

# Table of Contents

## 1 Introduction 5

About This Manual .....	7
User Assumptions .....	7
Important Note.....	7
What This Document Includes:.....	7
Conventions Used In This Manual.....	8
System Requirements.....	11
Operating System and Hardware Requirements.....	11
Install12	
Installing on Windows NT and Windows 95 .....	12
Installing On OS/2 .....	13

## 2 What Is Paragon? 17

Paragon Servers.....	20
Servers Are Function Based .....	21
Paragon Clients .....	22
Building Clients With RAD Tools .....	22
Anatomy Of A Paragon Application .....	24

## 3 Paragon Quick Start 27

Starting Paragon.....	29
Application Wizard .....	30
Starting The Application Wizard .....	31
Process I/O Setup .....	33
Communications Port .....	33
Tag Names.....	34
Selecting Tag Names .....	36
Data Manger History .....	37
Operator Interface Display .....	39
Application Summary.....	40
Chapter Summary.....	44

## 2 i Paragon Getting Started

### 4 Process I/O Values 45

Process I/O Subsystem .....	47
Editing The Wizard Library File .....	47
Creating The Configuration File .....	51
Operator Interface.....	54
Entering The Operator Interface.....	54
Creating A Display Window.....	57
Placing Objects.....	58
Setting Object Attributes .....	62
Editing The Data Entry Field .....	62
Editing The List Box.....	64
Editing The Data Display Field .....	66
Saving The Display.....	67
Application Manager .....	68
Control Panel.....	68
Saving The Application File .....	70
Running The Application .....	70
Engineering Interface.....	72
Chapter Summary.....	74

### 5 Using The Continuous Strategy 75

Continuous Strategy .....	77
Creating A Strategy Display Window.....	77
Building The Strategy .....	80
Connecting Process Blocks.....	81

Creating Function Blocks.....	82
Connecting Process I/O Functions.....	84
Saving The Strategy.....	87
Operator Interface.....	88
Opening Your Display .....	88
Placing Objects.....	88
Defining the Attributes .....	90
Assigning Actions For Display Call-ups.....	91
Application Manager.....	93
Runtime .....	94
Closing the Display.....	95
Chapter Summary.....	96
<b>6 Tank Fill Simulation 97</b>	
Continuous Strategy .....	99
Defining Function Blocks.....	99
Connecting Function Blocks.....	103
Saving Your Strategy .....	104
Operator Interface.....	105
<b>Table of Contents i 3</b>	
Designing The Tank .....	105
Digital Animation.....	108
Creating A PID Faceplate .....	114
Display Call-up.....	116
Runt118	
Chapter Summary.....	119
<b>7 Trend And Alarm Displays 121</b>	
Data Manager .....	123
Creating The library File .....	123
Configure Trend Functions.....	124
Creating The Configuration File .....	127
Configuring Alarm Functions .....	128
Continuous Strategy .....	129
Creating A Trend Display.....	131
Adding A Close Button. ....	133
Creating An Alarm Log.....	134
Adding A Close Button To Your Display.....	136
Runtime .....	137
Using the Display.....	138
Chapter Summary.....	140
<b>8 Index 141</b>	
<b>4 i Paragon Getting Started</b>	

# Introduction

## In This Chapter:

• [About This Manual 7](#)

• [System Requirements 11](#)

• [Installation 12](#)

**W**elcome to a first-time user's guide to Paragon, an introduction to today's premier factory automation program. Like other programs, there are many ways to complete a task. This guide provides the easiest way to do each one. Only the basics are covered, giving you the necessary information to get the job done. By the end of this document you will know:

Paragonの初心者ユーザーのためのガイド、現在最高のファクトリー・オートメーション・プログラムによろこそ。他のプログラムのように、タスクを完成する多くの方法があります。このガイドはその最も簡単な方法を提供します。基本のみをカバーして、仕事を完了させるために必要な情報を与えます。このドキュメントの最後までに、あなたは知るでしょう：

**The basic terms and concepts for using Paragon.**

Paragonを使用するための基本的な用語と概念

**How to create a Paragon**

Paragonを作成する方法

**To help you achieve these goals, this tutorial guides you through the application building process step-by-step.**

これらのゴールの達成を助けるため、この教本は、段階的にアプリケーション構築のプロセスをガイドします。

# About This Manual

## このマニュアルについて

### User Assumptions

#### 想定するユーザー

It is assumed that you have basic PC skills and an understanding of your operating system. No previous knowledge of Paragon is needed to complete the tasks in this manual. It is for anyone who is interested in using Paragon to automate their existing factory processes.

基本的なPC技能とオペレーティングシステムについての理解が必要です。Paragonの前提知識は、このマニュアルのタスクの完成に必要とされません。このマニュアルは、既存の工場プロセスを自動化するためにParagonを使用することに興味を持つ人のためのものです。

### Important Note

#### 重要な注意

If you are using the Paragon Test Drive product, please be aware that the application components created with the Test Drive product **CANNOT** be used with or converted to any other Paragon model. Paragon Test Drive is provided for evaluation and demonstration purposes only.

Paragon Test Drive製品を使用しているなら、Test Drive製品で作成されたアプリケーション・コンポーネントは、他のすべてのParagonモデルで使用ができず、また変換もできないことを知っておいてください。Paragon Test Driveは、評価とデモンストレーションのためのみに提供されています。

### What This Document Includes:

#### このドキュメントに含まれるのは：

This document is divided into six chapters:

このドキュメントは6つの章に分かれています：

#### Chapter 1 Introduction:

##### 第1章 イントロダクション

An introduction to this manual.

このマニュアルの紹介

#### Chapter 2 What Is Paragon?:

##### 第2章 Paragonとは何か？

This chapter gives you a quick overview of Paragon's components and capabilities.

この章は、Paragonのコンポーネントと能力の概観を提供します。

#### Chapter 3 Paragon Quick Start:

##### 第3章 Paragonクイック・スタート：

Shows you how to create a basic application using the Application Wizard.

アプリケーション・ウィザードを使用して、基本的なアプリケーションを作成する方法を教えます。

#### Chapter 4 Process I/O Values:

##### 第4章 Process I/Oの値：

Introduces you to the Process I/O, the Operator Interface, and the Application Manager. Here you will add to the Wizard Application.

Process I/O, Operator Interface, およびApplication Managerを紹介します。ここで、Wizardアプリケーションについてより深く知ることができます。

#### **Chapter 5 Using the Continuous Strategy:**

Introduces you to the Continuous Strategy.

Here you will implement a strategy to sum the values from the Process I/O.

#### **Chapter 6 Tank Fill Simulation:**

Gives you more details on using the Continuous Strategy.

In this section you will build a display to simulate the flow in and out of a tank.

#### **Chapter 7 Tank And Alarm Displays:**

Introduces you to the Data Manager.

In this section you will use the Data Manger to access trend and alarm data from the Tank display.

## Conventions Used In This Manual

### このマニュアルで使用される取り決め

## Using The Mouse

### マウスの使用について

Paragon supports both mouse buttons. Throughout this manual “click” refers to the left mouse button. If the right mouse button is needed to perform a task then it is specified by a “right-click.”

Paragonは2つのマウスボタンをサポートします。このマニュアルの全体にわたって、“クリック”はマウスの左ボタンを言及します。マウスの右ボタンがタスクを行なうために必要な場合、それは“右クリック”と指定されます。

If menu commands are listed in gray, you cannot access them in the current operation. If a command has an ellipsis (...) after it, a dialog box appears allowing you to enter additional information. Certain fields require that you double-click on them to complete additional information. If a field requires a text entry, click on the field before typing unless specified otherwise.

もしメニューのコマンドが灰色でリストされていたら、その時に行っている操作ではアクセスできません。後ろに省略(...)を持つコマンドは、追加情報を入力するダイアログ・ボックスがあらわれます。特定の項目によっては、追加情報を完成させるために、その上でダブルクリックすることを求めます。項目がテキスト入力を必要とする場合、他の方法が指定されていなければ、タイプするまえに項目をクリックしてください。

Certain windows contain list boxes which have selectable values and a vertical scroll bar. There are three ways to move through a list box:

あるウィンドウは、選択可能な値と垂直スクロール・バーを持つリストボックスを備えます。リストボックスを移動するには3つの方法があります：

**Position the mouse over the scroll bar, hold down the left mouse button, and slide the mouse up or down.**

スクロール・バー上にマウスの位置を決めて、マウスの左ボタンを押したまま、マウスを上もしくは下に滑らせます。

**Clicking on the up or down arrows.**

上向きあるいは下向きの矢印をクリックします。

**Type the first letter of the entry to move to it.**

選択したい項目の最初の文字をタイプします。

Most windows can be resized and moved. To move a window place the mouse cursor over the title bar, hold down the left mouse button, and drag the window. To resize a window place the mouse cursor over the frame side or corner until the cursor appears as a double arrow. Hold the left mouse button down and drag the edge of the window.

ほとんどのウィンドウはサイズ変更と移動ができます。ウィンドウを移動させるには、タイトル・バー上にマウス・カーソルを置いて、マウスの左ボタンを押さえたまま、ウィンドウをドラッグします。ウィンドウをサイズ変更するには、カーソル形状が二重の矢印としてあらわれるまで、マウス・カーソルをフレームの端か角に置いて、マウスの左ボタンを押さえたまま、ウィンドウの端をドラッグしてください。

## Using The Keyboard

### キーボードに使用について

Use the following commands to perform the listed operation:

リストの操作を行うには下記の指令を使用してください：

**Tab** moves you to the next field.

**Tab** 次の領域に移動する。

**Shift+Tab** moves the cursor to the previous field.

**Shift+Tab** 前の領域にカーソルを移動する。

**Down Arrow** moves the cursor to the next item in the list or to the next field.

下矢印(↓) リストの次の項目あるいは次の領域にカーソルを移動する。

**Up Arrow** moves the cursor to the previous item in the list if there is one, or to the previous field.

上矢印(↑) リストに前項目があればカーソルを移動させる、あるいは前の領域にカーソルを移動させる。

**Right Arrow** moves the cursor one character to the right in a data entry field or to the next right field.

右矢印(→) データ入力領域では1文字右にカーソルを移動させる、あるいは次の右の領域にカーソルを移動させる。

**Left Arrow** moves the cursor one character to the left in a data entry field, or to the next **right** field.

左矢印(←) データ入力領域では1文字左にカーソルを移動させる、あるいは次の左の領域にカーソルを移動させる。

**Enter** enters the information for a dialog box or enters a selection from a list.

Enter ダイアログ・ボックスに情報を登録するか、あるいはリストから選択を入力する。

**Esc** exits out of a dialog box without saving any modifications.

Esc すべての変更を保存しないでダイアログボックスから出る。

This has the same effect as the Cancel button.

これはCancelボタンと同じ効果があります。

**F1** generates a Help window on the current dialog box.

F1 現在のダイアログ・ボックスに関するヘルプ・ウィンドウを生成します。

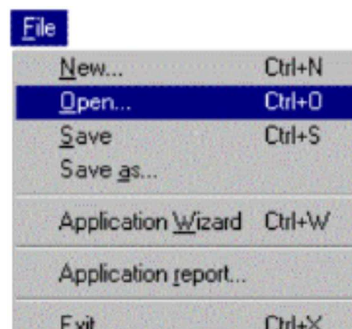
## Typeface Styles 書体

Throughout this manual **bold** is used to indicate a command, or to specify a field name. Italics are used whenever a field requires a text entry. **New Terms** appear in bold and italics, and are defined in the margin.

このマニュアルの全体にわたって、**太字**はコマンドを示すかあるいは領域名を指定するために使用されます。領域がテキスト入力を要求する場合は常に、*イタリック体*が使用されます。**新しい用語**は太字とイタリック体で、余白に定義があらわされます。

## Menu Commands メニュー・コマンド

Certain functions are completed using drop-down menus; others by use of the toolbars. Menu commands are illustrated by the following:



ある機能はドロップダウン・メニューを使用して完成します；ツールバーを使用する以外のもの。メニュー・コマンドは下記のように例示されます：

### File ▶ Open

This means that you open the file menu by clicking on the word **File** in the upper-left corner of your screen, dragging down to the word **Open** and releasing once it is highlighted.

これは、画面の左上隅のワード**File**をクリックし、ワード**Open**まで下げて、そしてそれがハイライトしてからマウスのボタンを離すことで、ファイルメニューを開くことを意味します。



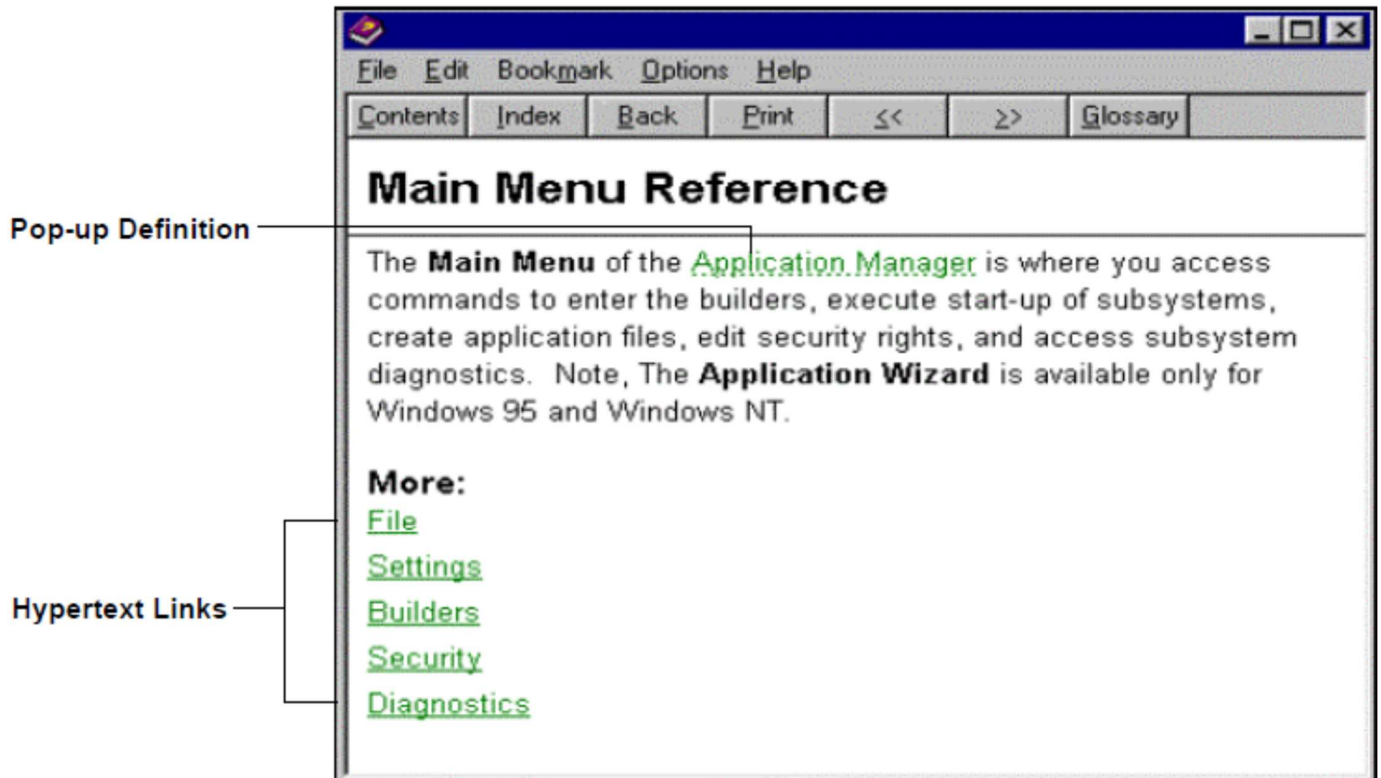
## Using Online Help オンライン・ヘルプの使用について

Help is accessed through each subsystem's Help Main menu, or by selecting the Help push button in the current application's dialog box. Online Help is context-sensitive: The help you get is determined by what is selected when you request it.

各サブシステムのメインメニューのHelpにより、または、現在のアプリケーションのダイアログボックスでHelpのプッシュボタンを選択することで、ヘルプはアクセスされます。オンライン・ヘルプは状況依存です：得られるヘルプはそれを要求したときに選択しているものにより決まります。

Many help windows reference information in other Help windows, providing a hypertext link that allows you to jump directly to that window. A hypertext link is indicated by highlighted text.

多くのヘルプ・ウィンドウが他のヘルプ・ウィンドウの情報を参照するため直接ジャンプすることを可能にするハイパーテキストリンクを提供します。ハイパーテキストリンクはハイライトされたテキストにより示されます。



**Contents** displays a hypertext table of contents for Paragon's help directory.

**Contents** Paragonのヘルプ・ディレクトリのハイパーテキストの目次を表示します。

**Index** allows you to search help by selecting or entering key words.

**Index** キーワードの選択か入力によりヘルプを検索します。

**Back** returns you to the previous help window.

**Back** 前のヘルプ・ウィンドウに戻ります。

**<<** moves you to the previous hypertext entry in a list.

**<<** リストの前のハイパーテキスト・エントリーに移動します。

**>>** moves you to the next hypertext entry in a list.

**>>** リストの次のハイパーテキスト・エントリーに移動します。

# System Requirements システム要求

## Operating System and Hardware Requirements オペレーティング・システムとハードウェアの要求

Paragon supports the following operating systems:

Paragonは次のオペレーティング・システムをサポートします：

Microsoft Windows NT version 4.00 or later.

Microsoft Windows 95 or later.

IBM OS/2 Warp Version 3.0 with FixPak 27 or later.

In addition to the operating system's requirements, Paragon also requires the following as a minimum for all operating systems:

オペレーティング・システムの必要条件に加えて、Paragonはすべてのオペレーティング・システムで最低限として下記を要求します：

One parallel port.

1つのパラレル・ポート

200 MB of free hard disk space.

200MBのハードディスク空き容量

486DX-66 MHz processor.

24MB RAM for Windows 95 and OS/2.


32 MB Ram for Windows NT.

# Installation インストール手順

## Installing on Windows NT and Windows 95 Windows NTとWindows 95のインストール作業

**To install Paragon on Windows NT and Windows 95:  
Windows NTおよびWindows 95にParagonをインストールするには：**

1. Place the Paragon CD into your CD drive.  
CDドライブにParagon CDを入れてください。
2. Double-click on the D: drive from Windows Explorer. Assuming that D: is the letter of your CD drive.  
WindowsエクスプローラからD:ドライブをダブルクリックしてください。CDドライブをD:と仮定します。
3. When the Paragon TNT file folder opens, double-click on the icon: Autorun.exe: . Follow the installation instructions.

Paragon TNTのファイル・フォルダーが開く場合、アイコン：Autorun.exe :  をダブルクリックしてください。インストール指示に従います。

If this does not work, then do the following:  
これが動作しない場合、以下を行ってください：

1. If using Windows NT login as a user with Administrative Privileges.  
管理者権限を持つユーザとしてWindows NTログインを使用する場合。

2. Place the Paragon CD into your CD drive.  
CDドライブにParagon CDを入れてください。

3. Open an MS-DOS window and type C:¥>D: <ENTER>  
MS-DOSウィンドウを開いて、C:¥>D: <ENTER>をタイプします。

Assuming that D is the letter of your CD drive. If not, use the letter designating the CD drive on your system.  
この説明ではCDドライブをD:と仮定しています。異なる場合は、システム上のCDドライブの文字を使用してください。

4. Type D:¥>CD¥Paragon¥TNT <ENTER>  
D:¥>CD¥Paragon¥TNT <ENTER> をタイプします。

This changes your directory to ¥Paragon¥TNT.  
これは¥Paragon¥TNTにディレクトリを変更します。

5. At D:¥ Paragon¥TNT> type: LOADWIN <ENTER>  
D:¥ Paragon¥TNT> で LOADWIN <ENTER> をタイプします。

This initiates Paragon's Custom installation program that will guide you through the installation process. Be sure to read the instructions on each screen.

これはインストールプロセスを通してあなたを案内するParagonのカスタム・インストール・プログラムを開始します。各スクリーンの説明を必ず読んでください。

**Note:** If you are installing the SI or Demo version, type LOADWIN SI or LOADWIN DEMO.

注意：SIもしくはDemoバージョンをインストールするなら、LOADWIN SIもしくはLOADWIN DEMOをタイプしてください。

6. At the Welcome window click on Next.

Welcomeウィンドウで、Nextをクリックしてください。

7. Select the **Paragon Test Drive** radio button from the next window and click on continue.

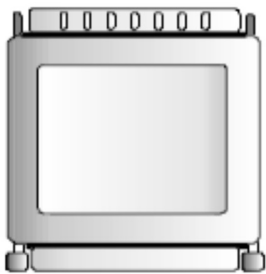
次のウィンドウから**Paragon Test Drive**のラジオボタンを選択して、continueをクリックします。

8. The Install to path at the top left of the screen should be C:\¥PARATNT. Under Installation Options make sure that **Builder, Runtime, and Onboard Demo** are all checked. Under PIO Drivers highlight Simulated I/O, then click on **Install**. A driver is a program that relays information from your PIO database to your I/O.

スクリーンの左上のInstall to pathはC:\¥PARATNTであるべきです。Installation Optionsで、**Builder, Runtime,および Onboard Demo**のすべてを確実にチェックしてください。そしてPIO DriversからSimulated I/Oをハイライトにして、**Install**をクリックします。ドライバはPIOデータベースからI/Oまで情報をリレーするプログラムです。

9. Attach the Paragon Protection Key that came with the application to the parallel port on your computer. Unless you are using the Test Drive. When you are done installing, reboot your computer. The protection key looks like the following:

アプリケーションに付属するParagon Protection Keyをコンピュータの平行ポートに取り付けてください。もし、Test Driveの使用で無い場合には、インストールを終えたら、コンピュータを再起動してください。プロテクトキーは下記のように見えます：




Paragon has created a program group that you can access from the Start Programs menu. Included in the Paragon directory are the following:

Paragonは、スタート・プログラム・メニューからアクセスできるプログラム・グループを作りました。下記がParagonディレクトリに含まれています：


The **Paragon icon**: is used to start the program.

**Paragon**アイコン： はParagonを開始するのに使用します。

The **Installation Utility icon**: is used to install updates or additional components. It is also used to view information on your installed products including Paragon TNT developer kits such as PDK. When open, the Installation and Maintenance window used to install the product will appear. Information contained here is required by NemaSoft Technical Support if you call with questions about the product.

**Installation Utility**アイコン： はアップデートや追加コンポーネントをインストールするために使用されます。さらに、それはPDKのようなParagon TNT開発者キットを含む、インストール済みの製品情報を見るのに使用されます。開くと、製品をインストールするために使用するインストールとメンテナンスのウィンドウがあらわれます。製品についての疑問を問い合わせるには、ここに含まれる情報をNemaSoft技術サポートに求められます。

The **ODBC Administrator**: is used to create data source names to interface with relational databases.

