for each node are correct.
The final workcell should look like the following:

![Completed Tutorial Workcell](image)

Figure 4-11: Completed Tutorial Workcell

**Saving Your Workcell**

We will be saving our workcell as **Filename.w**.

**NOTE:** As stated previously, all CIMServer workcells have the extension .w.

1. Click on the **Save** icon ![Save Icon](icon) from the icon bar, or select **File | Save** from the menu bar in the Tree Node Browser.

2. This will bring up the **Save As** box.

3. Find the directory that you want to save your file in
NOTE: It is generally not a good idea to store your workcells in the same location as the demos in the Cimetrix directory. That location is install_dir\lib\cimetrix\demos\ (install_dir is your installation directory, default is C:\Cimetrix). Instead, save workcell files to your users area.

4. Type the name you want to save your file as in the File name box. Filename.w is a good name for this file.

5. Click Save to save the file to the selected directory, or Cancel to close the box without any action taken.

Running Your Robot Model Using CIMTools

You can actually see your robot’s motion capabilities by using the Teach Pendant. The purpose of the Teach Pendant is to "jog" the robot. This means to move the robot axes from one location to another. The Teach Pendant allows the user to check the joint parameters to make sure they are correct. For example, if one joint is supposed to move 180 degrees, but only moves 90 degrees, you can see it and make corrections in the Node Inspector as necessary. To open the Teach Pendant:

1. Click on the Teach Pendant icon from the Icon Bar in the Tree Node Browser. This will display the Teach Pendant.

2. Click either the arrows to the right of Robot: or the Connect button. Either will connect you to the Teach Pendant.

Figure 4-12: Disconnected Teach Pendant
At the top of the Teach Pendant, you will see two boxes, **Robot** and **TCF**. If you click on the arrow to the right of the box, it will automatically select the **Robot** node in the Browser. The **TCF** field will automatically be filled in with the last joint of the robot as shown in the Tree Node Browser.

3. When the Teach Pendant is connected, it will look like the following:

![Connected Teach Pendant](image)

**Figure 4-13: Connected Teach Pendant**

As you can see, the fields 0r, 1t, 2t and 3r are filled in. The arrow-buttons next to these fields, when pushed, will move the different arms of the robots in the direction the joint parameters have been set at.
Chapter 5

Programming Your Model

Once you have created your workcell, how do you program it in order to accomplish your tasks? Using Visual C++, this section shows you how to take the workcell you created in the previous section and give it functionality.

Setting Up the Development Studio

To program the model, we will be using Visual C++ 5.0. So, if you have this software, let’s get started. Click Start | Programs | Microsoft Visual C++ 5.0.

New Development Environment

Be sure to set the following options before writing code:

1. Select File | New and the New dialog box will be displayed.

2. For this project, we will create a simple console application, so click on the Projects tab and select Win32 Console Application.

3. In the Project name field, enter the name of your project. For all intents and purposes, we will call this Tutorial.

4. Enter the location of your program in the Location field.

5. Click OK.

The development environment is divided into two frames. After the actions listed above are completed, the environment will look like the following:
6. Click **Project | Add to Project | New**.

7. Select **Text File** from the **Files** tab.

8. Type the file name (**Tutorial**) in the **File name** field. Here, you must enter the `.c` extension after the file name. This extension means that it is a source code file.

9. Click **OK**.

You will see the right frame in the development studio change color from grey to white. You may think that this means you are ready to start coding, but there are a few more options which must be set.

**Project Settings**

1. Click **Project | Settings**.

2. Click the **C/C++** tab.
3. Under **Category**, select **Code Generation**.

4. Under **Use run-time library**, select **Multithreaded DLL**.

5. Click on the **Link** tab.

6. While leaving the existing text in the **Object/library modules** field, enter the following at the beginning of the existing text:

   Cimetrix.lib robpac.lib wsock32.lib

7. Click **OK**.

8. Click **Tools** | **Options**

9. Click the **Directories** tab.

10. In the **Show Directories For** field, select **Include Files**.

11. In the **Directories** field, you will see a blank outlined box. Click on it.

12. Click on the box with the three periods in it. This will display the Directories dialog box.

13. Select the following path:

   C:\Cimetrix\Include

14. Go back to the **Show Directories For** field and select **Library Files**.

15. Click on the blank box in the **Directories** field.

16. Again, click on the box with the three periods in, which will display the **Directories** dialog box.

17. Select the following path:

   C:\Cimetrix\lib

---

**The Program**

The following is a short, extremely simple program which will jog the robot you created in Part IV. Type the following into Microsoft Visual C++. Though we have only given it three moves with this program, the number of possible moves is endless.

```
#include <iostream.h>
#include <code/robpac.h>
```


```c
#include <code/robpac_error.h>

void main( void )
{
    // Local Variables
    CxServer myServer;
    CxMechanism robot;
    CxNodeId nidRobot, nidTcf, nidTp1, nidTp2, nidTp3;
    char *server_name;
    long count;

    // Get Default CIM Server Name
    if( CxGetDefaultServer(&server_name)==CX_ERROR ) {
        cerr << "No default CxServer found\n";
        exit(-1);
    }

    // Open Connection To CIMServer
    myServer = CxOpenServer( server_name, CX_SMEM, 0 );
    if (myServer == NULL) {
        cerr << "Unable to open server.\n" ;
        exit (-1);
    }
    CxSetErrorPolicy ( myServer, CX_THROW_ERROR ) ;

    // Initialization
    try {
        CxRestoreDefaultState( myServer );
        CxSetSimrate(myServer,0.25);

        // Get Node IDs
        CxGetNamedNodeId(myServer, "robot", &nidRobot);
        CxGetNamedNodeId(myServer, "tcf", &nidTcf);
        CxGetNamedNodeId(myServer, "tp1", &nidTp1);
        CxGetNamedNodeId(myServer, "tp2", &nidTp2);
        CxGetNamedNodeId(myServer, "tp3", &nidTp3);

        // Open Connection To Robot
        robot = CxOpenMechanism( myServer, nidRobot, CX_CONTROL);
        CxSetInterpType( robot, CX_JOINT_INTERP );
        CxSetJointSpeed( robot, 0.25 );
    } catch (CxErrorStruct errordata) {
        cerr << "ERROR (catch):" << errordata.msg << '\n';
        CxRobpacExit();
    }

    // Very Simple Motion
    try {
        for (count = 0;count < 3; count++) {
            CxMoveToNode(robot, nidTp1, nidTcf);
            CxMoveToNode(robot, nidTp2, nidTcf);
            CxMoveToNode(robot, nidTp3, nidTcf);
        }
    } catch (CxErrorStruct errordata) {
        CxSendMechanismErrorAction(robot, CX_MECH_ABORT);
        cerr << "ERROR (catch):" << errordata.msg << '\n';
    }
}
```

---

**GETTING STARTED WITH CODE**

5-4
After this code has been entered into the development studio, make sure to save your work, then we will jog the robot using teach points (TP1, TP2 and TP3).

**Checking for Errors in Your Program**

After you have entered the above code, it is necessary to check for errors as the code is case sensitive and it is very easy to make mistakes.

To check for coding errors in the development studio, select **Build | Build Tutorial.exe** from the menu bar. Any errors will be displayed at the frame at the bottom of the screen.

**Running Your Robot**

The code you wrote above will run your robot automatically, except first, you must enter certain code elements into the Teach Pendant.

**Using the Teach Pendant**

1. Open the Teach Pendant (click the Teach Pendant icon).
2. When you connect to the Teach Pendant, the **Robot** field will automatically be filled in with **Robot**. The **TCF** field, however, will be filled in with whichever node you have selected in the Browser at the time. Therefore, after you connect to the Teach Pendant, select **TCF** in the Browser and click on the button with the two arrows to the right of the **TCF** field (see Figure 19).

If you look at the bottom of the Teach Pendant, you will see the **More** box. It is used as a space saver. If you click on it, an additional area of the Teach Pendant will be displayed as so:

![Figure 5-3: More Section of the Teach Pendant](image)

**NOTE:** Make sure the Graphics Window, Tree Node Browser, and Teach Pendant are all displayed on the screen.

3. Click **Node** (make sure you select **Node** from the bottom portion of the Teach Pendant).

4. Look at the robot in the Graphics Window. Using the up- and down-arrow buttons next to the **0r**, **1r**, **2t** or **3r** fields in the upper portion of the Teach Pendant, move the robot to a position of your choice.

5. Go back to the bottom portion of the Teach Pendant and type **TP1** in the **Teach point** field.

6. Click **Teach**, and you will see **TP1** appear in the white box on the left.

7. Go back to the Graphics Window and move the robot again to any coordinates of your choice using the up and down-arrow buttons next to the **0r**, **1r**, **2t** or **3r** fields of the Teach Pendant.

8. Go back to the bottom portion of the Teach Pendant and type **TP2** in the **Teach point** field.

9. Click **Teach**, and you will see **TP2** appear in the white box on the left.
10. Go back to the Graphics Window and move the robot again to any coordinates of your choice using the buttons in the upper portion of the Teach Pendant.

11. Go back to the bottom portion of the Teach Pendant and type TP3 in the Teach Point field.

12. Click Teach, and you will see TP3 appear in the white box on the left.

**Executing the Program**

Once you have entered the program information into the Teach Pendant as shown above, the movements you told the robot to make will be made when executing the program from the Visual C++ Development Studio.

Make sure you have the Graphics Window showing so that you can see the robot executing the program. Also note that this is a simple program with only three motions, and that you can create multiple axes robots with hundreds of motions depending on what you want your robot to do. For now, lets execute our program and make the robot run.

1. Go back to the Visual C++ Development Studio and open the previously created code file Tutorial.c.

Once the file is open, click Build | Execute Tutorial.exe from the Menu bar.

This will start the program running. A DOS Window may cover the Graphics Window, so you may have to move it out of the way in order to see your robot execute the program. If your robot is moving, then it is working correctly. Finally, make sure you save your work.

**Conclusion**

Thank you for viewing the information in *Getting Started With CODE*. Hopefully, you have gained an appreciation for what CODE is, and that the tutorial contained herein has been both informative and entertaining. If you have any comments or suggestions as to how this or any other manual in the CODE collection can be improved, please feel free to contact (801) 256-6500. For more information on CIMTools, please see the *CIMTools Reference Manual*. 
<table>
<thead>
<tr>
<th>U</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgrading CODE Programs .............................................. 2-2</td>
<td>Visual C++ Development Studio</td>
</tr>
<tr>
<td>New Development Studio ............................................... 5-1</td>
<td></td>
</tr>
<tr>
<td>Project Settings ................................................................ 5-2</td>
<td></td>
</tr>
</tbody>
</table>