**ODBC Administrator**: はリレーショナル・データベースと接続するデータソース名の作成に使用されます。

## Installing On OS/2

### To install Paragon on OS/2:

1. Place the Paragon CD into your CD drive.
2. Open an OS/2 window and Type C:¥>D: <ENTER>  
Assuming that D is the letter of your CD drive. If not, use the letter designating the CD drive on your system.
3. Type D:¥>CD¥Paragon¥TNT <ENTER>  
This changes your directory to ¥Paragon¥TNT
4. At D:¥Paragon¥TNT> type: LOADOS2 <ENTER>  
This initiates Paragon's Custom Installation program that will guide you through the installation process. Be sure to read the instructions on each screen.
5. Read the information on the Paragon Installation window, then click on **Continue**.
6. When the next Installation and Maintenance window appears, click on **Action**, then select **Install** from the pull-down menu.

### 14 ì Paragon Getting Started

7. Once the Install Window appears, make sure that the checkmark is present to update the CONFIG.SYS file. Then click **OK**. This will cause the protection key device driver to be added to your CONFIG.SYS file. It also adds the correct path information for Paragon TNT.  
**Note:** The installation program automatically adds a time zone setting for Eastern Standard Time in your CONFIG.SYS file.
8. In the Install Components window, it is advised that you only select the needed install components. Omitted components can be installed later. To Install all the components click on **Select All**. To install certain components click on it so it is highlighted. For this application you will need Builders, Runtime, and the Simulated I/O PIO Driver.
9. In the Root Directory field, enter the directory where you want to install Paragon. The default is **C:¥PARATNT**. If you want to change this, type in a new directory, then click on **OK**.
10. Click on **Install**.
11. When the installation is complete, exit from the Maintenance Window back to the OS/2 Desktop.
12. Reboot your system. The install program has placed a Paragon folder on your desktop. Double-click on the folder to open it. Inside are icons for Paragon, the Installation Utility, the ODBC Administrator, and the online documentation for the Process I/O.

0 The **Paragon** icon: is used to start the program.

0 The **Installation Utility** icon: is used to install updates or additional components. It is also used to view information on your installed products including: Paragon TNT developer kits such as PDK. When open, the Installation and Maintenance window used to install the product will appear. Information contained here is required by NemaSoft Technical Support if you call with questions about the product.

0 The **ODBC Administrator**: is used to create data source names to interface with relational databases.

13. The final step is to attach the Paragon Protection Key to your computer's parallel port. The protection key looks like the following:

### Chapter 1: Introduction ì 15

**Note:** Be sure to view the CAUTION.TXT file in the Paragon root directory for cautions regarding operation.

16 i Paragon Getting Started

&+\$37(5\_\_

# What Is Paragon?

## Paragonとは？

### In This Chapter:

この章では：

• **Paragon Servers 20**  
Paragonサーバ

• **Paragon Clients 22**  
Paragonクライアント

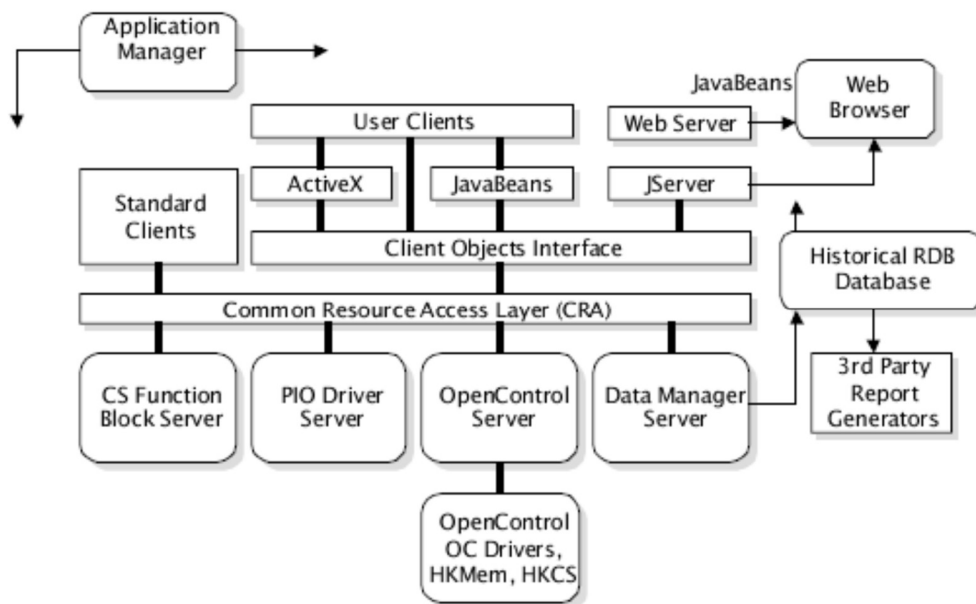
• **Anatomy Of A Paragon Application 24**  
Paragonアプリケーションの構造

Paragon is a 32 bit, client/server product that allows you to assemble mission critical Man Machine Interface (MMI) and Supervisory Control And Data Acquisition (SCADA) applications approaching Distributed Control System (DCS) level capability. Paragon is based on an open architecture with a full set of capable components. It is used in a wide range of industries, with applications ranging from below 100 I/O points to fully distributed systems containing thousands of I/O points.

Paragonは32bitの流通制御システム(Distributed Control System:DCS)レベルの能力にアプローチするミッション・クリティカルなMan Machine Interface(MMI)とSupervisory Control And Data Acquisition(SCADA)アプリケーションを組み立てるクライアント/サーバ製品です。Paragonは役に立つコンポーネントのフルセットを備えオープン・アーキテクチャに基づいています。それは広範囲の産業で使用され、アプリケーションは100のI/Oポイントから数千のI/Oポイントを持つ分散システムまで範囲にしています。

Application types range from fast sequential control using the OpenControl Server, to large distributed historical databases. These range from Relation Databases to internet structured systems, using the latest Java technology. The Paragon functional software architecture allows you to use (and pay for) only the components needed for your application. As your application grows, you can expand it by adding the necessary components. An introduction to these concepts follows:

アプリケーションのタイプは、OpenControlサーバを使用して、高速な逐次制御(sequential control)から、大きな流通履歴データベース(distributed historical database)まで及びます。これらは最新のJava技術を使用して、リレーショナル・データベースからインターネットの構造化システムにまで及びます。Paragonの機能的なソフトウェア・アーキテクチャは、アプリケーションで必要とされるコンポーネントだけを使用（および代金の支払い）することを可能にします。アプリケーションの成長に従って、必要なコンポーネントを加えて拡張することができます。これらのコンポーネントのイントロダクションが次に続きます：



The above figure shows the relationship between various application parts of a single Paragon installation. For distributed applications clients and servers can be located in a number of PC's in various combinations. Stations can contain servers, clients or a combination of the two. Networking is transparent to clients even when server applications are moved. Displays can be moved to different stations without modifying them.

上の図は1つのParagonインストールの様々なアプリケーション部分の関係を示しています。分散アプリケーションにおいて、クライアントとサーバは多くのPCに様々な組み合わせで位置することができます。ステーションは、サーバ、クライアントあるいは2つの連携を含むことができます。サーバ・アプリケーションが移動する時さえ、クライアントにとってネットワークは透過的（ユーザに意識させない）です。それらを変更しないで、表示を異なるステーションに移動することができます。

Client-Server architecture implies a separation of client and server functions, with a defined interface between them. In Paragon, each server performs a well-defined function deriving from information, control signals, and actions based on



measurements and schedules. Servers contain information and settings that are made available to clients. Clients contain data that is necessary for the client functions (display, recipe, etc.). Paragon's interface to servers is object-based and is called Client Object Interface (COI). COI is well documented with bindings for C, C++, Visual Basic, Delphi, Java and others. ActiveX and JavaBean data access components are available to simplify the interface for reading and writing properties.

クライアント-サーバのアーキテクチャは、それらの間に定義されたインターフェースおよびクライアントとサーバの機能の分離を意味します。Paragonでは、各サーバは、情報、制御信号、および測定とスケジュールに基づいた動作に由来する明確な機能を行いません。サーバは、クライアントに利用可能な情報と設定を持ちます。クライアントは、クライアント機能（ディスプレイ、レシピ、など）に必要なデータを持ちます。サーバへのParagonのインターフェースは、オブジェクトベースであり、Client Object Interface(COI)と呼ばれます。COIは、C, C++, Visual Basic, Delphi, Javaおよびその他のためのバインド（割り当て？）としてよくドキュメント化されます。ActiveXとJavaBeanのデータ・アクセス・コンポーネントは、プロパティを読み書きするためのインターフェースをシンプルにできます。

# Paragon Servers

## Paragonサーバ

Paragon's servers have no integrated user interface. This is provided by the clients. Typically, you build a server application that runs without user interaction. The five Paragon servers are:

Paragonのサーバーは統合ユーザ・インターフェースを持ちません。これはクライアントによって提供されます。一般に、ユーザの相互作用なしで稼動するサーバ・アプリケーションを構築します。5つのParagonサーバは以下の通りです。

**Application Manager:** AM is the nucleus of Paragon. It is used to access all the subsystems, and to execute Runtime. It also provides security access, subsystem diagnostics, and system and operator logs.

**Application Manager:** AMはParagonの核です。それは、すべてのサブシステムにアクセスして、Runtimeを実行するのに使用されます。それはまた、セキュリティ・アクセス、サブシステム診断、およびシステムとオペレーターのログを提供します。

**Process Input and Output:** PIO manages drivers for acquiring data from and sending changes to external plant devices. These devices include Programmable Logic Controllers (PLC's), Single Loop Controllers (SLC's), plain Input/Output, etc. It also provides signal conversion to Engineering units, linearization, alarming and error reporting. PIO supports multiple drivers that communicate through serial ports, PC adapters, TCP/IP, etc. Data is displayed by clients, typically without additional processing, and can be accessed over a network.

**Process Input and Output:** PIOは、外部のプラント装置のデータを取得し、それに変更を送信するため、ドライバを管理します。これらの装置はProgrammable Logic Controller(PLC)、Single Loop Controller(SLC)、単純な入出力、などを含みます。それはまたエンジニアリング・ユニットへの信号変換、線形化、警報とエラーの報告を提供します。PIOはシリアルポート、PCアダプタ、TCP/IP、などにより通信する多数のドライバをサポートします。データは追加処理なしでクライアントにより表示され、ネットワーク上でアクセスすることができます。

**Data Manager:** DM is event driven, managing alarm lists, event lists, journals, trend and historical data. It also contains periodic sampling functions, and compression functions used to build large historical data collections. History, trends, alarm and event information can be automatically moved to relational databases by adding an SQL command to the collection. The movement can be periodic or triggered by an event.

**Data Manager:**DMは、アラーム・リスト、イベント・リスト、ジャーナル、傾向および履歴を管理し、イベントドリブンです。それは周期的なサンプリング機能、および大きな履歴データのコレクションを構築するための圧縮機能を含みます。履歴、トレンド、アラームおよびイベントの情報は、コレクションにSQLコマンドを加えることでリレーショナル・データベースに自動的に移動することができます。その移動は周期的もしくはイベントをトリガーに行うことができます。

**Continuous Strategy:** CS is a function block based control. It uses a calculation and logic computation engine based on the periodic execution of functions. CS applications are built by graphically connecting to PIO or OpenControl data for inputs and outputs. CS has a complete set of functions including: analog, discrete, string, and data type conversions. Thousands of successful process applications have been developed.

**Continuous Strategy (連続戦略) :** CSは関数ブロックに基づいた制御です。それは関数の周期的な実行に基づいた計算と論理の演算処理エンジンを使用します。CSアプリケーションの入出力は、PIOまたはOpenControlのデータをグラフィカルに接続することで構築します。CSは次の関数を含む完全なセットを持ちます:analog, discrete, string, および data type conversions.

**OpenControl:** OC is a high-speed sequential control engine with high-speed drivers. Both OC and its drivers run under the Imagination Systems Hyperkernel. OC is classified as a "softPLC" with IEC1131 compatibility and is programmed using a visual flowchart. Variables are accessible from Paragon's clients both locally and over a network using either the 1131 DRV name or an alias. OC drivers provide an alternative way to interface external I/Os to Paragon's client and server components.

**OpenControl:** OCは高速ドライバによる高速なシーケンシャル制御エンジンです。OCとそのドライバの両方がImagination SystemsのHyperkernelで実行されます。OCはIEC1131互換の"softPLC"に分類されて、ビジュアル・フローチャートを使用してプログラムします。Paragonクライアントは、1131 DRV名もしくは別名(alias)を使用して、ローカルとネットワーク上の両方の変数にアクセス可能です。OCドライバはParagonのクライアントとサーバのコンポ

ーネットに外部I/Oをインターフェースする代替の方法を提供します。

## Servers Are Function Based

サーバは関数ベースです。

All Paragon servers are function based (except for OC). In IEC 1131 terms, Paragon's functions fall under the classification of extended "function blocks." Functions are referred to as "objects" in some contexts. Paragon's functions perform a specific task such as:

Paragonサーバはみな関数に基づいています（OCを除いて）。IEC 1131用語で、Paragon関数は拡張“関数ブロック” (extended “function blocks”)に分類されます。関数はいくつかのコンテキスト内で“オブジェクト”と呼ばれます。Paragon関数は次のような特定タスクを実行します：

**Adding two signals to generate a sum.**

2つの信号を加えて合計を生成する

**Modifying a control signal to maintain a setpoint.**

セットポイントを維持するよう制御信号を変更する

**Parsing a string from a barcode reader.**

バーコード・リーダーからの文字列(string)を解析する

**Collecting history into a buffer.**

バッファに履歴を集める

**Maintaining a current alarm collection.**

現在のアラーム収集を維持する

When you build a Paragon server application you are adding and configuring functions, to implement a process strategy. All functions have elements (also called attributes or properties) which have default values.

Paragonサーバ・アプリケーションを構築するには、プロセス戦略を実装するために関数を加えて構成します。すべての関数には、デフォルト値を持つ要素 (attributeあるいはpropertieと呼ばれる) があります。

# Paragon Clients

## Paragonクライアント

A Paragon client is any function that accesses Paragon's servers' data. Some Paragon servers can access other server's data. They are still classified as servers, because they implement server access interfaces. The most common clients are:

ParagonクライアントはParagonサーバのデータにアクセスするあらゆる関数です。いくつかのParagonサーバが他のサーバのデータにアクセスすることができます。サーバ・アクセス・インターフェースを実装するので、それらはサーバとしてまだ分類されます。最も一般的なクライアントは次のとおりです：

**Operator Interface :** OI allows you to create and format operator displays without programming or scripting. You select objects from a palette and connect their attributes to server data. OI can also perform actions on servers such as setting values to trigger processes or change setpoints. OI displays are portable across Windows and OS/2 platforms. If they are moved on OS/2 they have to be resaved.

**Operator Interface :** OIはプログラムやスクリプトを作成せずに操作画面の作成と形式設定を行います。パレットからオブジェクトを選び、サーバ・データにそれらの属性(attribute)を接続します。さらにOIは、プロセスのトリガーあるいはセットポイントの変更に値を設定して、サーバを動作させることができます。OIディスプレイはWindowsとOS/2のプラットフォームに対応します。OS/2上に移動する場合、再保存する必要があります。

**Interactive Trends:** Itrends are activated from an action on an OI display and are configured on-the-fly. You can select DM history or trend data to plot, select zoom, mark, etc. Itrends allow you to plot historical data from an ODBC compliant database as long as it supports adequate date and time stamps. Itrend configurations can be saved and restored.

**Interactive Trends:** ItrendsはOIディスプレイへのアクションで活性化され、オンザフライ（大急ぎ）で構成されます。DM履歴の選択あるいはトレンド・データのプロット、ズームの選択、マーク、などを行えます。Itrendsは、適正な日時スタンプがサポートされる限りODBC対応データベースから履歴データをプロットすることが可能です。Itrendsの設定は保存し復元することができます。

**Engineering Interface:** EI allows you to browse Paragon's server data to validate server applications without having to build OI displays. You can produce displays of all runtime elements for a function and trend selected data. EI layouts can be saved and restored from the EI. EI is an effective tool for learning about Paragon's extensive data type support, and how clients view the server data. Also, you can analyze DM history collections.

**Engineering Interface:** EIは、OIディスプレイを構築する必要なしに、サーバ・アプリケーションを有効にするためにParagonサーバのデータをブラウズします。関数とトレンドの指定データのすべてのランタイム要素のディスプレイを作ることができます。EIレイアウトはEIから保存し復元することができます。EIは、Paragonの広範囲なデータ・タイプのサポートと、クライアントがどうサーバデータを見るかに関して学ぶための有効なツールです。さらに、DM履歴コレクションを分析することができます。

**Recipes:** Recipes load server values from a text file or a relational database table. It is run as a program from OI, a command line, or any third-party application which runs a program.

**Recipes:** レシピは、テキストファイルあるいはリレーショナル・データベース・テーブルからサーバー値をロードします。それは、OIやコマンドラインからのプログラム、あるいはプログラムを実行するあらゆるサードパーティのアプリケーションとして実行されます。

## Building Clients With RAD Tools

### RADツールを備えたクライアントの構築

Development tools such as MS Visual Basic, PowerSoft, PowerBuilder and Borland Delphi are classified as Rapid Application Development (RAD) tools. They have evolved into powerful client-development environments. Typically, they require more programming skills than building clients with Paragon's OI. However, Paragon ActiveX components minimize programming allowing you to build multiple displays. This is done by pasting objects from a palette and selecting Paragon's server data with a browser.

MS Visual Basic, PowerSoft, PowerBuilderおよびBorland Delphiのような開発ツールは、迅速なアプリケーション開発(Rapid Application Development:RAD)ツールとして分類されます。それらは強力なクライアント開発環境に発展しました。通常、これらはParagonのOIでのクライアント構築よりも多くのプログラミング技能を必要とします。しかしながら、Paragon ActiveXコンポーネントは複数ディスプレイを構築するプログラミングを最小にします。これは、パレットからオブジェクトを貼って、ブラウザからParagonサーバのデータを選択することにより行われます。

Java 1.1 and Internet technology is extending this same concept to provide excellent remote and multi-platform access; reducing client maintenance costs. Java tools are available from several vendors Lotus BeanMachine1.1, Borland JBuilder, Symantec Visual Café, IBM Visual Age Java, etc. Paragon 5.0 and greater include JavaBean components that can be used with Java RAD tools to create applets for web browsers with Java support, or Java standalone applications. This is an emerging technology that NemaSoft is focused on developing with BeanMachine, JBuilder and Visual Age for Java. Visual Café and others will be validated in time.

優れたリモート及びマルチプラットフォームのアクセスを提供するためにJava 1.1とインターネットの技術はこの共通のコンセプトを発展しています；クライアントのメンテナンス・コストの縮小。Javaツールは、Lotus BeanMachine1.1, Borland JBuilder, Symantec Visual Café, IBM Visual Age Java,など、さまざまなベンダーから利用できます。Paragonは5.0以上で、Javaをサポートしウェブ・ブラウザ用アプレットを作成するJava RADツールと共に使用することができるJavaBeanコンポーネント、あるいはJavaのスタンド・アロンのアプリケーションを含みます。これは開発を行っているNemaSoftがBeanMachine, JBuilder, およびVisual Age for Javaと共に集中している未来技術です。 Visual Cafeおよび他のものは適切な時期に確認されるでしょう。

It seems clear that component-based client development is more effective than using proprietary scripting. With Paragon's components little programming is necessary, but when it is, a standard language is more effective. For Windows MS Visual Basic is more effective. Java is more appropriate for Web Browsers and non-Windows' platforms. コンポーネントに基づいたクライアント開発が専用のスクリプトを作成するより効果的なのは明確に見えます。Paragonのコンポーネントにおいてプログラミングはほとんど必要ありませんが、必要な場合は標準言語がより効果的です。WindowsではMS Visual Basicがより効果的です。Javaはウェブ・ブラウザおよびWindows以外のプラットフォームでより適切です。

# Anatomy Of A Paragon Application

## Paragonアプリケーションの構造

After logging in, you are ready to begin developing and running a Paragon application. A Paragon application consists of the following:

ログインした後、あなたはParagonアプリケーションの開発と実行を開始する準備ができています。Paragonアプリケーションは下記から構成されます：

Two Process I/O files created with the Process I/O builder:

Process I/O builderで作成する2つのProcess I/Oファイル：

**0 The Library File (\*.PID)** contains all the Process I/O processes and functions. A process is a grouping of Paragon functions and interconnections in a given subsystem. A function consists of a single Paragon operation for device communications, computations, control algorithms, data collection, and alarm monitoring. Connections are made between functions in the same or different subsystem to provide interaction between different parts of the application and to produce data flow.

ライブラリ・ファイル(\*.PID)は、Process I/Oのプロセスと関数をすべて含みます。プロセスはある特定のサブシステムのParagon関数と相互連結のグループ分けです。関数は、装置通信、演算処理、制御アルゴリズム、データ・コレクション、そして警報監視のうち1つのParagonオペレーションから構成されます。アプリケーションの異なる部分間の相互作用を提供し、かつデータ・フローを作り出すために、接続は、同一もしくは異なるサブシステムの関数の間でなされます。

**0 The Configuration File (\*.PIC)** is used as the start-up file for the current subsystem. It defines which processes execute when you run the application (run-time).

設定ファイル(\*.PIC)は現在のサブシステムをスタートアップするためのファイルとして使用されます。アプリケーション(ランタイム)を実行するとき、プロセスが何を実行するのかを定義します。

Both files must be present for your Process I/O engine to acquire data.

2つのファイルは、データを取得するProcess I/Oエンジンに存在するに違いありません。

Two Operator Interface files created with the Operator Interface builder:

Operator Interface builderで作成する2つのOperator Interfaceファイル：

**0 The uncompiled Operator Interface file (\*.OIL)** is used to modify the application.

コンパイルされていないオペレーター・インターフェース・ファイル(\*.OIL)はアプリケーションを修正するために使用されます。

**0 The compiled Operator Interface file (\*.OID)** is used to run the application.

コンパイルされたオペレーター・インターフェース・ファイル(\*.OID)はアプリケーションを実行するために使用されます。

For advanced Data Manager functions two files must be created with the Data Manager builder:

高度なデータ・マネージャ関数のために、2つのファイルがData Manager builderで作成される必要があります：

**0 The Library file (\*.DMD).**

ライブラリ・ファイル(\*.DMD)

**0 The Configuration file (\*.DMC).**

設定ファイル(\*.DMC)

Both files must be present for the Data Manager to log data. If default Data Manager alarms and events are used, then these files are not needed.

2つのファイルはデータ・マネージャがデータを記録するために存在しなければなりません。デフォルトのデータ・マネージャのアラームとイベントが使われる場合、これらのファイルは必要ありません。

Three Continuous Strategy files created with the Continuous Strategy builder:  
Continuous Strategy builderで作成する3つの連続戦略(Continuous Strategy)ファイル:

0 The Graphics file (\*.CSB) contains the graphics used to create your strategy.

グラフィックス・ファイル(\*.CSB)は、あなたの戦略を作成するために使用されるグラフィックスを含みます。

0 The Configuration file (\*.CSC).

設定ファイル(\*.DMC)

0 The Library file (\*.CSD).

ライブラリ・ファイル(\*.DMD)

Additionally you can create a CSL file which creates a portable file to move between Windows (95 and NT) and OS/2.  
さらに、ウインドウズ(95とNT)とOS/2の間に移動するためにポータブルファイルを作成するCSLファイルを作成することができます。

One Application file (\*.APP) is created with the Application Manager. All the information on the various files of a Paragon Application is kept here. The Application file contains the following information:

Application Managerで作成する1つのアプリケーション・ファイル(\*.APP)。Paragonアプリケーションの様々なファイルに関するすべての情報がここに保持されます。アプリケーション・ファイルは次の情報を持ちます。

0 The start-up file for each subsystem utilized in the application.

アプリケーションで利用される各サブシステムのスタートアップ・ファイル。

0 The serial port parameters.

シリアル・ポートのパラメタ。

0 The local node name.

ローカルのノード名。

Application files allow users with multiple applications to quickly change the system configuration by selecting another Application file.

アプリケーション・ファイルによって、複数のアプリケーションを持ったユーザは、別のアプリケーション・ファイルを選択することにより、瞬時にシステム構成を変更することが可能になります。

Data Manage internal storage files (\*.DMS).

データ管理内蔵記憶装置ファイル(\*.DMS)。

Two additional files Paratnt.TEC and Paratnt.USR

2つの追加ファイルParatnt.TEC と Paratnt.USR 。

0 **Paratnt.TEC:** is configured to your hardware key. It contains your serial number, which is checked when you start Paragon. If the serial number of the key does not match the serial number contained in the tec file, you cannot run Paragon. It also contains product information including the number of enablers, and the type of package you purchased.

**Paratnt.TEC:** はあなたのハードウェア・キーに設定されます。それはシリアル番号を持ち、Paragonを開始すると、それがチェックされます。キーのシリアル番号がtecファイルの持つシリアル番号と一致しない場合、Paragonを実行することができません。また、それはenablerの数、および購入パッケージのタイプを含む製品情報を持っています。

0 **Paratnt.USR:** contains security settings and enabler settings. When you enable or disable Enabler options, the usr file records the information. When you get a new .usr file, the enabler and security settings are set back to their defaults.

**Paratnt.USR:** はセキュリティ設定およびenabler設定を持ちます。Enablerオプションを有効あるいは無効にする時、usrファイルは情報を記録します。新しい.usrファイルを得る時、enablerとセキュリティの設定はそれらのデフォルトに戻ります。



**These Paragon packaging files must reside in the start-in directory.**

これらのParagonに実装されるファイルはstart-inディレクトリに存在するに違いありません。

**By default the files reside in the Application directory (APP)**

デフォルトで、ファイルはアプリケーション・ディレクトリ(APP)に存在します。

C:¥Paratnt¥App.

# Paragon Quick Start

## In This Chapter:

### • Starting Paragon

Paragonをはじめよう

### • Application Wizard

アプリケーション・ウィザード

### • Chapter Summary

章の要約

## 28 i Paragon Getting Started

This quick start gives you an overview of what constitutes a Paragon application using the New Application wizard. このクイック・スタートは、新しいアプリケーション・ウィザードを使用して、Paragonアプリケーションを構成するための概要を提供します。

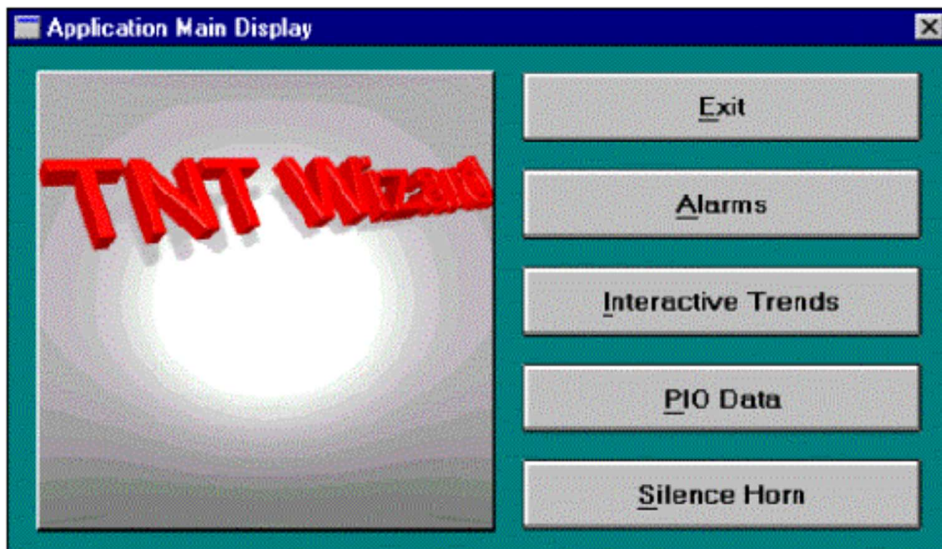
**(OS/2 USERS CANNOT COMPLETE THIS SECTION** since there is no Application Wizard for the OS/2 operating system. If you are using OS/2, read through this section for an understanding of Paragon and the key terms used throughout this manual. To do the steps in this manual, complete the OS/2 section in the next chapter). You will build an application step-by-step and learn about Paragon as you go. Paragon has a very capable, comprehensive set of application tools. Therefore, you cannot learn about all of it in a short look. Once you have a running application you can extend it in ways that are not covered in this quick start. You can obtain this information from the User's Guide and Reference Manual.

If you are running the Paragon Test Drive, please read the warning in the introduction indicating that the application will not work with a purchased product. However, the work you are doing here is a small investment of time that can easily be repeated after purchasing Paragon.

Paragon Test Driveで実行する場合は、アプリケーションが購入された製品で動作しないことを示すイントロダクションの警告を読んでください。しかし、ここで行う作業はParagonを購入した後で簡単に繰り返せる小さな時間の投資です。

You will create a Main display that loads three other displays by clicking on the appropriate button. These include an Interactive Trend, a Process I/O Data Log, and an Alarm Log. The main display looks like this:

適切なボタンをクリックして他の3つの画面をロードする主画面を作成します。これらはインタラクティブ（対話型）な傾向、Process I/Oデータログおよびアラームログを含みます。メイン画面は次のように見えます：




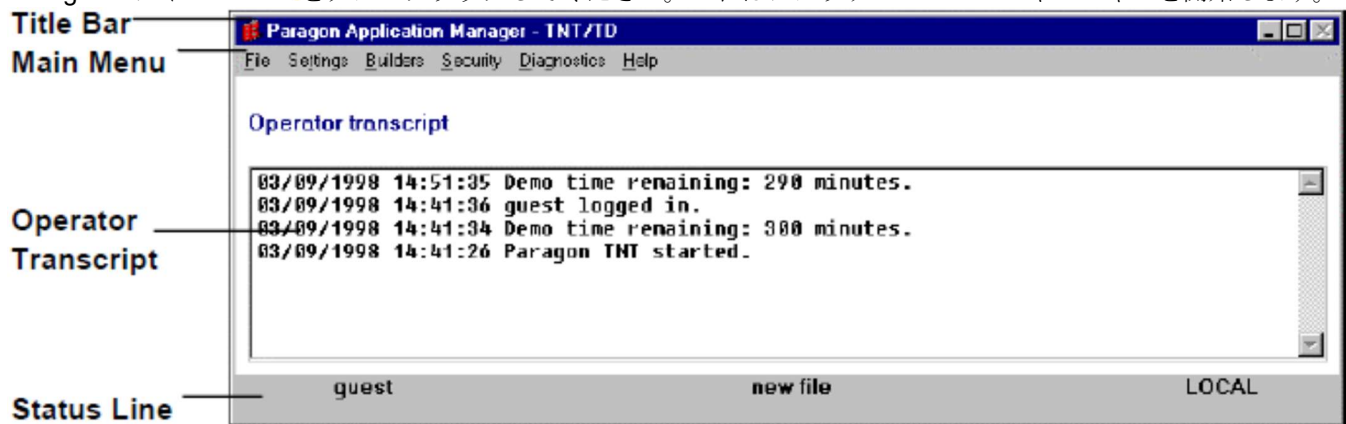
## Starting Paragon Paragonをはじめよう

Almost all Paragon work is started from the Application Manager.  
ほとんど全てのParagonの作業はApplication Managerから始めます。

**To start Paragon:**  
**Paragonをはじめするには :**

1. Double-click on the **Paragon Icon**: . This initiates the Application Manager.

Paragonのアイコン :  をダブルクリックしてください。これはアプリケーション・マネージャーを開始します。



2. From the Main menu select **Security**►**Login**. When the Paragon Login window appears type Guest in the User Name field, then click on **OK**.

メイン・メニューから、**Security**►**Login**を選択します。その後、Paragon Loginウィンドウがあらわれたら、User Name欄にGuestとタイプして、**OK**をクリックしてください。

The Paragon Application Manager controls security access to various operations and application building functions. You can define a group with specific access privileges and then assign individuals to an access group.

Paragon Application Managerは、様々な操作とアプリケーション構築の機能へのセキュリティ・アクセスをコントロールします。特定のアクセス権を備えたグループを定義し、次に、アクセス・グループに個人を割り当てることができます。

**To see how the Application Manager controls security access, do the following:**

**Application Manager**がどのようにセキュリティ・アクセスをコントロールするか確かめるには、下記を行ってください:

1. Select **Security**►**Edit Access Groups**.

**Security**►**Edit Access Groups**を選択します。

2. Select a name from the Group list box, then click the Parameters button to see the current configuration.

Group list boxから名前を選んで、次に、現在の設定を見るためにParametersボタンをクリックしてください。

There are several system rights that are permanently defined and 16 custom rights. You can define the meaning of the custom rights using the **Security**►**Login** command. Rights are group based rather than level based, allowing groups to be mutually exclusive. Some of Paragon's other security features include:

あらかじめ決められている幾つかのシステム権限および16のカスタム権限があります。 **Security**►**Login** コマンドを使

って、カスタム権限に意味を定義できます。権限はレベル・ベースではなくグループ・ベースで、グループが相互に排他的であることを可能にします。その他にParagonは次のセキュリティ機能を持っています：

**Establishing a password to protect your entire application.**

すべてのアプリケーションを保護するためパスワードを設定する。

**Encrypting your application to work with a specific protection key.**

特定のプロテクトキーでのみ動作するようにアプリケーションを暗号化する。

**Protecting individual database files.**

個々のデータベース・ファイルを保護する。

## 30 i Paragon Getting Started

# Application Wizard

A Paragon application consists of several files depending on which components you are using. All these files are typically stored in one directory for your convenience. The Wizard will do all of the following for you:

Paragonアプリケーションは、どのコンポーネントを使用するかによって依存するいくつかのファイルから成ります。これらのファイルはすべて、あなたの利便性に備えて通常は1つのディレクトリに保存されます。ウィザードはあなたのために下記をすべて行います：

Create a new application directory.

新しいアプリケーション・ディレクトリを作成

Copy Paragon configuration files to that directory

そのディレクトリへのParagonコンフィギュレーション・ファイルのコピー  
(Paratnt.TEC and Paratnt.USR).

Make start-up application files that can later be exported into a full application.

完全なアプリケーションにエクスポートするためのスタートアップ・アプリケーション・ファイルの作成。

Create an icon that allows you to re-start Paragon or the application in that directory.

そのディレクトリ内でParagonまたはアプリケーションを再開するためのアイコンの作成。

The Application Wizard uses a “driver database” that determines which process types can be configured, and the required parameters for creating tag names. A tag name consists of the following format:

Application Wizardは、どのプロセス・タイプを形成することができるかを決める“driver database”およびタグ名を作成するための必要なパラメーターを使用します。タグ名は次のフォーマットから成ります：

### Subsystem.Process.Function.Element

**Subsystem Name:** Identifies the Paragon subsystem containing the named value either PIO, CS, DM, OI, AM, OC or OC\_DRV (OpenControl Direct Representation Variable).

**Subsystem Name:** Paragonサブシステムを識別するためのPIO, CS, DM, OI, AM, OCもしくはOC\_DRV(OpenControl Direct Representation Variable)の値を含む名前。

**Process Name:** Is designated by the user and can be up to twelve characters. A process is a grouping of Paragon functions and interconnections that form a part of an application library in a given subsystem. Processes can be designated to run on different stations of a local area network (LAN).

**Process Name:** ユーザ指定で12文字以内。プロセスは、与えられたサブシステム内のアプリケーション・ライブラリの部品を形成するParagon機能および相互接続のグループ設定です。プロセスは、ローカル・エリア・ネットワーク(LAN)の異なるステーション上で実行するように指定できます。

**Function Name:** Is designated by the user and can be up to twelve characters. A function consists of a single Paragon operation for device communications, computations, control algorithms, data collection, and alarm monitoring. To provide interaction between different parts of the application and produce data flow, connections are made between functions in the same or different subsystems.

**Function Name:** ユーザ指定で12文字以内。機能には、装置コミュニケーション、計算、コントロール・アルゴリズム、データ収集およびアラーム監視のための単一のパラゴン・オペレーションから成ります。適用の異なる部分間の相互作用を提供し、かつデータ・フローを生産するために、接続は同じか異なるサブシステムの中で機能間でなされます。

**Element Name:** Is the label used to reference a particular data value within a Paragon function. The last name “ReadIn” is a Paragon name for the input element of a PIO function. Typical element names are In, Out, Description, HiRange, and HiAlarm.

**Element Name:** Paragon関数内の特別なデータ値を参照するために使用されるラベルです。最後の名前“ReadIn”は、PIO関数の入力要素のParagon名です。一般的な要素名はIn, Out, Description, HiRangeおよびHiAlarmです。

Refer to the User's Guide and Reference Manual for a full description of element names.

要素名の詳細な記述に関してはユーザガイドおよびリファレンスマニュアルを参照してください。

An example of a tag name would be:


タグ名の例は次のとおりです：

### **PIO.Reactor1.LIC100.ReadIn{0}**

This instructs Paragon to "Read In" the information from the Process I/O subsystem, Reactor1 process, LIC100 function to perform a specified action. For example, this tag name could be collecting data from a tank's rising contents. By connecting it to the appropriate actions in the Data Manager, you could plot the data on a trend device.

この命令はParagonにProcess I/OサブシステムのReactor1プロセスのLIC100関数で指定されている動作の実行の情報を"Read In"させます。例えば、このタグ名はタンクの増加する中身からデータを収集することかもしれません。Data Managerの適切な動作に繋げることで、傾向装置にデータをプロットできます。

**Note:** It is recommended that you use the **Chevron**: to select point names. The Chevron allows you to select choices from a proceeding menu, preventing spelling errors. A misspelled tag name will cause an error in the application.

**注意：**ポイント名の選択には**Chevron**: の使用を薦めます。Chevronは進行メニューから選択を認めて、あなたのスペルミスを防ぎます。タグ名のミススペルは、アプリケーションにエラーを引き起こします。

All Paragon clients, ActiveX and JavaBeans use this same naming scheme for accessing Paragon server data. However, while building an application you seldom have to type in a full name like the one above. Paragon's builders query library files and present you with selectable lists. Paragon uses the full name and does not use indexes, offsets or other non-robust methods. Therefore, you should choose your process and function names carefully the first time. すべてのParagonクライアント、ActiveXそしてJavaBeansは、Paragonサーバのデータにアクセスするために、これと同じ命名スキームを使用します。しかし、アプリケーションを構築する間、上記のようなフルネームをタイプする必要はめったにありません。Paragonのビルダーはライブラリのファイルについて質問して、選択可能なリストを与えます。Paragonはフルネームを使用して、インデックス、オフセットや他の着実でない方法は使用しません。したがって、あなたは最初にプロセスと関数の名前を注意深く選ぶべきです。

## Starting The Application Wizard

### Application Wizardの開始

**To start the Application Wizard:**  
**Application Wizardをはじめるには：**

1. Select **File**Application Wizard.

**File**Application Wizardを選択します。

2. Click in the field below **Enter A Name For the Application.**

**Enter A Name For the Application**項目をクリックしてください。

Name your application Firstapp. The name can be up to 8 characters long, but cannot include ".", "{", "}" characters. A name is used to:

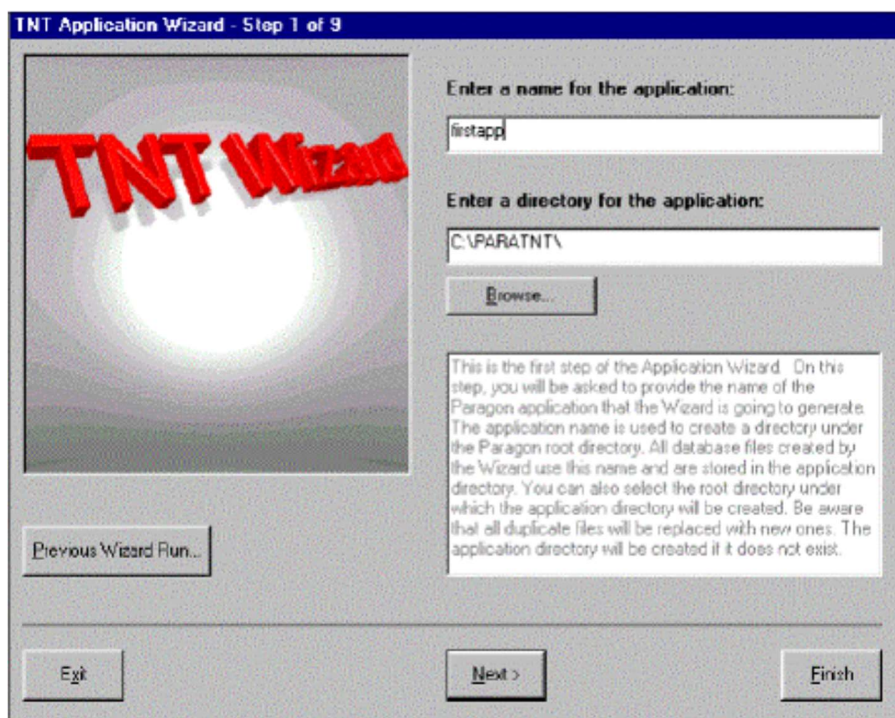
アプリケーションをFirstappと命名してください。名前は長さ8文字以内で".", "{", "}"は含められません。名前は次のものに使用されます：

0 Name the files created by the Paragon Application Wizard.

Paragon Application Wizardで作成されるファイルに名前を付けます。

0 Create a subdirectory where all the application files are stored.

すべてのアプリケーション・ファイルが保存されるサブディレクトリを作成します。



**Note:** When building applications in the future you should create a separate directory inside the Paragon directory for each application. Paragon installs a default working directory called APP inside the main Paragon directory. The full path is C:\PARATNT\APP.

**注意:** 今後アプリケーションを構築する場合、Paragonディレクトリ内に各アプリケーションの個別ディレクトリを作成する必要があります。Paragonは、メインのParagonディレクトリ内にAPPと呼ばれるデフォルト作業ディレクトリをインストールします。フルパスはC:\PARATNT\APPです。

3. To proceed to the next section, click on **Next**.

次のセクションに移るため、**Next**をクリックしてください。

**Previous Wizard Run:** If you want to retrieve a previous Application Wizard file. The current Wizard loads all its values including:

**Previous Wizard Run:** 以前のApplication Wizardファイルを検索する場合、現在のウィザードは、次を含むその値をすべてロードします:

0 The application name and directory.

アプリケーションの名前とディレクトリ

0 The Process I/O name and type.

Process I/Oの名前とタイプ

0 Previously generated tag names.

以前に生成されたタグ名

**Note:** If a previous default run file LAST.WIZ is loaded, you cannot change the type of process in Step 2. You can specify a different filename to store the Wizard run information at the end of the Wizard's execution.

**注意:** 以前のデフォルトの実行ファイルLAST.WIZがロードされる場合、ステップ2のプロセスでタイプを変更することはできません。ウィザードの実行の終わりにウィザード動作時の情報を格納するために異なったファイル名を指定することができます。

**Intro:** Displays a dialog window containing information about the program.

**Intro:** プログラムに関する情報を持ったダイアログ・ウィンドウを表示します。



**Finish:** Halts the Wizard. Information entered up to that point is saved. The wizard will create a new subdirectory under the PARATNT¥APP directory.

**Finish:** ウィザードを停止します。それまで入力した情報は保存されます。ウィザードはPARATNT¥APPディレクトリ内に新しいサブディレクトリを作成します。

**Exit:** Terminates the Application Wizard, and no information is saved.

**Exit:** Application Wizardを終了しますが、情報は保存されません。

## Process I/O Setup

### Process I/Oセットアップ

**Process I/O:** interfaces Paragon to external measurement and control devices. PIO supports multiple drivers that communicate through serial ports, PC adapters, TCP/IP, etc. PIO data is displayed by clients, typically without additional processing, and are accessed over a network

**Process I/O:** Paragonと外部測定と制御装置のインターフェース。PIOは、シリアルポート、PCアダプター、TCP/IPなど通信のための多くのドライバをサポートします。PIOデータは、通常は追加の処理なしでクライアントに表示され、ネットワーク上でアクセスされます。

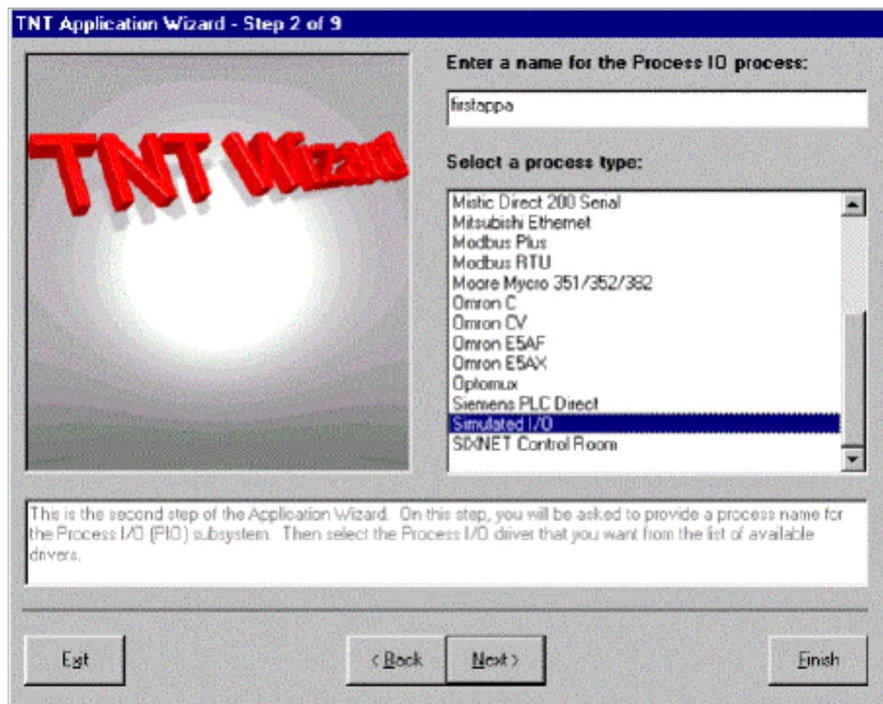
Step 2 is used to setup a process for the Process I/O (PIO) subsystem, and determine the type of hardware that is used.

ステップ2では、Process I/O(PIO)サブシステムのプロセスをセットアップして、使用されるハードウェアのタイプを決定します。

#### To setup a process for the Process I/O:

**Process I/Oのプロセスをセットアップするには：**

1. Leave the default name Firstappa as the name for your application. The name can be up to 12 characters long. アプリケーション名はデフォルトの名前Firstappaを残します。名前は長さ12文字以内です。
2. From the **Select A Process Type** list box select the type of hardware you are currently using. If it is not listed select Simulated I/O. This will allow you to complete the steps outlined in manual. Your process type can be added by contacting your NemaSoft Sales Representative.  
**Select A Process Type** リストボックスから、現在使用しているハードウェアのタイプを選んでください。それが記載されていない場合は、**Simulated I/O**を選択してください。これで、あなたはマニュアルで概説されたステップを終了します。あなたがプロセスタイプを追加したければ、あなたを担当するNemaSoftセールス代表に連絡してください。
3. Click on **Next**.  
**Next**をクリックしてください。



## Communications Port

### 通信ポート

Step 3 is used to configure the communication parameters for the Process I/O driver type. This step varies according to what process type you selected from the previous screen. Some drivers, such as the Simulated I/O, will not show a step 3 at all. The communication information does not need to be set.

ステップ3では、Process I/Oドライバ・タイプのために通信パラメータを設定します。ここまでの画面で選択したプロセスのタイプにより、このステップは異なります。Simulated I/Oなど幾つかのドライバはステップ3の全てを示しません。通信情報の設定は必要ありません。

**Select a Communications Port:** is used to select the port your hardware is on. This could also be a plug-in card.

**Select a Communications Port:** ハードウェアを接続するポートの選択に使用します。これはプラグインのカードかもしれません。

**Select a Baud Rate:** specifies the speed of transmission that is used with the communications port.

**Select a Baud Rate:** 通信ポートで使用する転送速度を指定します。

**Select The Number of Data Bits:** specifies the number of data bits to be used with the communications port.

**Select The Number of Data Bits:** 通信ポートで使用するデータ・ビットの数を指定します。

**Select The Number of Stop Bits:** specifies the number of stop bits to be used with the communications port.

**Select The Number of Stop Bits:** 通信ポートで使用するストップビット数を指定します。

**Note:** This information can be found in the PIO online help for the specific driver.

**注意:** 特定のドライバのためにPIOオンラインヘルプでこの情報を見つけることができます。

## Tag Names

**Comma Separated Variable:** a standard ASCII (text) format that most third party packages can interpret or generate. This allows you to create recipe files in host computers and either download them to Paragon or upload them into host programs.

**カンマ区切り値(Comma Separated Variable):** ほとんどのサードパーティ製パッケージが解釈および生成することのできる標準のASCII (テキスト) フォーマット。これにより、ホストコンピュータでレシピファイルを作成して、それをParagonにダウンロードするか、それらをホストプログラムにアップロードできます。

**Recipes:** Load server values from a text file or a relational database table.

**レシピ(Recipe):** テキストファイルからレシヨナル・データベース・テーブルからサーバの値をロードします。

Step 4 is used to create tag names for the Process I/O. You can either import the tags from a **Comma Separated Variable (CSV)** file or generate them using the Application Wizard. A CSV format is a standard ASCII (text) format that most third party packages can interpret or generate. This allows you to create **Recipe** files in host computers and either download them to Paragon or upload them into host programs.

ステップ4では、Process I/Oのタグ名を作成します。Application Wizardを使ってカンマ区切り (CSV)ファイルからタグをインポートしたり、それを生成することができます。CSV形式はほとんどのサードパーティ製パッケージで解釈や生成することのできる標準のASCII(テキスト)フォーマットです。これで、ホストコンピュータでレシピファイルを作成して、それをParagonにダウンロードしたり、それをホストプログラムにアップロードしたりできます。

### To create Tag Names:

タグ名を作成するには :

1. Click on the **Generate** button, and the Tag Wizard will appear.

**Generate** ボタンをクリックすると、Tag Wizardがあらわれます。

**CSV:** imports tag names from a CSV file.

**CSV:** CSVファイルからタグ名をインポートします。

**Generate:** launches the Paragon Tag Wizard to generate tag names.

**Generate:** タグ名を生成するためにParagon Tag Wizardを起動します。

## Tag Generation Wizard

The Tag Generation Wizard is used to specify the types of tags that are created. The title of the Tag Generation display includes the process type selected from Step 2.

Tag Generation Wizardで、作成するタグのタイプを指定します。Tag Generationに表示されるタイトルには、ステップ2で選択したプロセス・タイプが含まれます。

### To create the tag names:

タグ名を作成するには：

1. To create the first part of the tag name select **LIC** from the **Tag Prefix** field.

タグ名の最初の部分を作成するために、**Tag Prefix**項目から**LIC**を選択してください。

2. To create the second part of the tag name (suffix) leave the **Starting Increment** at 100. Step 1 and Step 2 produce LIC100 as the first part of your tag.

タグ名の第2部分(接尾語)の作成では、**Starting Increment**に100を残してください。ステップ1とステップ2でタグの最初の部分としてLIC100が形成されています。

3. Since you are only creating one tag, leaving the tag increment at 1 will not affect the tag name. The tag increment causes the suffix for each new tag to increase by the value selected:

1つのタグしか作成していないので、tag increment (タグ増分)を1にしてもタグ名に影響しません。tag incrementはそれぞれの新しいタグのために選択した値で増加した接尾語を引き起こします：

LIC100, LIC101, LIC102 etc.

**Analog:** any value with magnitude, such as voltage, temperature, or numeric.

**Analog:** 電圧、温度あるいは数値のような大きさを持つあらゆる値。

4. Select **Analog** as the Function Type. An analog value is any value with magnitude, such as voltage, temperature, or numeric.

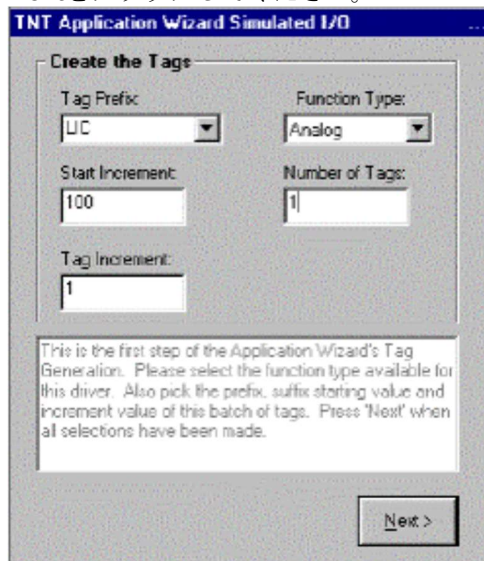
関数タイプで**Analog**を選択してください。アナログの値は電圧、温度、または数値などの大きさがあるあらゆる値です。

5. Type 1 as the number of Tags to generate.

作成するタグ数(number of Tags)に1をタイプしてください。

6. Click on **Next**.

**Next**をクリックしてください。



The screenshot shows a dialog box titled "TNT Application Wizard Simulated I/O" with a sub-header "Create the Tags". It contains several input fields and a "Next >" button. The fields are: "Tag Prefix" (dropdown menu with "LIC" selected), "Function Type" (dropdown menu with "Analog" selected), "Start Increment" (text box with "100"), "Number of Tags" (text box with "1"), and "Tag Increment" (text box with "1"). Below the fields is a text box with instructions: "This is the first step of the Application Wizard's Tag Generation. Please select the function type available for this driver. Also pick the prefix, suffix starting value and increment value of this batch of tags. Press 'Next' when all selections have been made."

## Setting Parameters For PIO Functions

This step allows you to set predefined parameters for the tags, and generate them using those values. These parameters specify the actions that are associated with each tag.

このステップは、タグのためにあらかじめ定められたパラメーターをセットし、それらの値を使用して、それらを生成することを可能にします。これらのパラメーターは、各タグに関連している動作を指定します。

### To generate the tags:

タグを生成するには：

1. Leave the window at its default values.

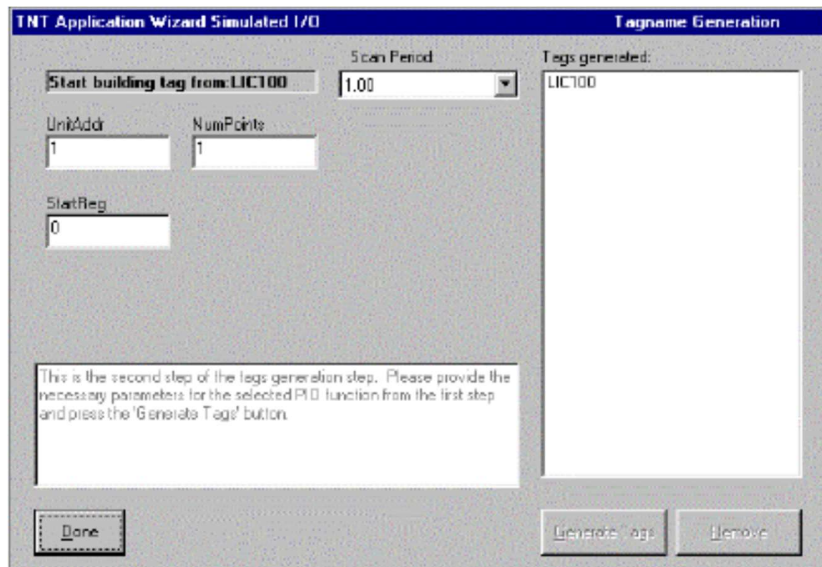
デフォルト値をウィンドウに残してください。

2. Click on **Generate Tags**. Your tags are created in the **Tags Generated** field.

**Generate Tags**をクリックしてください。あなたのタグが**Tags Generated**項目に作成されます。

3. Click on **Done**. This brings you back to Step 4, with one new tag defined. At Step 4 click on **Next**.

**Done**をクリックしてください。1つの新しいタグが定義された状態でステップ4に戻ります。ステップ4で**Next**をクリックしてください。



The screenshot shows a dialog box titled "TNT Application Wizard Simulated I/O" with a sub-tab "Tagname Generation". The dialog contains several input fields and a list box:

- Start building tag from:** LIC100
- Scan Period:** 1.00 (dropdown menu)
- UnitAddr:** 1
- NumPoints:** 1
- StartReg:** 0
- Tags generated:** LIC100

Below the input fields is a text box with the following text: "This is the second step of the tags generation step. Please provide the necessary parameters for the selected PID function from the first step and press the 'Generate Tags' button."

At the bottom of the dialog, there are three buttons: "Done" (highlighted with a dashed border), "Generate tags", and "Remove".

## Selecting Tag Names

**Library File:** contains all the PIO processes and functions.

**Library File:** すべてのPIOプロセスと関数を含みます。

Step 5 allows you to select which tags to generate and add to the PIO subsystem. This is mostly used when importing tags from a CSV file, but also when excluding certain tags. A list of tags generated from Step 4 is shown to the left. Once the tags are added to the **TNT Tag Names** list they are added to the **PIO Library file** once the application is created.

ステップ5では、PIOサブシステムに生成し追加するタグを選択します。CSVファイルからタグをインポートする場合や特定のタグを除外する場合に、普通使われます。ステップ4で生成されたタグのリストが左に示されます。一旦タグが**TNT Tag Names**リストに追加されたり、一旦アプリケーションが作成されたりすると、それらは**PIOライブラリ・ファイル(Library file)**に追加されます。

### To add the Tag Names to Process I/O Library file:

タグ名を**Process I/O**ライブラリ・ファイルに加えるには：

1. Click on **Add** to add the **LIC100** tag to the **TNT Tag names** list.

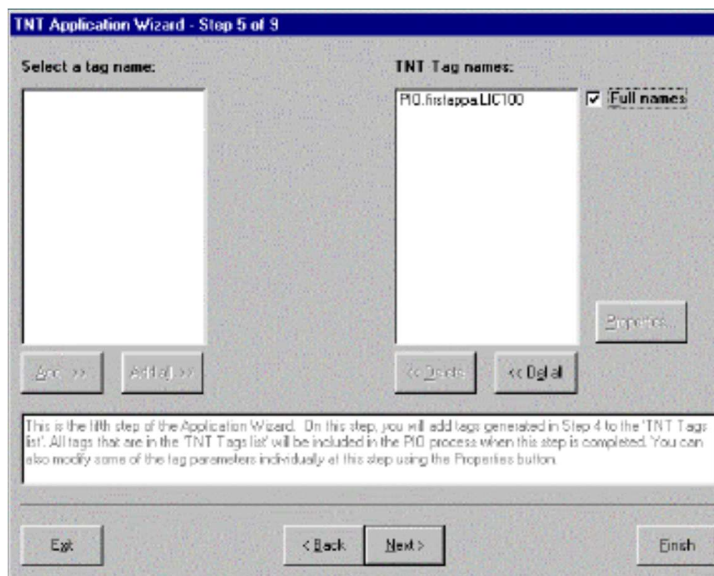
**TNT Tag names**リストに**LIC100**タグを加えるために**Add**をクリックしてください。

2. To see the tag name in the format **Subsystem.Process.Function**, click on the **Full Names** check box.

**Subsystem.Process.Function**のフォーマットでタグ名を見るには、**Full Names**チェックボックスをクリックしてください。

3. Once the tag is added to the **TNT Tag names** list box, click on **Next**.

一旦タグが**TNT Tag names**リストボックスに追加されたら、**Next** をクリックしてください。



## Data Manger History

**Data Manager:** is event driven, managing alarm lists, event lists, journals, trends, and historical data. DM also contains periodic sampling functions and compression functions used to build large historical data collections. History, trends, alarms and events can automatically be moved to relational databases by adding an SQL command to the collection. The movement can be periodic or triggered by an event.

**Data Manager:** アラーム・リスト、イベントリスト、ジャーナル、傾向および履歴データを管理するイベントドリブンです。DMはさらに周期的なサンプリング機能を持ちます。また、圧縮機能はかつては大きな履歴収集を構築しました。履歴、傾向、アラームおよびイベントは、収集にSQLコマンドを加えることによりリレーショナル・データベースに自動的に移動することができます。その移動は周期的もしくはイベントをトリガーとすることができます。

Step 6 is used to configure the three types of Dead band History collections that can be assigned to the **Data Manager**. The History type is added to the Data Manger History collection in the next step, allowing you to vary the information that is collected.

ステップ6では、Data Managerに割り当て可能な履歴収集のための3つのタイプの許容変化幅(Deadband)を形成します。履歴タイプは収集する情報の変更を可能にし、次のステップでデータ管理履歴の収集に加えられます。

### To configure the Data Manager Deadband History type: Data Managerの許容変化幅の履歴タイプは形成するには：

1. Leave the window at its default values.  
デフォルト値をウィンドウに残してください。

2. Click on **Next**.  
**Next** をクリックしてください。



**Select a Type:** changes the parameter values for the selected type. After you select one of the three types, select its parameters from the fields to the right.

**Select a Type:** 選択したタイプに対してパラメーター値を変更します。3つのタイプから1つを選択した後、右の項目からパラメーターを選択してください。



**Scan Rate:** how often Data Manager samples the required points from the PIO. The values are shown in milliseconds. Typically the scan rates are greater than one second.

**Scan Rate:** Data ManagerがPIOから必要なポイントをサンプリングする頻度。値はミリ秒で示されます。通常スキャン・レートは1秒より大きいです。

**Delta:** the amount of change required between two data samples, in order to output a value.

**Delta:** 変化量は、値の出力のため2つのデータ・サンプル間で必要です。

## 38 i Paragon Getting Started

### History Collection Points

Step 7 is used to add tag names to the Data Manager. Each tag name represents a function in a Data Manager History process.

ステップ7では、Data Manager にタグ名を加えます。それぞれのタグ名はData Manager 履歴プロセスの機能をあらわします。

#### To add Tag Names to the Data Manager: Data Managerにタグ名を加えるには：

1. Leave the window at its default values.

デフォルト値をウィンドウに残してください。

2. The **Enter A Name For The DM Process:** field should say Firstappa. The name can be up to 12 characters long, but cannot include “.”, “{”, “}” characters. The name is used to create the DM library file.

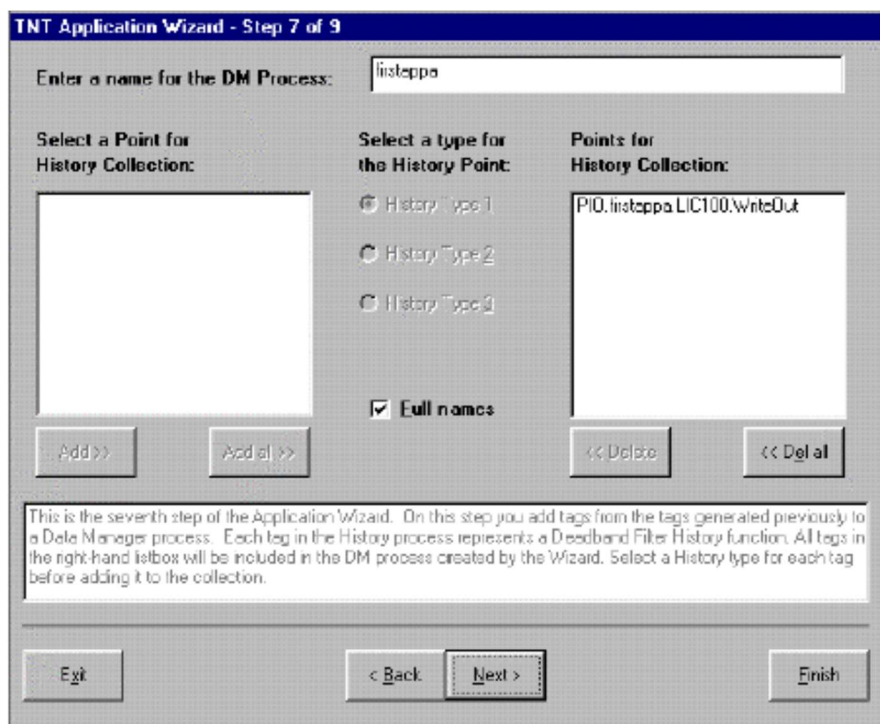
**Enter A Name For The DM Process:**項目はFirstappaであるべきです。名前は長さ12文字以内ですが、“.”, “{”, “}” は含められません。名前はDMライブラリファイルの作成に使用されます。

3. Click on **ADD** to add **PIO.firstappa.LIC100.WriteOut** to the **Points For History Collection** list box.

**ADD** をクリックして、**Points For History Collection**リストボックスに**PIO.firstappa.LIC100.WriteOut**を加えてください。

4. Click on **Next**.

**Next** をクリックしてください。



**Add:** adds the highlighted point from the Select A Point For History Collection list box to the Points For History Collection list box.

**Add:** Select A Point For History Collection リストボックスでハイライトされたポイントをPoints For History Collection リストボックスに加えます。

**Delete:** sends the highlighted point back to the unused points list.

**Delete:** 未使用ポイントのリストにハイライトされたポイントを戻します。

**Del All:** sends all the points back to the unused points list.

**Del All:** 未使用ポイントのリストにすべてのポイントを戻します。

## Operator Interface Display

**Operator Interface:** allows you to create and format operator displays without programming or scripting. OI can also perform actions on servers such as, setting values to trigger processes or change setpoints. OI displays are portable across Windows and OS/2 platforms.

**Operator Interface:** プログラムもスクリプトも作成せずに、操作画面の作成とフォーマットができます。さらにオペレータ・インターフェース(OI)は、トリガー・プロセスや変更セットポイントの値を設定してサーバへアクションを実行することができます。OI画面はWindowsとOS/2のどちらのプラットフォームでも使用できます。

Step 8 is used to specify the **Operator Interface Display Starter Set** that is created as part of your application. This helps you to get started in one of two common directions: OI and ActiveX (Visual Basic).

ステップ8では、アプリケーションの一部として作成する**Operator Interface Display Starter Set**を指定します。これは、2つの共通命令から1つを開始するのを補助します：OIとActiveX(Visual Basic)。

### To select the Operator Interface Display Starter Set:

**Operator Interface Display Starter Set**を選択するには：

1. Select the **OI Starter Set**.

**OI Starter Set**を選択してください。

2. Click on **Next**.

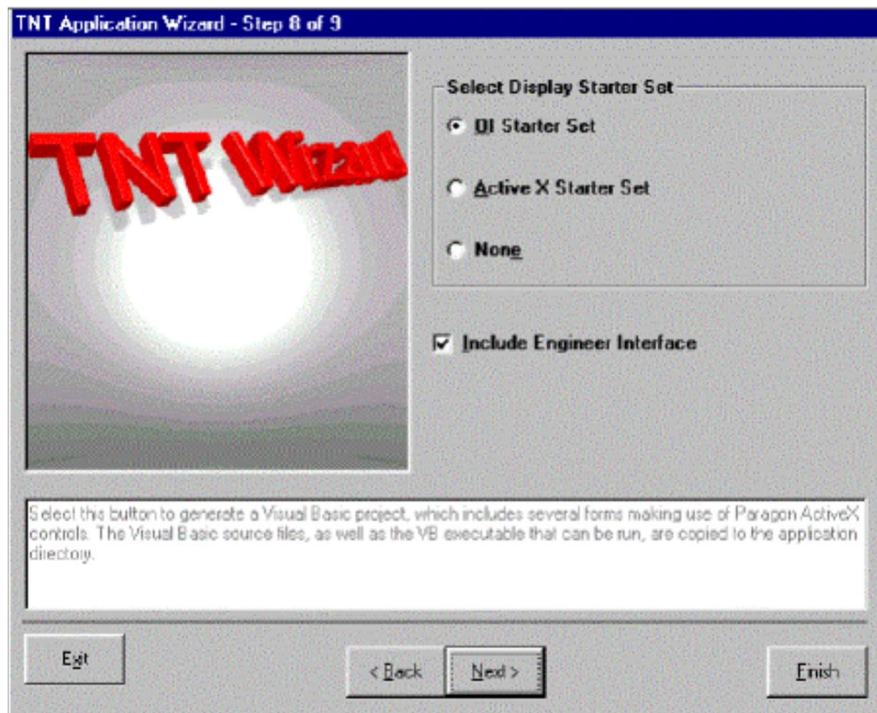
**Next** をクリックしてください。

**OI Starter Set:** includes the Paragon Operator Interface (OI) display files created by the Application Wizard. Four displays will be created including: a Main display, an Alarm display, a Journal display and a PIO data display.

**OI Starter Set:** Application Wizardによって作成されるParagon Operator Interface(OI)画面ファイルを含みます。次の4つの画面を含んで作成されます：Main画面、Alarm画面、Journal画面、およびPIOデータ画面。

**Active X Starter Set:** includes the Visual Basic 4.0 files created by the Application Wizard. These files contain the Paragon ActiveX controls.

**Active X Starter Set:** Application Wizardによって作成されるVisual Basic 4.0ファイルを含みます。これらのファイルはParagon ActiveXコントロールを含んでいます。



## Operator Interface Display Setup Operator Interface画面のセットアップ

The points to be added to the OI are shown in Step 9. Here you can modify the display colors.  
OIに加えられるポイントはステップ9で示されます。ここで、画面の色を変更することができます。

### To modify your display colors:

画面の色を変更するには：

1. Leave the window at its default values, or select the desired colors.  
ウィンドウのデフォルト値を残すか、あるいは希望の色を選択してください。

2. Click on **Next**.

**Next** をクリックしてください。

TNT Application Wizard - Step 9 of 9

Select a Point for OI Display

Foreground color: Black

Background color: Cyan

Full names

Points for OI Display

PI0.IIISLAPPA.UC100.WriteOut

Add >> Add all >> << Delete << Del all

This is the ninth step of the Application Wizard. On this step you will add tags to an OI display, selecting from the P10 tags generated previously. A maximum of 10 tags are allowed to be included in the OI data display. Any tags that are in the list on the right will be included in the OI data display. The first 10 tags are added to this list

Exit < Back Next > Finish

## Application Summary

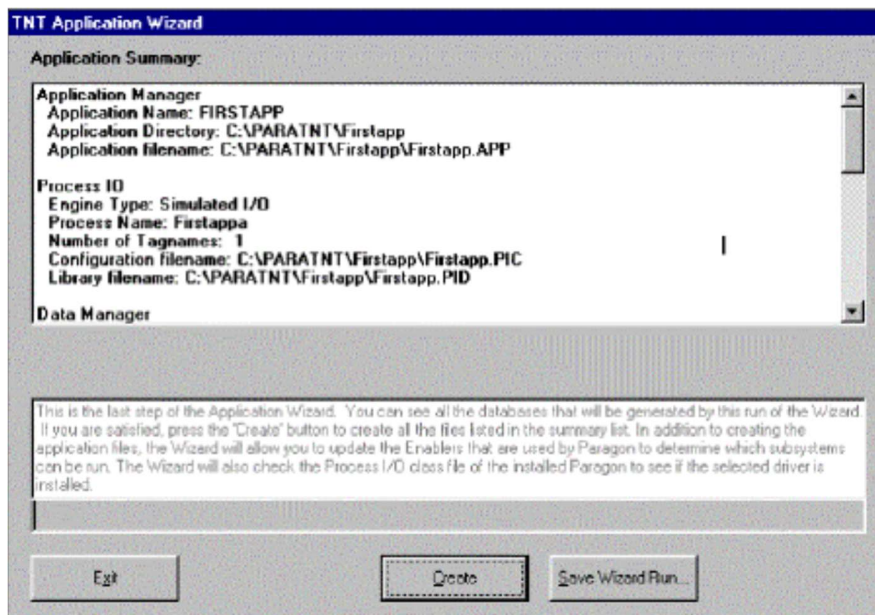
### アプリケーション・サマリー

Step 10 shows a list of all the files that will be created. Each included subsystem lists the details of its associated files. ステップ10では、作成されるすべてのファイルのリストが示されます。それぞれ含まれるサブシステムは、その関連するファイルの詳細をリストします。

#### The create the Application Summary:

アプリケーション・サマリーを作成するには：

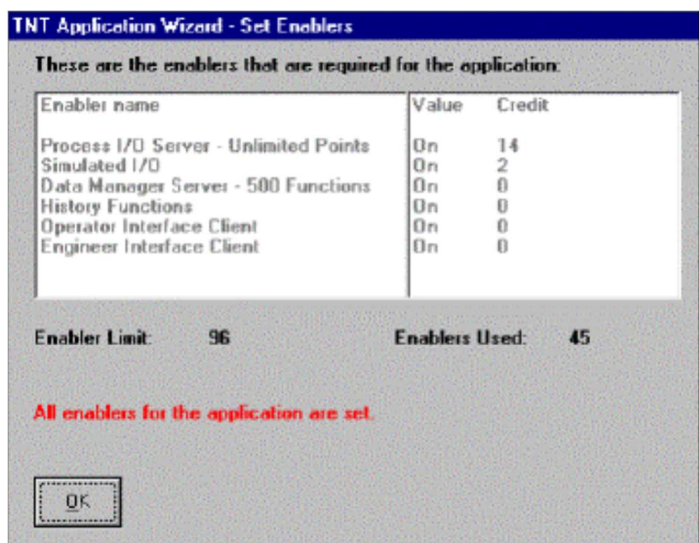
1. Click on the **Create** button. The files for the required components are created. Additionally, the Process I/O class file is checked. If the driver is not found you are prompted to install another driver. Otherwise the application will not run. **Create** ボタンをクリックしてください。要求されたコンポーネント用のファイルが作成されます。さらに、Process I/Oクラス・ファイルは検査されます。ドライバが見つからない場合、他のドライバを設定するよう求められます。そうでなければ、そのアプリケーションは実行できません。



**Enablers:** allow you to use Paragon's various features. Each feature requires a certain number of Enablers. Various levels of enabler packs are available from NemaSoft. These make use of the encrypted protection key.

**Enablers:** パラゴンの様々な機能を使用することを可能にしてください。各機能は有効化のための番号を要求します。様々なレベルの有効化パックがNemaSoftから利用可能です。これらは暗号化されたプロテクトキーを必要とします。

2. The TNT Application Wizard-Set **Enablers** window appears. It should read Enabler Limit: 96 and Enablers Used: 45. The TNT Application Wizard-Set **Enablers** ウィンドウがあらわれます。Enabler Limit: 96 と Enablers Used: 45 となっているのが読み取れる筈です。



3. Click on **OK**, then on Done. The Application Wizard installs an icon called Firstapp in your Paragon TNT folder. To run this application later, select it from the Start menu.

**OK**をクリックして完了してください。Application Wizardは、Paragon TNTフォルダーにFirstappと呼ばれるアイコンをインストールします。今後このアプリケーションを実行するには、スタート・メニューからそれを選んでください。

4. Click on the **Run Application** checkbox, then on **Yes**.

**Run Application**チェックボックスをクリックしてから、**Yes**をクリックしてください。



5. Login again and this application will appear.

再びログインすると、このアプリケーションがあらわれます。

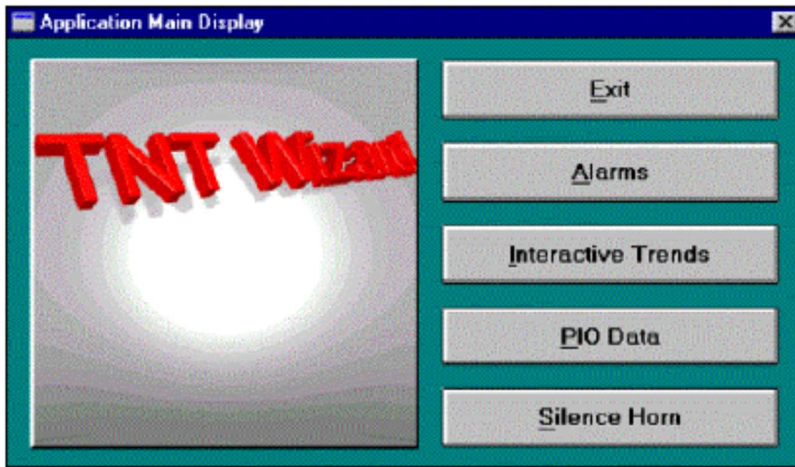
**Save Wizard Run:** saves the application file you just created to another directory. LAST.WIZ is the file that is created each time you make a new application with the Wizard. By default, this saves to the same location for each new application. If you do not change the location where the file is saved, you will overwrite existing files.

**Save Wizard Run:** 作成したアプリケーション・ファイルを別のディレクトリに保存します。LAST.WIZはウィザードで新しいアプリケーションごとに作成されるファイルです。これはデフォルトで新しいアプリケーションで同じ場所に保存されます。ファイルが保存される位置を変更しなければ、既存のファイルが上書きされてしまうでしょう。

**What's Next:** see what else you can do with your Paragon Application.

**What's Next:** あなたのParagon Applicationと共にできる他の何かを見る。

The application that the Wizard has created for you starts off with the following display:  
ウィザードがあなたのために作成したアプリケーションは次の画面から始まります：



By clicking on the right buttons you can:

右ボタンのクリックで、次のことができます：

Exit this application.

アプリケーションの終了

Load other displays.

他の画面のロード

Silence the alarm horn.

アラーム音を無くす

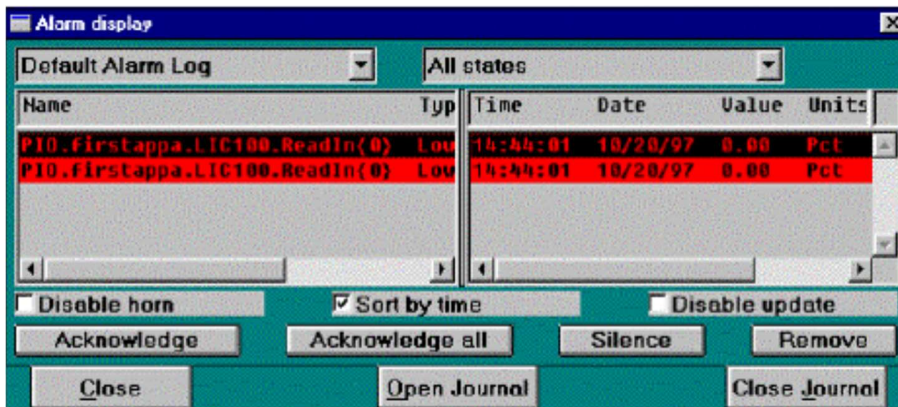
If you selected the Simulated I/O as part of your application's configuration, no information will be collected or displayed. However, if this application was connected to your factory process driver, data would be collected and displayed.

もしアプリケーション構成の一部としてSimulated I/Oを選択していれば、情報の収集や表示は行われません。しかし、もしこのアプリケーションが工場プロセス・ドライバーに接続されれば、データは収集されて表示されるでしょう。

The following display is an Alarm log that logs alarms in real-time (as they occur). If you selected the Simulated I/O driver you will notice that the display has logged two alarms, even though it is not collecting information. This is because Paragon's default low alarm value is 0. Since there is no data being collected, a 0 is sent to this process.

次の画面は、リアルタイム(それらが生じるとともに)にアラームを記録するアラーム・ログです。もしSimulated I/O ドライバを選べば、たとえそれが情報収集しなくても、画面が2つのアラームを記録したことに気づくでしょう。これはParagonのデフォルトの下限アラーム値が0だからです。データが集められていないので、0がこのプロセスに送られます。





**ODBC:** Open Database Connectivity (ODBC): A standard method of sharing data between databases and other programs. ODBC drivers use the standard Structured Query Language (SQL) to gain access to data from sources outside Paragon.

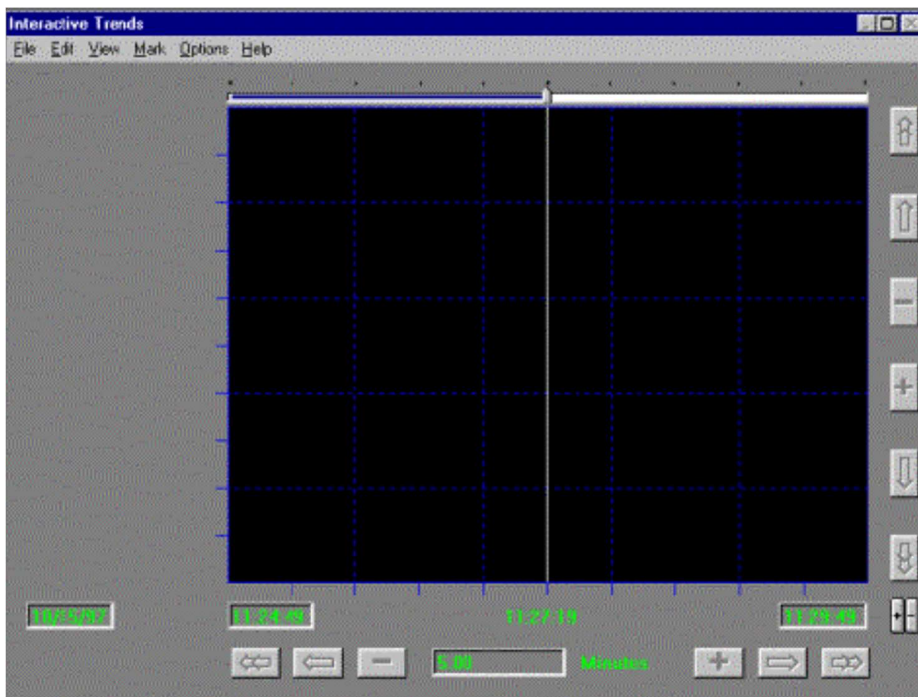
**ODBC:** Open Database Connectivity (ODBC): データベースと他のプログラム間のデータを共有する標準方法。 ODBCドライバーは、Paragonの外のソースからデータへのアクセスを得るのに、標準Structured Query Language (SQL)を使用します。

The Interactive Trend (Itrend) allows you to choose DM history or trend data to plot, select zoom, mark, etc. Itrends also allow you to plot historical data from an **ODBC** compliant database as long as it supports adequate date and time stamps. Itrend configurations can be saved and restored.

Interactive Trend(Itrend)はDM履歴や傾向データのプロット、ズーム、マーク、その他を選択させます。また、適正な日時スタンプをサポートすれば、Itrendsは**ODBC**対応するデータベースからの履歴データをプロットします。 Itrendの設定は、保存および復元することができます。

If connected to your factory process driver, you would see fluctuations and consistencies relative to the collected data, displayed as a trend.

もし工場プロセス・ドライバーに接続されれば、傾向の表示に、収集データに関する、変動および一貫性を見るでしょう。



The PIO Data Display shows the data collected from the **LIC100** function. Since no values are being collected, a 0 is displayed.

PIO Data Displayは、**LIC100**関数からの収集データを示します。値が集められていないので、0が表示されます。



## Chapter Summary

### 章のまとめ

Congratulations! You now know how to start Paragon, and use the Application Wizard to create a basic application. In the next chapter you will build on what you have learned, using the Process I/O, Operator Interface, and Application Manager to create a basic display.

おめでとうございます! あなたは、今、基本的なアプリケーションを作成するためにどのようにParagonを始めて、Application Wizardを使用するかを知っています。次の章で、あなたは自分の学んだProcess I/O, Operator Interface, およびApplication Managerを使用して基本的な画面を構築するでしょう。

# Process I/O Values

## In This Chapter:

この章に含まれるのは :

ä **Process I/O Subsystem 47**

ä **Operator Interface 54**

ä **Setting Object Attributes 62**

ä **Application Manager 68**

ä **Engineering Interface 72**

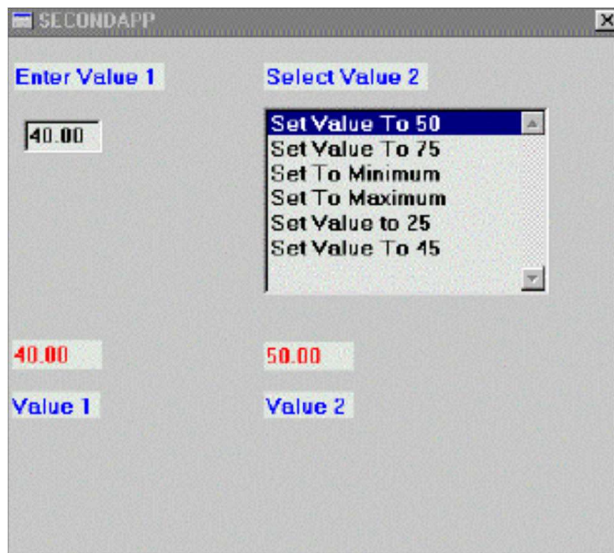
ä **Chapter Summary 74**

章のまとめ

In this exercise you are going to build two analog functions in the Process I/O subsystem (PIO) that access the engine (Simulated I/O or your actual driver) you selected from the Application Wizard.

**(OS/2 USERS** should read through the following sections, then complete the instructions in the shaded box on the fourth page of this section.) The display will include two data entry fields; one with a Data entry faceplate and the other with a list box. These are the input fields for the PIO functions. Lastly, you will create two data display fields that read the data. When you are done your display will look like the following:

この課題では、Application Wizardで選択したエンジン(Simulated I/Oあるいは実際のドライバー)にアクセスするProcess I/Oサブシステム(PIO)内の2つのアナログ関数を構築します。(OS/2ユーザーは次のセクションを通読し、さらに、このセクションの第4ページで灰色の箱に記述されている指示を行うべきです)。ディスプレイは2つのデータ登録項目を含みます；データ登録フェースプレートとリストボックスです。これらはPIO関数のための入力項目です。つまり、データを読む2つのデータ表示項目を作ります。完成すると、画面は次のように見えるでしょう：



In the completed display, the value you enter into the Enter Value 1 field will be displayed in the Value 1 display field. Similarly, the value you select from the Select Value 2 list box will be displayed in the Value 2 display field.

完成した画面では、Enter Value 1欄に入力した値はValue 1表示欄に表示されます。同様にSelect Value 2リストボックスから選んだ値はValue 2表示欄に表示されます。

## Process I/O Subsystem

The Process I/O subsystem interfaces Paragon TNT to external measurement and control devices. This allows other TNT subsystems to access data from these devices. The Process I/O performs collection and transformation of data from various hardware devices including:

Process I/Oサブシステムは、Paragon TNTを外部の測定と制御の装置に接続します。これにより、他のTNTサブシステムはこれらの装置からデータにアクセスすることが可能になります。Process I/Oは、次のものを含む様々なハードウェア装置からのデータの収集および変換を実行します：

Programmable Logic Controllers (PLC's).

Single Loop Controllers (SLC's).

Measurement hardware.

Internal plug-in PC-bus cards.

**Functions:** consist of a single Paragon operation for device communications, computations, control algorithms, data collection, and alarm monitoring.

**Functions:** 装置の通信、演算処理、制御アルゴリズム、データ収集およびアラーム監視のうち1つのParagon Operationで成ります。

Data transformation takes the form of scaling, conditioning, filtering, or raising to a power. PIO also performs automatic groupings of I/O points according to the communications protocol of the connected hardware, maximizing communications. The first step in this exercise is to define the Process I/O functions.

データ変換はスケールリング（拡大や縮小）、コンディショニング（状態調整）、フィルタリング（情報の選別）、あるいはパワーの上昇の形式を取ります。さらにPIOは、コミュニケーションを最大限にして、接続しているハードウェアの通信プロトコルによるI/Oポイントの自動グルーピングを行いません。この練習の第一歩はProcess I/O関数を定義することです。

## Editing The Wizard Library File

### ウィザード・ライブラリ・ファイルの編集

The Process I/O is configured in a two-stage process. The first is to build the Library File (\*.PID), which contains all the PIO processes and functions. The second is to build the Configuration File (\*.PIC), which contains all the PIO system-wide settings. In this section you will edit the Application Wizard library file. The library file does the following:

Process I/Oは2段階のプロセスで構成されます。1番目はLibrary File(\*.PID)の構築です。それはPIOプロセスおよび関数をすべて含みます。2番目はConfiguration File(\*.PIC)の構築で、それはPIOシステム全体に渡る設定をすべて含みます。このセクションでは、Application Wizard library fileを編集します。ライブラリ・ファイルは次のことを行います：

Defines the I/O functions.

I/O関数の定義

Assigns the I/O engine to the defined processes in the configuration file.

configuration file内の定義されたプロセスにI/Oエンジンを割り当てます。

The I/O functions relate to individual data points or sets of data points. They are tailored to each hardware device they are attached to. Functions may be seen as the equivalent of the connected hardware blocks, such as an Opto 22 Optomux B2 board.

I/O関数は、個々のデータ・ポイントあるいはデータ・ポイントのセットに関係します。それらは、取り付けられるハードウェア装置ごとに適合します。関数は Opto 22 Optomux B2ボードのような接続しているハードウェア・ブロックの同等物と見なされているかもしれません。

### To edit the Wizard Library File:

ウィザード・ライブラリ・ファイルを編集するには：

1. If Paragon is still running, exit by selecting **File** and **Exit** from the Application Manager.  
Paragonが動作している場合は、Application Managerから**File**と**Exit**を選び終了してください。

2. Select the **Firstappa** icon: from the **Paragon TNT** directory.

Paragon TNTディレクトリから**Firstappa**アイコンを選択します。

3. Login by selecting **Security** and **Login** and typing Guest.

**Security**と**Login**を選択し、Guestとタイプしてログインしてください。

**Important Note:** To complete the exercises in this guide, you must login from the Firstappa icon each time. This ensures that your files are in the current working directory. If they are in different directories, this application will not work.

**重要な注意:** このガイドにおける課題を完了するには、毎回Firstappaアイコンからログインしなければなりません。これは、あなたのファイルが現在の作業ディレクトリにあることを保証します。それらが異なるディレクトリにある場合、このアプリケーションは働きません。

4. From the Application Manager select **File** and **Open**. Click on **Firstappa.APP**, then on **OK**.

Application Managerから**File**と**Open**を選択します。**Firstappa.APP**をクリックして、**OK**してください。

5. From the Application Manager select **Builders** and **Process I/O**.

Application Managerから**Builders**と**Process I/O**を選択します。

6. Click on the process **Firstappa** to highlight it. In the function field click on **LIC100**. Its parameters are displayed in the right-hand column.

プロセス**Firstappa**をクリックしてハイライトしてください。関数項目で**LIC100**をクリックしてください。右欄にパラメータが表示されます。

**OS/2 USERS:** Follow these steps in order to complete the exercises in this manual:

1. From the Application Manager select **Builders** and **Process I/O**.
2. From the PIO Main menu select **File** and **New**.
3. Double-click on **Empty** under the Process field, then type Firstappa. Click on **OK**.
4. Select the PIO driver you need, or the Simulated I/O.
5. Double-click on **Empty** under the Function field, then type LIC100. Click on **OK**.
6. Start with step 8 below and follow the rest of these steps.

7. The fields listed in bold are the entries that need to be changed from their default values. In the Parameter Value field change the following values by:

太字でリストされた項目は、それらのデフォルト値から変更が必要な登録です。Parameter Value項目で次の値を変更してください:

0 Clicking on the specified field to select it.

指定項目をクリックして選択します。


0 Editing it in the parameter value field.

parameter value項目を編集します。

0 Pressing Enter when you are done.

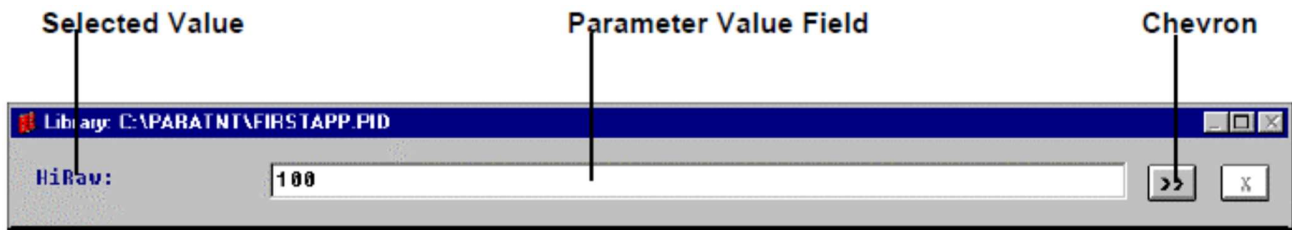
完了したなら、Enterを押します。

The **Chevron**: is used to select parameters for a particular field by clicking on it, and then choosing the appropriate value from the proceeding screen.

**Chevron**: は、進行スクリーンからそれをクリックして、次に、適切な値を選ぶことで特定項目のパラメータを選択するのに使用されます。







UnitAddr	:1	LoLoAlrmVal	:1.0000
StartReg	:0	HiHiAlrmVal	:99.0000
NumPoints	:1	AlrmDeadBand	:0.0000
InitMode	:Read First	AlrmInhibit	:Y
ReadIn	:0.0000	AlrmPriority	:0
WriteOut	:0.0000	HiLoDestCfg	:0
EngUnit	:	HLLDestCfg	:0
Description	:	CondTblName	:
LoRange	:0.0000	FilterTime	:0
HiRange	:100.0000	XFormType	:Disable
LoRaw	:0	<b>ScanPeriod</b>	<b>:0.25</b>
<b>HiRaw</b>	<b>:100</b>	ScanAlways	:N
LoAlrmVal	:5.0000	CommEnb	:Y
HiAlrmVal	:95.0000	CommEnbName	:

**HiRaw:** The maximum value sent to the hardware when the PIO specifies "HiRange" engineering units to be sent.  
**HiRaw:** PIOが送られる"HiRange"エンジニアリング・ユニットを指定するとき、最大値はハードウェアに送りました。

**ScanPeroid:** The period between updates of data points in the hardware, expressed in seconds. The minimum scan period is 0.10.

**ScanPeriod:** ハードウェアのデータ・ポイントのアップデート間隔（単位は秒）。最小スキャン間隔は0.10です。

For definitions of the other parameters see the Reference Manual or the User's Guide.

他のパラメータの定義については、リファレンス・マニュアルあるいはユーザガイドを見てください。

## Creating A Second Function For The Library File

### Library Fileの2番目の関数の作成

In this section you are going to add a second function to the process Firstappa.

このセクションでは、プロセスFirstappaに2番目の関数を加えます。

#### To add a second function to the Firstappa process:

**Firstappa**プロセスに2番目の関数を追加するには：

1. Highlight the **Firstappa** Process in the **Process** list box.

**Process** リストボックスの**Firstappa**プロセスをハイライトしてください。

2. Click on **Description** and type Reads/Writes Analog Functions into the **Parameter Value** field. Press **Enter** and the description will appear in the process parameters list box. Click on **OK**

**Description**をクリックして、**Parameter Value**項目にReads/Writes Analog Functionsとタイプしてください。**Enter**を押すと、記述がprocess parametersリストボックスにあらわれます。**OK**をクリックしてください。

**Selected Value Parameter Value Field Chevron**



## 50 i Paragon Getting Started

### Defining Function Parameters

#### 関数パラメータの定義

The next step is to name the functions and assign their parameters.

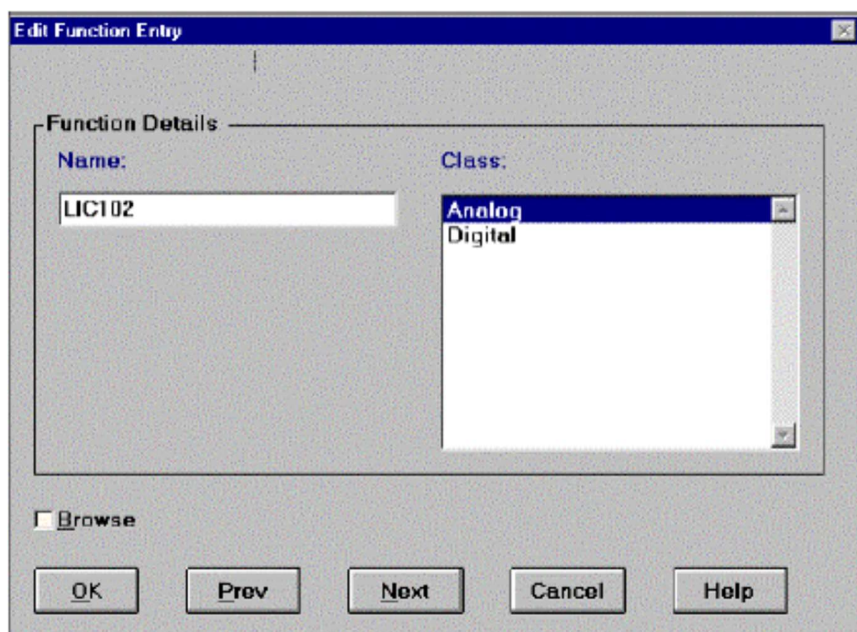
次のステップは、関数を命名して、そのパラメータを割り当てます。

1. Highlight the **Firstappa** process, then double-click on the **\*\*\*EMPTY\*\*\*** field under Function Name to bring up the **Edit Function Entry** dialog box.

**Firstappa**プロセスをハイライトにして、Function Nameの下の**\*\*\*EMPTY\*\*\***項目をダブルクリックして、**Edit Function Entry**ダイアログボックスを表示してください。

2. In the **Name** field type LIC102, then click on **Analog** in the **Class** field. Click on **OK**.

**Name**項目に**LIC102**をタイプして、**Class**項目で**Analog**をクリックしてください。**OK**をクリックしてください。



3. The fields listed in bold are the entries that need to be changed from their default values. In the Parameters Value field change the following values by:

太字でリストされた項目は、それらのデフォルト値から変更が必要な登録です。Parameter Value項目で次の値を変更してください：

0 Clicking on the specified field to select it.

指定項目をクリックして選択します。

0 Editing it in the Parameter Value field.

parameter value項目を編集します。

0 Pressing Enter when done.

完了したなら、Enterを押します。

UnitAddr	:1	LoLoAlrmVal	:1.0000
StartReg	:1	HiHiAlrmVal	:99.0000
NumPoints	:1	AlrmDeadBand	:0.0000
InitMode	:Read First	AlrmInhibit	:Y
ReadIn	:0.0000	AlrmPriority	:0
WriteOut	:0.0000	HiLoDestCfg	:0
EngUnit	:	HLLDestCfg	:0
Description	:	CondTblName	:
LoRange	:0.0000	FilterTime	:0
HiRange	:100.0000	XFormType	:Disable
LoRaw	:0	ScanPeriod	:0.25
HiRaw	:100	ScanAlways	:N
LoAlrmVal	:5.0000	CommEnb	:Y
HiAlrmVal	:95.0000	CommEnbName	:

**StartReg:** The number for the first set of referenced data points.

**StartReg:** 参照されたデータポイントの最初のセットのための数。

## Saving The Library File

### ライブラリ・ファイルの保存

You have completed defining your functions. The final step is to name and save the library file.

関数の定義を完成しました。最終ステップは、ライブラリ・ファイルに名前を付けて保存することです。

#### To save the Library file

#### Library fileを保存するには

1. Select **File**►**Save File**►**Library**. Type *Firstappa.PID* into the **File Name** field, then press **OK**.

**File**►**Save File**►**Library**を選択してください。 **File Name**項目に*Firstappa.PID*をタイプして、**OK**を押してください。

## Creating The Configuration File

### Configuration Fileの作成

**Runtime:** the process of running your Paragon application.

**Runtime:** Paragonアプリケーションを実行するプロセス。

**Station Name:** identifies the I/O node that the process will run on.

**Station Name:** プロセスを実行するI/Oノードの識別。

The Application Wizard has done this step for you. (**OS/2 USERS** should do all of the following steps except for step 3). To show you how to make the Configuration file, you are going to create it again. The configuration file is used as the start-up file for the Process I/O. It defines which processes execute during **Runtime** and their global Runtime parameters. These parameters include:

Application Wizardはあなたのためにこのステップを行いました。(OS/2ユーザーはステップ3以外の次のステップのすべてをすべきです)。どのようにConfiguration fileを作るかをあなたに示すために、あなたは再びそれを作成するでしょう。 configuration fileはProcess I/Oの開始ファイルとして使用されます。それは、どのプロセスがランタイムとそのグローバル・ランタイム・パラメータを実行するか定義します。これらのパラメータは次のものを含みます:

The process name.

プロセス名

The **station name** that the process will run on.

プロセスが実行されるステーション名(**station name**)

The communication port.  
通信ポート

The scan periods.  
スキャン間隔

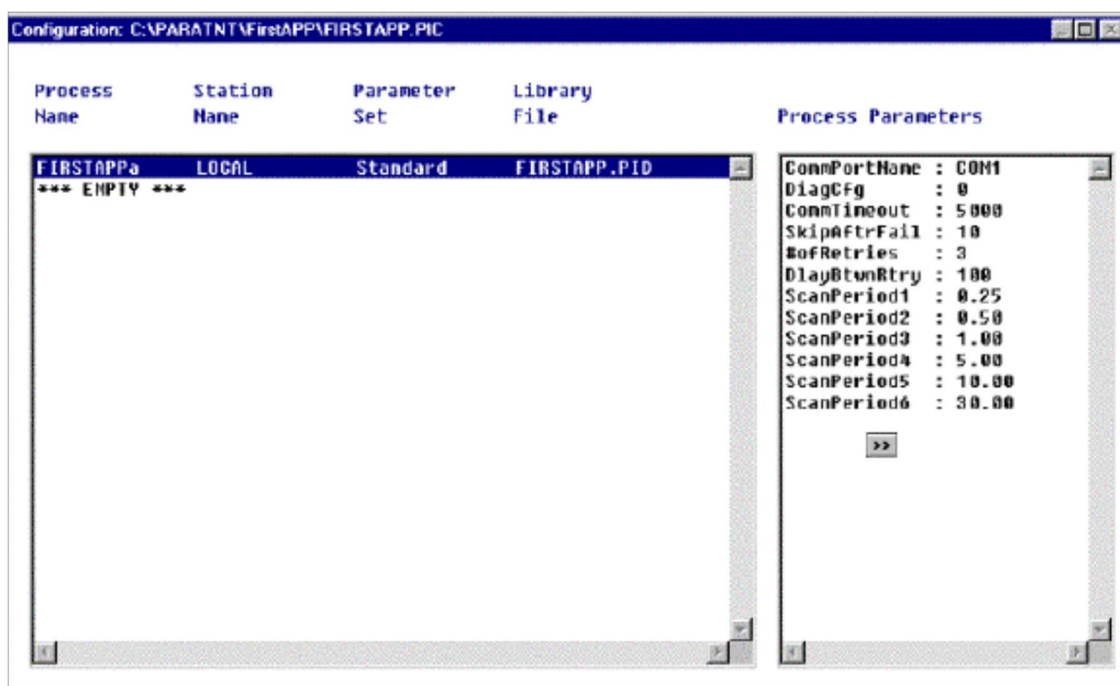
It also references the library file.  
さらに、それはライブラリ・ファイルに参照を付けます。

### To create the configuration file: configuration fileを作成するには :

1. From the PIO builder Main menu select **Window&Configuration**. If using OS/2, double-click on **\*\*\*EMPTY\*\*\***.  
PIO builderのメイン・メニューから**Window&Configuration**を選択します。OS/2を使用する場合は、**\*\*\*EMPTY\*\*\***の上でダブルクリックしてください。

2. In the **Process Name** field, click on **Firstappa** to select it. Then select **Edit&Cut** to delete this process. Now that you have deleted the old file, you can recreate it.

**Process Name**項目で、**Firstappa**をクリックして選択してください。そして、**Edit&Cut**でこのプロセスの削除を選んでください。古いファイルを削除したので、それを再作成できます。



3. Double-Click on **\*\*\*EMPTY\*\*\***, then use the **Chevron:** to select **Firstappa** from the **Process Name** field.  
**\*\*\*EMPTY\*\*\***の上でダブルクリックしてから、**Process Name**項目から**Firstappa**を選ぶために**Chevron: >>**を使用してください。

4. Leave the default **Station Name** as **Local** .  
**Local**というデフォルトを**Station Name**に残してください。

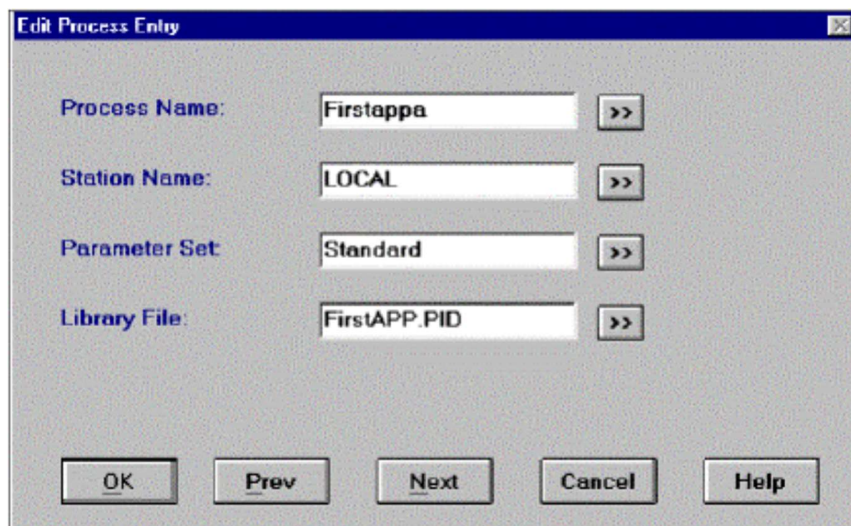
5. Leave the default **Parameter Set Name** as **Standard**. This field specifies:  
**Standard**としてデフォルトを**Parameter Set Name**に残してください。この項目は指定します :

0 The name of a custom I/O configuration for each process.

各プロセスのカスタムI/O構成の名前。

0 The global runtime parameters, such as communication port names, scan periods, etc.  
通信ポート名, スキャン間隔, などのグローバル・ランタイム・パラメータ

6. Click on the **Chevron:** next to the **Library File** field. Select **Firstappa.PID**. Click on **OK**, then **OK** again.  
Library File項目の隣の**Chevron: >>**をクリックしてください。 **Firstappa.PID**を選択してください。 **OK**をクリックして、さらに**OK**をクリックしてください。



**Note:** Click on the Firstappa process to highlight it. Notice that the parameter set under Standard is listed in the Process Parameters list box. You can change any of these parameters by double-clicking on the entry and then entering a new value into the dialog box.

注意 : Firstappaプロセスをクリックしてハイライトしてください。 Standardの下に設定されたパラメータがProcess Parametersリストボックスにリストアップされているのに注目してください。登録でダブルクリックして、続けてダイアログボックスで新しい値を入れることにより、ほとんどのパラメータを変更することができます。

## Saving The Configuration File Configuration Fileの保存

To save the Configuration File  
Configuration Fileを保存するには

**Control Panel:** is used to execute subsystem start-up files for Runtime. It is also used to shut down executing subsystems, view subsystem diagnostics, and start and stop Control Panel Extensions.

**Control Panel:** ランタイムのスタートアップ・ファイルはサブシステムを実行するために使用されます。さらにそれは、実行しているサブシステムのシャットダウン、サブシステム診断のビュー、そしてコントロール・パネル拡張の開始と停止に使用されます。

1. Select **File>Save File>Configuration**.

**File>Save File>Configuration**を選択します。

2. Type Firstappa.PIC into the **File Name** field then click on **OK**. This is the Control Panel start-up file for the Process I/O. It tells Runtime which processes to load and where to access their parameters. You have created a process called Firstappa which contains:

**File Name**項目にFirstappa.PICをタイプして**OK**をクリックしてください。これはProcess I/Oのコントロール・パネルのスタートアップ・ファイルです。それは、どのプロセスをロードするべきであるか、また、どこでそれらのパラメータにアクセスするべきであるかランタイムに伝えます。次のものを含むFirstappaと呼ばれるプロセスを作成しました:

Two analog functions: LIC100 and LIC102.  
2つのアナログ関数 : LIC100 と LIC102.

The library file, Firstappa.PID, which stores the information for the process and its functions (think of it as the database).

プロセスおよびその関数（それをデータベースと見なす）に備えて情報を蓄えるライブラリ・ファイル (Firstappa.PID)。

The configuration file, Firstappa.PIC, which references the library file and tells Runtime what processes to execute.

ライブラリ・ファイルを参照して、どのプロセスを実行するべきかランタイムに伝える configuration file(Firstappa.PIC)。

These files will be located in your working directory.

これらのファイルはあなたの作業ディレクトリに存在します。

## Operator Interface

**Analog:** any value with magnitude, such as voltage, temperature, or numeric.

**Analog:** 電圧、温度あるいは数値のような量を備えた任意の値。

**Discrete:** is also referred to as a digital value. It is a binary value always equal to 1 or 0. Some examples are high/low, on/off, true/false, yes/no.

**Discrete:** あるいはデジタル値と呼ばれます。それは1または0の2進値です。いくつかの例はhigh/low, on/off, true/false, yes/noです。

**String:** is alphanumeric information represented by ASCII characters such as text messages, names, barcodes, or time and date stamps.

**String:** テキストメッセージ、名前、バーコード、あるいは時間および日付のスタンプなど、ASCIIキャラクタによって表される英数字の情報です。

The next step is to design your display in the Operator Interface builder. The Operator Interface provides graphical representations of real-time and historical data from other Paragon subsystems.

次のステップではOperator Interface builderで画面を設計します。オペレーター・インターフェースは、他のParagonサブシステムからのリアルタイムおよび履歴のデータのグラフ表示を提供します。

The Operator Interface, contains the following display elements:

オペレーター・インターフェースは、次の表示要素を含みます：

Animated process graphics.

動作を示すプロセス・グラフィック

Data display of process values ( **analog**, **discrete**, and **string**).

プロセス値(**analog**, **discrete**, および **string**)のデータ表示

Data entry of analog, discrete, and string values.

analog, discreteおよびstringの値のデータ登録

Real-time and historical trends.

リアルタイムと履歴によるトレンド（傾向）

Alarm and event journals.

アラームとイベントのジャーナル

Graphical user interface (GUI) control objects, such as list boxes, radio buttons, push buttons, and sliders.

リストボックス、ラジオボタン、プッシュボタン、およびスライダーのようなグラフィカル・ユーザー・インターフェース(GUI)

Bitmaps.

ビットマップ

Third party and multimedia applications.

サードパーティおよびマルチメディアのアプリケーション

## Entering The Operator Interface

**To enter the Operator Interface builder:**

**Operator Interface builder**を始めるには:

1. From the Application Manager Main menu, select **Builders**▶**Operator Interface**.

Application Managerのメイン・メニューから、**Builders**▶**Operator Interface**を選択してください。



The OI Builder contains the following features:

OI Builderは次の機能を含みます：

The Main menu, which contains the commands used to create your displays.

メイン・メニュー、それに含まれるコマンドは画面を作成するため既に使ったことがあります。

The Status Line, which contains current information about the status of the window, including:

ステータス・ライン、ウィンドウとそれに含まれているものの状態に関する現在の情報を持ちます。

0 The current mode of the application.

アプリケーションの現在のモード

0 The X and Y coordinates of the current cursor location or current selected object.

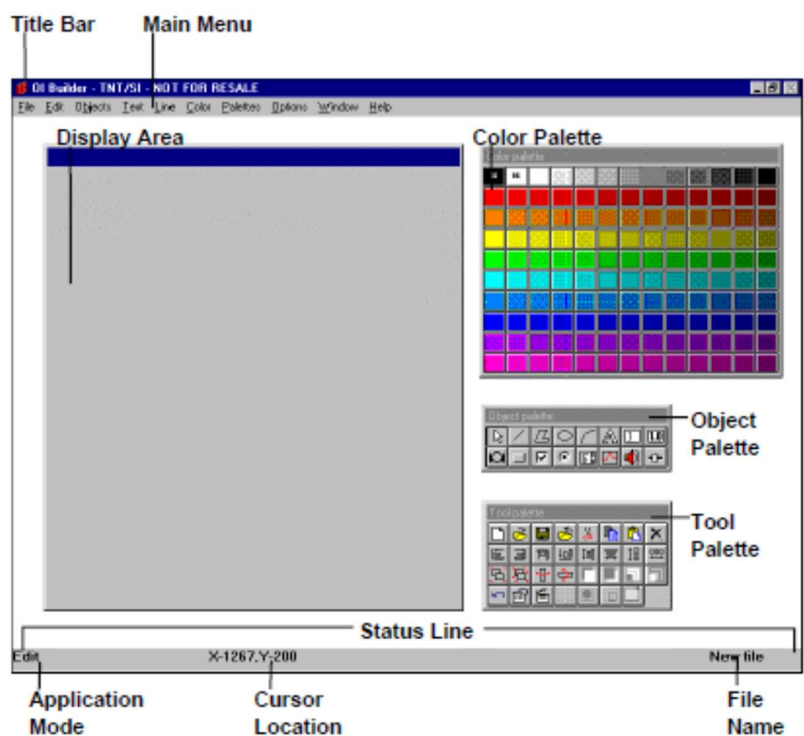
現在のカーソル位置あるいは現在の選択オブジェクトのXY座標

0 The paint style.

ペイント・スタイル

0 The name of the active display file.

アクティブなディスプレイ・ファイルの名前



## Object and Color Palettes オブジェクトとカラーのパレット

In the primary window there are two floating palettes. One for placing objects and the other for selecting colors. You can move, resize or close them.

主なウィンドウには、2つの浮遊パレットがあります。1つはオブジェクト配置用、そしてもう1つは色の選択用です。それを移動、サイズ変更、クローズすることができます。

The Object Palette contains objects that are used to configure your display. Objects are selected from the palette by clicking on the appropriate icon and then clicking on the display window where you want to place the object. The Basic Object palette contains the following:

オブジェクト・パレットは、画面を構成するのに使うオブジェクトを持ちます。オブジェクトは、適切なアイコンをクリックすることによりパレットから選択して、それからオブジェクトを置きたいところで、表示ウィンドウをクリックします。基本のオブジェクト・パレットは次のものを含みます。

**Edit tool:** Select, move, or resize objects.



**Edit tool:** オブジェクトの選択、移動、サイズ変更

**Polyline tool:** Draw line segments.



**Polyline tool:** 直線の描画

**Polygon tool:** Draw multi-sided objects.



**Polygon tool:** 多角形の描画

**Ellipse tool:** Draw circles and ellipses.



**Ellipse tool:** 円と楕円の描画

**Arc tool:** Draw open or closed arcs.



**Arc tool:** 開いた弧あるいは閉じた弧の描画

**Text tool:** Create a text label



**Text tool:** テキスト・ラベルの作成

**Data Entry tool:** Add a data entry field.



**Data Entry tool:** データ登録欄の追加

**Data Display tool:** Add a data display field.



**Data Display tool:** データ表示欄の追加

**Bitmap tool:** Place pre-designed bitmap objects.



**Bitmap tool:** あらかじめデザインされたビットマップ・オブジェクトの配置

**Push Button tool:** Add a push button.



**Push Button tool:** プッシュ・ボタンの追加

**Check Box tool:** Add a check box.



**Check Box tool:** チェック・ボックスの追加

**Radio Button tool:** Add a radio button.



**Radio Button tool:**ラジオ・ボタンの追加

**List Box tool:** Add a list box to select values from.



**List Box tool:**値を選択するリストボックスの追加

**Plot tool:** Add a display that plots events in real-time.



**Plot tool:**リアルタイムにイベントをプロットするディスプレイの追加

**Event tool:** Create an event journal.



**Event tool:**イベント・ジャーナルの作成

**Slider tool:** Add a slider to manipulate values.



**Slider tool:**値を操作するスライダの追加

## Color Palette

### カラー・パレット

The Color Palette contains colors that you can use to change the foreground, background, or selected object.  
カラー・パレットは、前景、背景、あるいは選択したオブジェクトの変更に使用する色を持ちます。

You can customize both palettes to contain any amount or arrangement of objects and colors. For the exercises in this manual you will only need basic palettes.

オブジェクトとカラーの任意の量やアレンジを含むため両方のパレットをカスタマイズできます。このマニュアルの課題では、基本的なパレットだけを必要とします。

#### To select the basic palettes:

基本パレットを選択するには：

1. Right-click on the Color palette and select **Small.fcp**.

カラー・パレットのうえで右クリックして、**Small.fcp**を選択する。

2. Right-click on the Object palette and select **Basic.fop**.

オブジェクト・パレットのうえで右クリックして、**Basic.fop**を選択する。



## Tool Palette

### ツール・パレット

The Tool Palette allows you to perform shortcuts for Main menu commands. Commands are grouped into four categories.

ツール・パレットはメイン・メニューのコマンドへのショートカットを提供します。コマンドは4つのカテゴリに分類されます。

#### 1. Standard Tools

標準ツール

#### 2. Layout Tools

レイアウト・ツール

#### 3. Object Tools

オブジェクト・ツール

#### 4. Workspace Tools

ワークスペース・ツール

You can customize the palette by right-clicking on it to either show or hide each of the categories, either from the main menu or the context menus (see the Tool Palette section in the User's Manual for more details).

メイン・メニューあるいはコンテキスト・メニューから各カテゴリを右クリックして、表示(show)/非表示(hide)のパレットのカスタマイズができます(より詳細に関してはユーザーズマニュアルのツール・パレットのセクションを参照してください)。

## Creating A Display Window

### ディスプレイ・ウィンドウの作成

#### To create a display window:

ディスプレイ・ウィンドウを作成するには :


1. From the Main menu select **File&New** to enter the **Edit Window Parameters** dialog box. Configure the window by doing the following:

メイン・メニューから、**File▶New**を選択して**Edit Window Parameters**ダイアログボックスに入ってください。次の手順でウィンドウを構成してください。

0 Click in the **Title** field and type **Firstappa**.

**Title**項目をクリックして、**Firstappa**とタイプしてください。

0 Click on the **Chevron:** next to the **Background** field and select **light gray**.

**Background**(背景)項目の**Chevron:**  をクリックして、薄い灰色(**Light gray**)を選択してください。

0 Click on **OK**. This becomes your new Background color.

**OK**をクリックします。これで新しい背景色が設定されます。

2. Click on **OK** again to save the parameters and exit the dialog box.

パラメーターを保存して、ダイアログ・ボックスを終えるため、再び**OK**をクリックしてください。

**Note:** If at any time you need to edit these parameters, you can do so by selecting **Options&Window Parameters**, or right-click and use the context menu.

**注意 :** これらのパラメーターを編集する必要がある場合は、いつでも、**Options▶Window Parameters**を選択するか、マウスを右クリックしてコンテキストメニューを使用することができます。

3. To resize the window, move the mouse to the right side of the Firstappa display window until the cursor becomes a double arrow. Hold the left mouse button down, drag the mouse to the right about 1 1/2", then release.

ウィンドウをサイズ変更するには、Firstappaディスプレイ・ウィンドウの右端にマウスを動かして、カーソルが二重矢印になるまで置いてください。そしてマウスの左ボタンを押したまま、マウスを右の方へ1 1/2インチだけドラッグしてから、放してください。

## Placing Objects

### オブジェクトの設置


The next step is to place the Enter Value 1 data entry field on your display (refer to the display on the second page of this chapter as needed).

次のステップはEnter Value 1データ入力項目をディスプレイに設置することです（必要に応じて本章第2ページのディスプレイを参照してください）。

#### To place the object for this display:

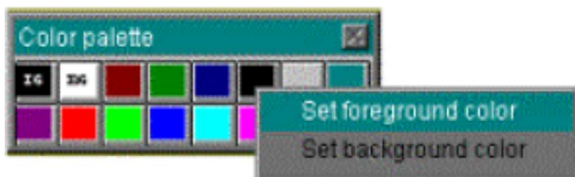
このディスプレイにオブジェクトを設置するには：

1. On the **Object Palette** click on the **Data Entry Tool**: Before you place the data entry field on the display, you need to change its color.

**Object Palette**で、**Data Entry Tool**:  をクリックしてください：データ入力項目をディスプレイに設置する前に、色を変更する必要があります。

2. On the **Color Palette** right-click on the light gray square and select **Set Foreground Color**.

**Color Palette**のライトグレイ（明るい灰色）の四角の上で右クリックして、**Set Foreground Color**を選択してください。



3. Move the cursor to the display window and click on the upper-left corner to paste the data entry field.

ディスプレイ・ウィンドウにカーソルを移動して、データ入力項目を貼るために左上の角をクリックしてください。

**Note:** When you paste objects on the screen, click where you want the lower-left corner of the object to be. Objects pop-up and to the right.

注意：スクリーンにオブジェクトを貼る場合、オブジェクトのより低い左の角をそうしてほしいところでクリックしてください。オブジェクト・ポップ・アップ、そして右に。

## Adding A List Box

### リストボックスの追加


The second data entry field will be a List Box. Using the List Box tool you can display a scrollable list of options to choose from. Move through the list using either the scroll bars, or arrows. Double-click on an option to select it.

2つ目のデータ入力項目はリストボックスになります。List Box toolを使用して、選択するオプションのスクロール可能なリストを表示することができます。スクロールバー、あるいは矢印を使用して、リストを動かしてください。オプションを選択するには、それをダブルクリックしてください。

#### To add a list box:

リストボックスを追加するには：

1. On the **Object Palette** click on the **List Box Tool**: . Move the cursor to the right of the Data Entry field, and then about 1 1/2" down. Click to paste it.

**Object Palette**から**List Box Tool:** をクリックしてください。データ入力項目の右にカーソルを動かして、1 1/2インチくらい下げてください。クリックして、それを貼りつけてください。

2. If needed, resize the list box so that it is about 1 1/2" by 1 1/2" by:

もし必要ならば、それが約1 1/2インチ×1 1/2インチになるように、次の手順でリストボックスのサイズを変更することができます。

0 Move the cursor to the **Object Palette** and click on the **Edit Tool:** .

**Object Palette**にカーソルを移動して、**Edit Tool:** をクリックしてください。

0 Move back to the display window and click on the list box to select it. A frame is drawn around the list box.

ディスプレイ・ウィンドウに戻り、リストボックスをクリックして選択してください。リストボックスのまわりにフレームが引かれます。

0 Place the cursor on the bottom-left corner of the frame until the cursor becomes a double arrow.

カーソルが二重矢になるまで、カーソルをフレームの左下の角に置いてください。

0 Hold the mouse button down and move it up and to the right until the list box is the desired size. Release the mouse button.

マウスボタンを押したまま、リストボックスが希望のサイズになるまで、上そして右に移動してください。そしてマウスボタンを放してください。

## Repositioning An Object オブジェクトの位置の変更


If you make a mistake placing your objects, you can reposition the object by:

オブジェクトの設置する位置を間違えた場合、次の手順でオブジェクトの位置を変更できます：

**To reposition an object:**

オブジェクトの位置を変更するには：

1. Click on the **Edit Tool:** and then clicking on the object (placeholders will show that it has been selected).

**Edit Tool:** をクリックして、次にオブジェクトをクリックします(プレースホルダーは、それが選択されたことを示します)。

2. Once selected, place the mouse over the center of the object. Hold the mouse button down, drag the object to the desired position and release.

選択されているオブジェクトの中心にマウスを置きます。マウスボタンを押したまま、希望の位置までオブジェクトをドラッグして、放してください。

## Data Display Fields

### データ・ディスプレイ項目

Next, you need to place the Value 1 and Value 2 data display fields to show the values, from the Enter Value 1 data entry field and the Select Value 2 list box.

次に、Value 1データ入力項目の入力およびValue 2リストボックスの選択からの値を表示するために、Value 1とValue 2のデータ・ディスプレイ項目の設置が必要です。

**To add a data display field:**

データ・ディスプレイ項目を追加するには：

1. Click on the **Data Display Tool:**

**Data Display Tool:**をクリックしてください。

2. Right-click on the red square, then select **Set Foreground Color** and release.

赤色の四角の上で右クリックして、**Set Foreground Color**を選んで放してください。

3. Move back to the display window and place the cursor about 2 1/2" beneath the first data entry field. Click to paste the data display.

ディスプレイ・ウィンドウに戻り、最初のデータ入力項目の約2 1/2インチ下にカーソルを置きます。クリックするとデータ・ディスプレイに貼られます。

4. Move the cursor to the right and paste another data display field next to it.

右にカーソルを移動させて、その隣に別のデータ表示項目を貼ってください。

## Labeling The Display

### ディスプレイもラベルを付ける

**To label the display:**

ディスプレイにラベルを付けるには：

1. Select the **Text Tool** :

**Text Tool:**  を選択します。

2. Right-click on dark blue and set it as the foreground color.

濃い青(dark blue)のうえで右クリックして、foreground colorとしてセットしてください。

3. Move to the display window and click directly above the first data entry field and type Enter Value 1.

ディスプレイ・ウィンドウに移動して、最初のデータ入力項目の真上をクリックして、Enter Value 1をタイプしてください。

4. Click above the list box on the right and type Select Value 2.

右にあるリストボックスの上をクリックして、Select Value 2をタイプしてください。

5. Click directly beneath the first data display field and type Value 1. Under the second data display field type Value 2.

1つ目のデータ表示項目の真下をクリックしてValue 1をタイプしてください。2つ目のデータ表示項目の下でValue 2をタイプしてください。

## Aligning Objects

### オブジェクトの整列


Using the Align option, you can perfectly align your objects in the display window.

Alignオプションを使用して、ディスプレイ・ウィンドウのオブジェクトをきれいに整列できます。

**To align your objects:**

オブジェクトを整列するには：

1. Select the **Edit Tool** :

**Edit Tool:**  を選択してください。

2. Move the mouse to the display window, left of the **Enter Value 1** text field.

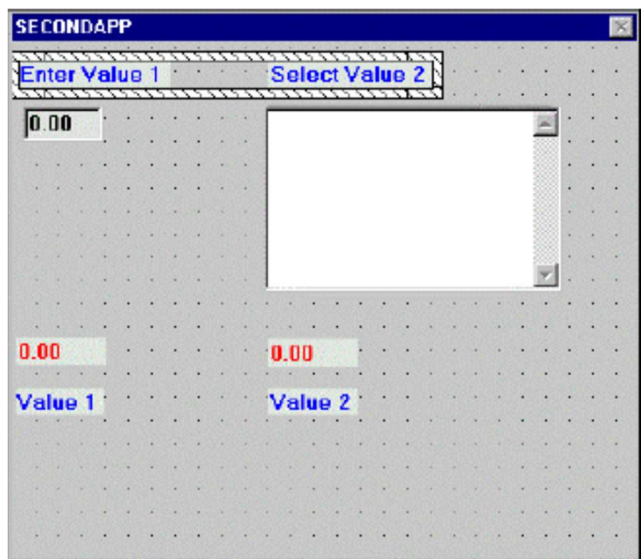
ディスプレイ・ウィンドウのEnter Value 1テキスト項目の左にマウスを移動させてください。

3. Select the top two text fields by holding down the mouse button and dragging it to surround both fields.

上の2つのテキスト項目を選択するために、マウスボタンを押したまま2つの項目を囲むようにドラッグしてください。

4. Release the mouse button, and a frame is drawn around the two fields.

マウスボタンを放すと、2つの項目を囲んで枠線が引かれます。

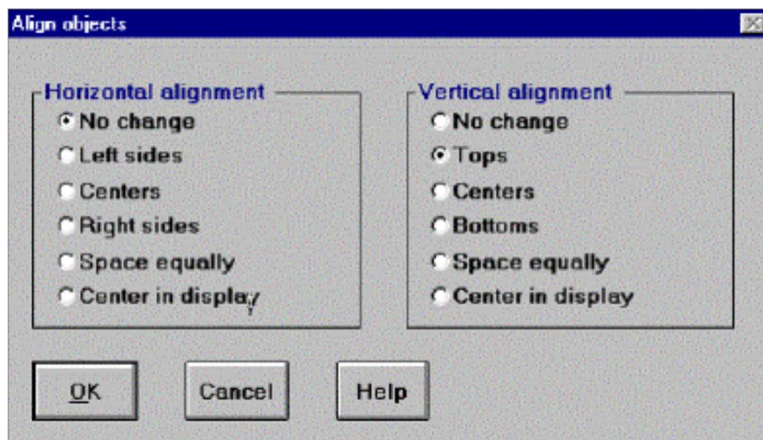


5. From the Main menu select **Options**→**Align**.

メイン・メニューから、**Options**→**Align**を選択してください。

6. Under **Vertical Alignment**, click on the **Tops** radio button, then on **OK**.

**Vertical Alignment**で、**Tops**をクリックして、**OK**してください。



Repeat the same procedure for the two data display fields, and their labels. Reselect the **Edit** tool, or right-click on the display window and select **Edit** mode. You can center each object's label by selecting the object and its label, choosing **Options**→**Align**, and then **Centers** under Horizontal Alignment.

2つのデータ表示項目とそのラベルのために同じ手順を繰り返してください。**Edit tool**を再選択するか、ディスプレイ・ウィンドウの上で右クリックして**Edit mode**を選択してください。各オブジェクトのラベルをセンタリング（中央寄せ）するには、オブジェクトとラベルを選択して、**Options**→**Align**を選び、そしてHorizontal Alignmentで**Centers**を選んでください。



# Setting Object Attributes

## オブジェクト属性の設定

**Attributes:** specified parameters that determine the actions of an object.

**Attributes:** オブジェクトのアクションを決定する、指定されたパラメーター

To link an object to a process you set **attributes** for it. Each object has its own set of attributes which are defined from its **Edit Object Attributes** dialog box. Here you can enter process parameters to connect objects to, and then define its properties.

You can specify attributes that alter the style of the object such as line type, color, or fill style.

オブジェクトをプロセスにリンクするために**attributes** (属性) を設定します。各オブジェクトには、それ自身の**Edit Object Attributes** ダイアログ・ボックスから定義される属性のセットがあります。ここで、オブジェクトを接続するプロセス・パラメータを入力して、次に、プロパティを定義します。

**line type** (ライン・タイプ), **color** (色), または**fill style** (注入スタイル) などオブジェクトのスタイルを変更する属性を指定できます。

In this section, you will connect objects to the Process I/O subsystem functions.

このセクションでは、Process I/Oサブシステム関数にオブジェクトを接続します。


## Editing The Data Entry Field

### データ入力項目の編集

**To edit the data entry field:**

データ入力項目を編集するには :

1. Select the **Edit Tool** :

**Edit Tool:**  を選択してください。


2. Move to the display window and double-click on the first data entry field. This brings up its **Edit Object Attributes** dialog box. Once the attributes dialog box is open, you can edit its parameters. When an object attribute is selected, the dialog box changes presenting fields specific to that choice. The Object Attributes list box lists all the available attributes for the selected object.

ディスプレイ・ウィンドウに移動して、最初のデータ入力項目をダブルクリックしてください。これはその**Edit Object Attributes** ダイアログ・ボックスを出します。**Attributes** ダイアログ・ボックスが開けば、そのパラメータを編集できます。オブジェクト属性が選択されると、ダイアログ・ボックスはその選択専用に表示項目を選びます。**Object Attributes** リストボックスは、選択されたオブジェクトのための利用可能な属性をすべてリストします。

3. Type 6 in the **Length** field and 2 in the **Decimal Places** field. This determines the length of the data entry field.

**Length**項目で6、**Decimal Places**項目で2をタイプします。これはデータ入力項目の長さを決めます。

4. To enter the **LIC100** function that you created earlier from the Process I/O subsystem click on the **Chevron:** next to the **Connection field**. Then do the following

以前にProcess I/Oサブシステムで作成した**LIC100**関数を入力するには、**Chevron:**  そして**Connection field**をクリックしてください。そして、次のことを行ってください。

0 From the **Topic** field select **Process I/O**. This specifies the subsystem that you are connecting to.

**Topic**項目から**Process I/O**を選択します。これは、接続しているサブシステムを指定します。

**Note:** Whenever you do not see a function that you are trying to connect to, click on Browse to find it.

注意 : 接続しようとしている関数が見つからない時は、**Browse to find it**をクリックしてください。

0 In the **Data Source** field click on **Firstappa.PID**. This is the library file which contains all the PIO processes and functions.

**Data Source**項目で**Firstappa.PID**をクリックしてください。これはすべてのPIOプロセスと関数を含むライブラリ・ファイルです。

0 From the **Process** field click on **Firstappa**. A process is a grouping of Paragon functions and interconnections that form part of an application library in a given subsystem.

Process項目から**Firstappa**をクリックしてください。プロセスは与えられたサブシステムによるアプリケーション・ライブラリの一部を形成するParagon関数と相互接続のグループ設定です。

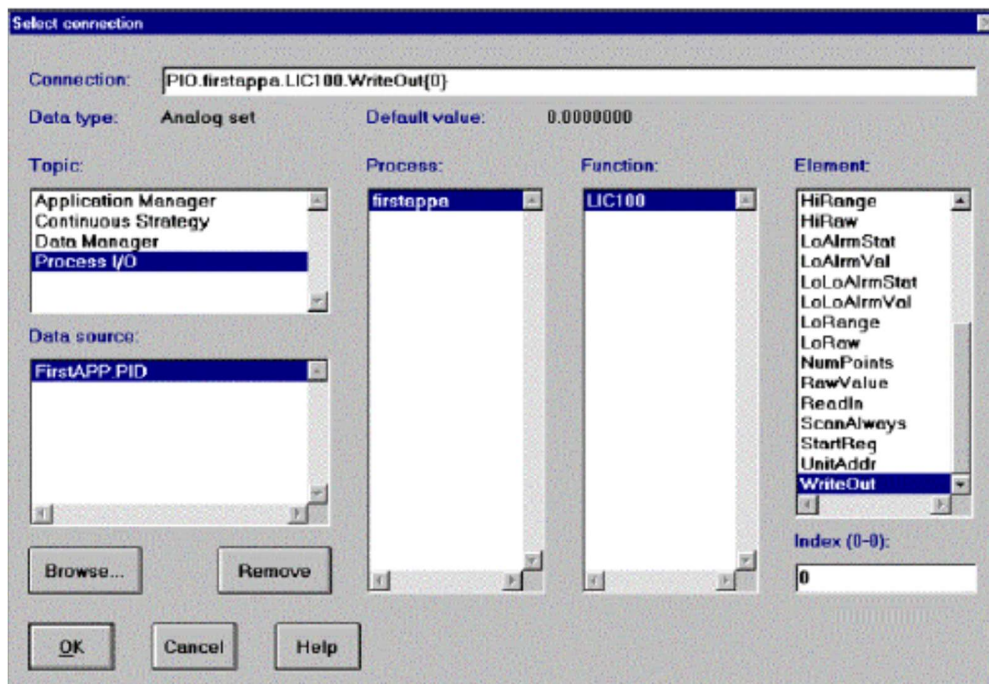
0 Click on the function **LIC100**, and the available elements will be listed in the Element list box. Use the scroll bars to move through the Element list box, and select **WriteOut**.

関数**LIC100**をクリックすると、利用可能な要素がElementリストボックスにリストされます。スクロールバーを使用してElementリストボックスを動かして、**WriteOut**を選択します。

5. Click on **OK**. The **Connection** field shows the function in the format:**Subsystem.Process.Function.Element{Index}**:

**OK**をクリックしてください。**Connection**項目は次のフォーマットで関数を示します:  
**Subsystem.Process.Function.Element{Index}**:

**PIO.Firstappa.LIC100.WriteOut{0}**.



This particular tag will cause the Data Entry field to “WriteOut” its value to another source. If you type a 1 in this field it will write out the value 1. By configuring the corresponding Data Display field with the tag name PIO.Firstappa.LIC100.ReadIn, it will read the WriteOut information and display a 1.

この特定のタグにより、別のソースの“WriteOut”にデータ入力項目の値を引き渡します。この項目に1をタイプすると、値1が書き出されます。タグ名PIO.Firstappa.LIC100.ReadInの対応するデータ表示項目の設定によって、それはWriteOutの情報を読んで1と表示します。

6. Click on **OK**, then **OK** again.

**OK**をクリックして、もう1度**OK**をクリックします。

**Edit object attributes**

Data Entry

Object attributes:

- Length
- Decimal places
- Connection
- Enable/Disable
- Entry limits
- Special cursor
- Quality
- Foreground color

Length:  Decimal places:

**Connection**

>>

0000-0000-0000-0000 >>

String connection

**Enable/disable connection**

>>

Invert

**Limits**

Limit entered values High:  Low:

Special runtime highlight cursor  Quality indicator by color

OK

Next object

Previous object

Enter composite

Leave composite

Show display

Cancel

Help

## Editing The List Box

### リストボックスの編集

In this section you are going to add selectable values to the list box. Each entry will be assigned a certain value by specifying its object attributes. By double-clicking on a specific entry, the value will appear in the Value 2 display field. このセクションでは、リストボックスに選択可能な値を加えます。各エントリーはそのオブジェクト属性の指定によりある値を割り当てられます。特定のエントリーをダブルクリックすることで、Value 2値2表示フィールドに値があらわれます。

**To add selectable elements to the list box:**  
リストボックスに選択可能な要素を追加するには：

1. Double-click on the list box to bring up its **Edit Object Attributes** dialog box.  
リストボックスをダブルクリックして**Edit Object Attributes**ダイアログボックスを出します。
2. In the **Object Attributes** list box click on **List** to enter the values for the list box and its connections.  
**Object Attributes**リストボックスで**List**をクリックして、リストボックスとその接続のために値を入れてください。
3. Click on the **List** radio button at the top of the dialog box to add new entries to the list box.  
リストボックスに新しいエントリーを加えるために、ダイアログボックスの一番上の**List**ラジオボタンをクリックしてください。
4. In the **Item** field type Set Value To 50. Click on **Add** to add it to the list. Add the following entries by typing them into the **Item** field and pressing **Add** after each one:  
**Item**項目に50の値をタイプしてください。**Add**をクリックするとリストに追加されます。**Item**項目にタイプして、その後**Add**を押すことで、以下のエントリーを追加してください。  
  
**0 Set Value To 75**  
75の値をセット  
  
**0 Set To Minimum**  
Minimumをセット  
  
**0 Set To Maximum**  
Maximumをセット  
  
**0 Set Value To 25**  
25の値をセット  
  
**0 Set Value To 45**  
45の値をセット
5. Click on **Set Value To 50**, and then click on the **Action** radio button to assign its action. The dialog box changes.  
**Set Value To 50**をクリックして、そのアクションを割り当てるために**Action**ラジオボタンをクリックしてください。ダイアログボックスが変更します。
6. In the **Actions** list box, scroll down and click on **Set Value**. The dialog box changes to show the parameters that need to be specified for this action.  
**Actions**リストボックスをスクロールして**Set Value**をクリックしてください。ダイアログ・ボックスはこのアクションのために指定が必要なパラメーターを示すために変わります。
7. Click on **Analog** for the type.  
typeで**Analog**をクリックしてください。
8. Click on the **Chevron**: next to the **Connection** field to locate the **LIC102** function, then do the following:

LIC102関数を見つけるために**Connection**項目の横の**Chevron: >>**をクリックして、それから次のことを行ってください :

0 From the **Topic** field select **Process I/O**.

**Topic**項目から**Process I/O**を選択

0 In the **Data Source** field click on **Firstappa.PID**.

**Data Source**項目で**Firstappa.PID**をクリック

0 From the **Process** field click on **Firstappa**.

**Process**項目から**Firstappa**をクリック

0 From the **Funtions** list box select **LIC102**.

**Funtions**リストボックスから**LIC102**を選択

0 From the **Elements** list box select **WriteOut**.

**Elements**リストボックスから**WriteOut**を選択

9. Click on **OK**.

**OK**をクリックしてください。

10. In the **Analog Value** field type **50**, then click on **Add** to add this action to the list. The action should read:

**Analog Value**項目に**50**とタイプして、リストにこのアクションを加えるために**Add**をクリックしてください。そのアクションは次のように読める筈です :

**Set Value:PIO.Firstappa.LIC102.WriteOut{0}:50**

The 50 at the end of the tag indicates the value to be written out.

50はタグの最後書き出される値を示します。

11. Repeat this procedure for the **Set Value To 75** entry. Click on it in the **List** list box, type **75** into the **Analog Value** field, then click on **Add**.

**Set Value To 75**エントリーのために手順を繰り返してください。 **List**リストボックスでそれをクリックして、**Analog Value**項目に**75**をタイプして、**Add**をクリックしてください。

**Note:** The dialog box remembers the parameters you set for the initial entry. You only have to enter information that has changed.

**注意 :** ダイアログボックスは初期のエントリーで設定したパラメータを記憶しています。あなたは変更する情報だけを入力すれば良いです。

12. For the other parameters enter the following information, and press **Add** after each action:

他のパラメータには次の情報を入力して、その後で**Add**を押してください :

0 For **Set To Minimum**, type 5 into the **Analog Value** field.

**Set To Minimum**には、**Analog Value**に5をタイプしてください。

0 For **Set To Maximum**, type 95 into the **Analog Value** field.

**Set To Maximum**には、**Analog Value**に95をタイプしてください。

0 For **Set Value To 25**, type 25 into the **Analog Value** field.

**Set Value To 25**には、**Analog Value**に25をタイプしてください。

0 For **Set Value To 45**, type 45 into the **Analog Value** field.

**Set Value To 45**には、**Analog Value**に45をタイプしてください。

13. Click on **OK** to exit the dialog box.

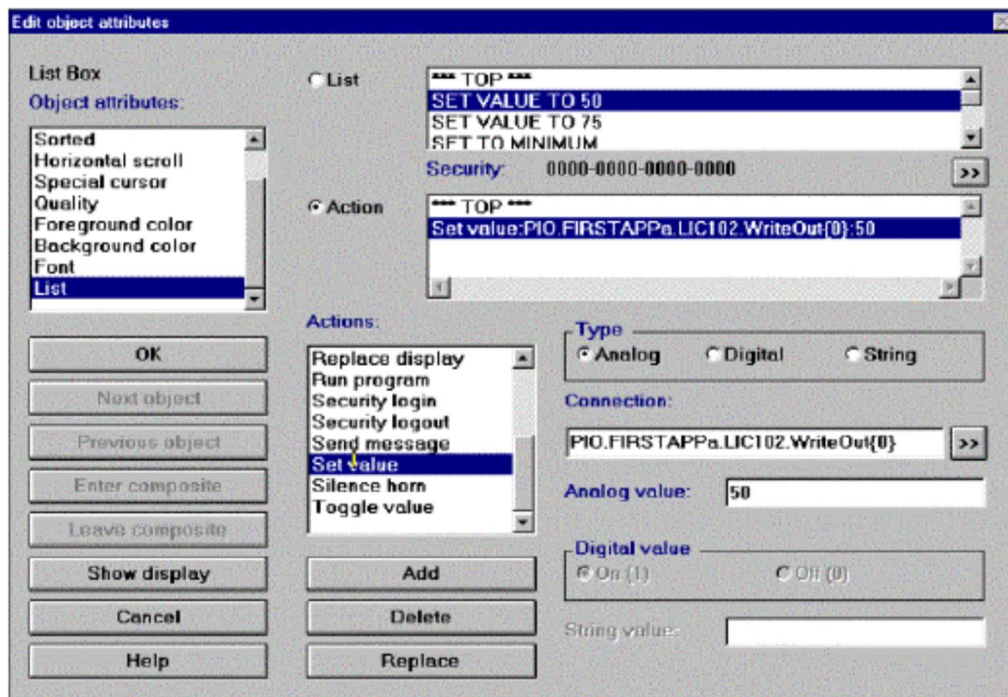
OKをクリックして、ダイアログボックスを終了してください。

This is configured similarly to the data entry field. In both cases the "WriteOut" Element is being used. In the next section you will configure the Value 2 Data Display field to "ReadIn" this value and display it.

これは同様にデータ入力項目を構成します。 どちらの場合も、"WriteOut"要素は使用されます。 次のセクションでは、Value 2データ・ディスプレイがこの値を"ReadIn"して表示するのを設定します。

**Note:** You can check your entries for the correct actions by clicking on each one in the List list box and then noting the displayed action in the Action list box. Use the Replace button to make any changes.

注意：Listリストボックスの各エントリーをクリックして、Actionリストボックスに表示されるアクションに注意することで、エントリーの正しいアクションをチェックできます。 Replaceボタンを使用して、任意の変更を行ってください。



14. You may need to resize the list box so that all the entries are visible.

エントリーがすべて見えるように、リストボックスをサイズ変更する必要があるかもしれません。

## Editing The Data Display Field

### データ表示項目の編集

The next step is to edit the display fields so that they read in the values from either the Enter Value 1 data entry field or the Select Value 2 list box. You will configure the Value 1 data display field first.

次のステップは、Enter Value 1データ入力項目あるいはSelect Value 2リストボックスの一方から値を読むように、表示項目を編集します。Value 1データ表示項目から設定してゆきます。

#### To edit the data display field:

データ表示項目を編集するには：

1. Double-click on the **Value 1** data display to bring up its **Edit Object Attributes** dialog box.

**Value 1**データ表示をダブルクリックして、**Edit Object Attributes**ダイアログボックスを出します。

2. To define the length of the data entry field, type 7 in the **Length** field and 2 in the **Decimal Places** field.

データ入力項目の長さを定義するために、**Length**項目で7をタイプして、**Decimal Places**項目で2をタイプします。

3. Click on the **Chevron:** next to the **Connection** field to enter the **LIC100** function from the PIO subsystem by, do the following:

PIO サブシステムから LIC100 関数を入力するために **Connection** 項目の **Chevron:**  をクリックして、次のことを行ってください:

0 From the **Topic** field select **Process I/O**.

**Topic**項目から**Process I/O**を選択

0 In the **Data Source** field click on **Firstappa.PID**.

**Data Source**項目で**Firstappa.PID**をクリック

0 From the **Process** field click on **Firstappa**.

**Process**項目から**Firstappa**をクリック

0 From the **Funtions** list box select **LIC100**.

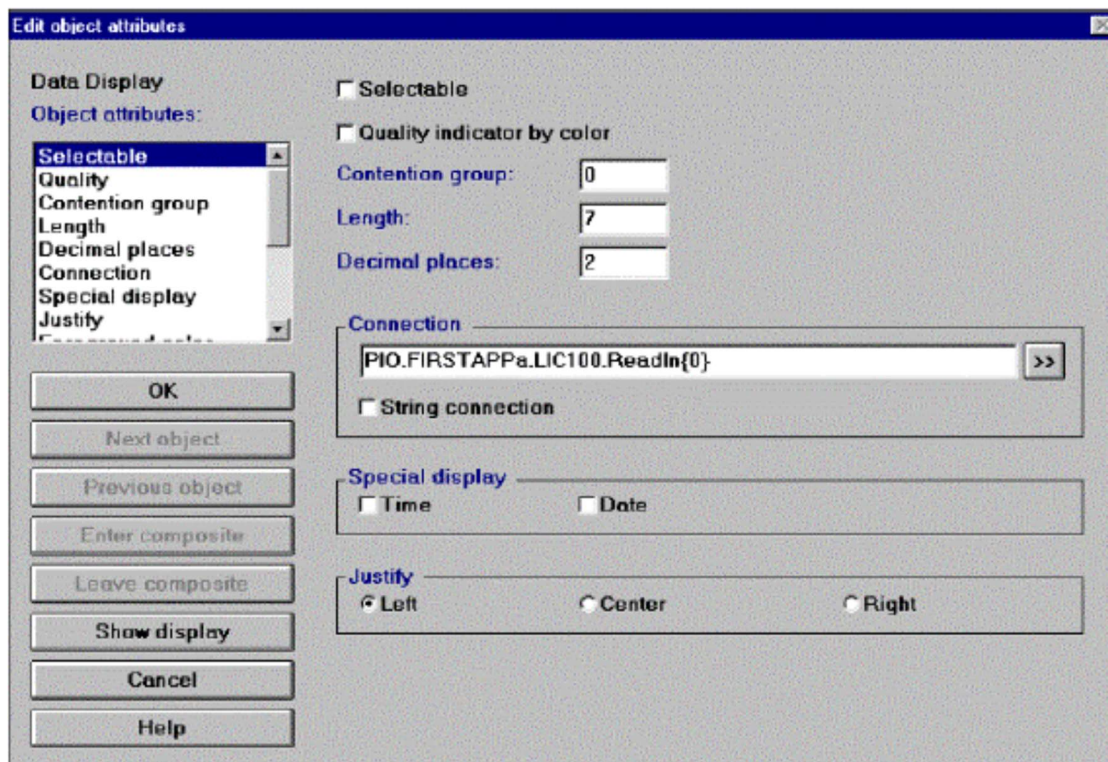
**Funtions**リストボックスから**LIC100**を選択

0 From the **Elements** list box select **ReadIn**.

**Elements**リストボックスから**ReadIn** を選択

4. Click on **OK**, then **OK** again.

**OK**をクリックして、再度**OK**をクリックしてください。



5. Repeat the above procedure for the second data display object, except enter the **LIC102** function into the **Connection** field:

2つ目のデータ表示オブジェクトのために、**Connection**項目にLIC102関数を入力する以外は手順を繰り返してください :

**PIO.Firstappa.LIC102.ReadIn{0}.**



## Saving The Display

### ディスプレイの保存

Your display is now finished. The final step is to save it.

ディスプレイはいま完成しました。最後のステップはそれを保存することです。

#### To save your display:

ディスプレイを保存するには：

1. From the Main menu select **File&Save**. This brings up the **Save Display** dialog box.  
メインメニューから、**File▶Save**を選択します。これは**Save Display**ダイアログボックスを出します。
2. In the **File Name** field type Firstapp.OIL, then press **OK**. Once the “Finished Compiling OI Display” appears, the files will be located in your working directroy.  
File Name項目にFirstapp.OILとタイプして、OKを押してください。“Finished Compiling OI Display”があらわれた時点で、ファイルは作業ディレクトリに存在します。
3. From the Main menu select **File&Close**.  
メインメニューから**File▶Close**を選択します。
4. Select **File&Exit** again to leave the **Operator Interface** builder and return to the **Application Manager**.  
**Operator Interface** builder から、**Application Manager** に戻るために再度 **File▶Exit** を選んでください。

## Application Manager

### アプリケーション・マネージャ

**Application File:** designates the local node, and the correct processes and display files to execute during runtime. These files are posted in the Control Panel.

**Application File:**ランタイムで実行するローカル・ノード、および正しいプロセスとディスプレイのファイルを指定します。これらのファイルはコントロール・パネルに掲示されます。

**Start-up Files:** are the files designated by the application file to execute during Runtime.

**Start-up Files:**ランタイムで実行するアプリケーションファイルによって指定されたファイルです。

The Application Manager is the nucleus of Paragon. It is used to access all the subsystems, and to execute Runtime. It also provides security access, subsystem diagnostics, and operator and system logs.

アプリケーション・マネージャはParagonの中心部分です。それは、サブシステムすべてにアクセスし、かつランタイムを実行するために使用されます。さらに、それはセキュリティ・アクセス、サブシステム診断ルーチン、そしてオペレーターおよびシステムログを提供します。

The next step is to create an **Application file** using the Application Manager. The files designated in the application file are known as the **start-up files**.

次のステップはアプリケーション・マネージャを使用して、**アプリケーション・ファイル(Application file)**を作成することです。アプリケーション・ファイルの中で指定されたファイルは、**スタートアップ・ファイル(start-up files)**として知られています。

In this exercise, the start-up file for the Process I/O is the **Firstappa** Configuration file (**Firstappa.PIC**) and the start-up file for the Operator Interface is the **Firstappa** display file (**Firstappa.OID**). The Application file is defined from the Control Panel.

この課題において、Process I/Oのためのスタートアップ・ファイルは**Firstappa** Configurationファイル(**Firstappa.PIC**)で、Operator Interfaceのためのスタートアップ・ファイルは**Firstappa**ディスプレイファイル(**Firstappa.OID**)です。アプリケーション・ファイルはコントロールパネルから定義されます。

## Control Panel

### コントロール・ファイル

The next step is to add your start-up files to the Control Panel. The Control Panel is used to execute subsystem start-up files during Runtime. It is also used to shut down executing subsystems, view subsystem diagnostics, and start and stop Control Panel extensions.

Your Enabler settings determine what is listed in your Control Panel.

次のステップはコントロールパネルにあなたのスタートアップ・ファイルを加えることです。コントロールパネルはランタイムでサブシステムのスタートアップ・ファイルを実行するために使用されます。またそれは、実行中のサブシステムのシャットダウン、サブシステム診断のビュー、およびコントロールパネル拡張の開始と停止に使用されません。

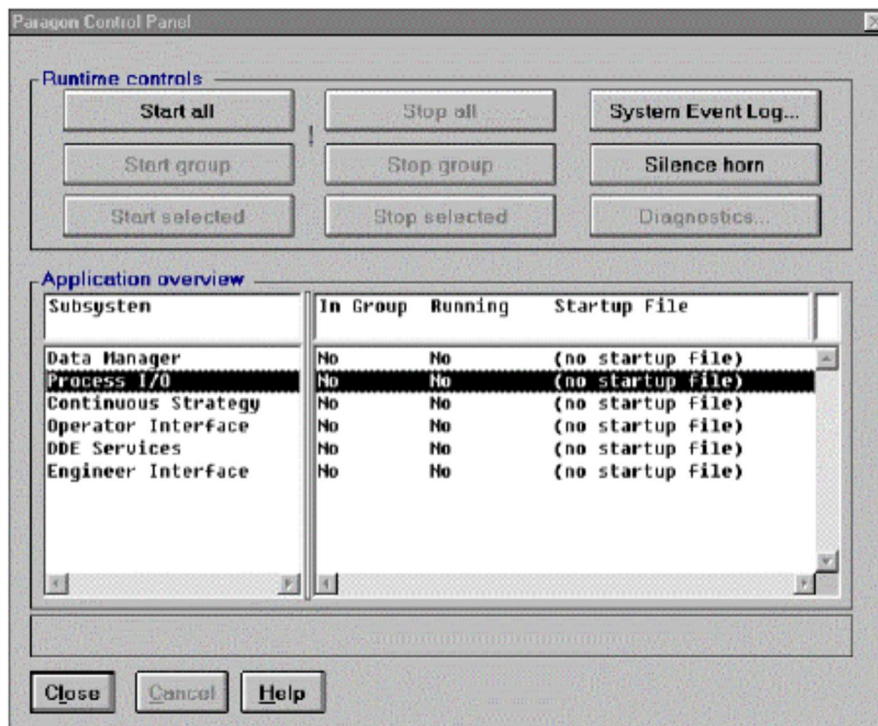
Enabler設定はコントロールパネルに何がリストされるかを定めます。

#### To add your start-up files to the Control Panel

コントロールパネルにスタートアップ・ファイルを追加するには：

1. From the Main menu select **Settings** Control Panel. The control Panel appears.

メインメニューから**Settings** Control Panelを選択します。コントロールパネルがあらわれます。



2. Under **Application Overview**, double-click on the **Process I/O** line to bring up the Subsystem Configuration Detail window.

**Application Overview**から**Process I/O**の行をダブルクリックして、Subsystem Configuration Detailウィンドウを出してください。

3. Click on the **Subsystem In Application Group** check box.

**Subsystem In Application Group**チェックボックスをクリックします。

4. Click on the **Chevron:** to choose the file from the **Start-Up File** field.

**Start-Up File**項目でファイルを選ぶために**Chevron: >>**をクリックしてください。

5. Click on **Firstappa.PIC**, then on **OK**. The file is shown with its full path in the **Start-Up File** field. Click on **OK**. The file is listed in the Start-up File column for the Process I/O.

**Firstappa.PIC**と**OK**をクリックしてください。ファイルは**Start-Up File**項目にフルパスで表示されます。**OK**をクリックしてください。ファイルはProcess I/OのStart-up File欄にリストされます。

6. Double-click on the **Operator Interface** line to bring up the **Subsystem Configuration Detail** window.

**Operator Interface**の行をダブルクリックして、**Subsystem Configuration Detail**ウィンドウを出してください。

7. Click on the **Subsystem In Application Group** field.

**Subsystem In Application Group**項目をクリックしてください。

8. Click on the **Chevron:** to view the display files. Select **Firstappa.OID** and then click on **OK**.

ディスプレイ・ファイルを見るために**Chevron: >>**をクリックしてください。**Firstappa.OID**を選択して、**OK**をクリックしてください。

9. Click on **OK** again and the **Firstappa** display file is listed in the **Start-up File** column for the Operator Interface.

もう一度**OK**をクリックすると、**Firstappa**ディスプレイファイルはOperator Interfaceの**Start-up File**欄にリストアップされます。

10. Click on the **Application Manager** to bring it to the foreground. From the Main menu select **Settings**8**Network Node Name**. **Local** should be listed in the **New Node Name** field. If not type it in, press **Tab**, then click on **OK**. This

names the local node (the station you are at) as Local.

**Application Manager**をクリックして、前面に表示してください。メインメニューから**Settings**▶**Network Node Name**を選択してください。**Local**は**New Node Name**項目にリストされているべきです。もし無ければ、それをタイプして、**Tab**を押して、**OK**をクリックしてください。これは**Local**とローカル・ノード（あなたがいるステーション）を命名します。

## Saving The Application File

### アプリケーション・ファイルの保存

The final step is to save the application file.

最終ステップはアプリケーション・ファイルを保存することです。

#### To save the Application file:

アプリケーション・ファイルを保存するには：

1. From the Application Manager Main menu select **File**▶**Save**.

アプリケーション・マネージャのメインメニューから**File**▶**Save**を選択してください。

2. In the dialog box, type Examples.APP into the **File Name** field, then click on **OK**.

ダイアログ・ボックスで、**File Name**項目にExamples.APPをタイプして、OKをクリックしてください。

You are ready to run your display. Examples.APP will be located in your working directory.

あなたは、あなたのディスプレイを実行する準備ができました。Examples.APPはあなたの作業ディレクトリーに位置するでしょう。

## Running The Application

### アプリケーションの実行

From the Control panel select your start-up files.

コントロールパネルからあなたのスタートアップ・ファイルを選択します。

#### To run your application:

アプリケーションを実行するには：

1. Click on the **Control Panel** dialog box to bring it to the foreground.

**Control Panel**ダイアログ・ボックスをクリックして、前面に表示してください。

2. Start the Data Manager without a configuration file. This enables event logging to track system errors. No enablers are needed to use this feature.

configuration fileなしでData Managerを開始してください（開始します?）。これは、イベントのログ収集がシステム・エラーを追跡するのを可能にします。enablersは、全くこの機能を使用するのに必要ではありません。

3. Double-click on the **Process I/O** and make sure that the Subsystem in Application Group checkbox is selected.

**Process I/O**をダブルクリックして、Application Groupチェックボックスでサブシステムが選択されていることを確かめてください。

4. Do the same thing for the **Operator Interface** line.

**Operator Interface**の行でも同じことを行ってください。

5. Click on **Start Group**.

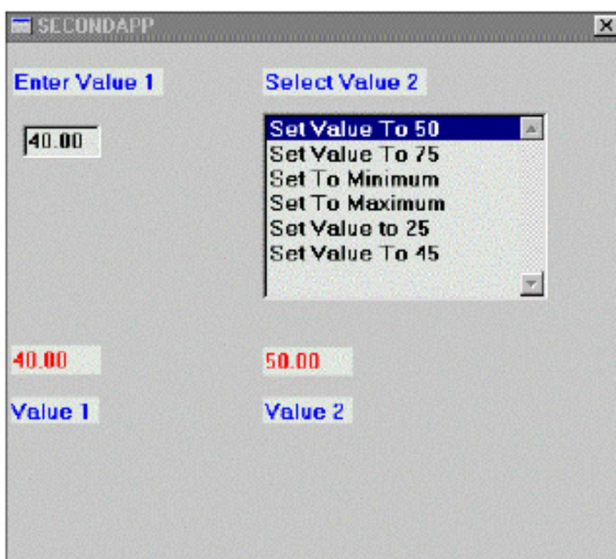
**Start Group**をクリックしてください。

## Using the Display ディスプレイの使用

### To see how the display works:

ディスプレイがどのように動作するか確かめるには：


1. Type 40 into the Enter Value 1 data entry field, then press **Enter**. The Value 1 field changes to 40.  
Enter Value 1データ入力項目に40をタイプして、**Enter**を押してください。Value 1項目が40に変わります。
2. Move the mouse to the list box and double-click on **Set Value To 50**. The Value 2 field changes to 50.  
リストボックスにマウスを移動させて、**Set Value To 50**をダブルクリックしてください。Value 2項目が50に変わります。



3. Double-click on **Set To Minimum**. The Value 2 field changes to 5. Continue selecting different values for both Value 1 and Value 2.

**Set To Minimum**をダブルクリックしてください。Value 2項目が5に変わります。Value 1とValue 2の両方に対して異なる値を選択し続けてください。

4. To close the display, click on the command button: in the upper-left corner of the display window. Select **Close** from the pull-down menu.

ディスプレイを閉じるには、ディスプレイ・ウィンドウの左上角のcommandボタン: をクリックしてください。プルダウン・メニューから**Close**を選んでください。

5. When you return to the Control Panel, stop the application by selecting **Stop Group**.

コントロールパネルに戻る場合、**Stop Group**を選択してアプリケーションを停止してください。

# Engineering Interface

**Engineering Interface (EI):** allows you to browse Paragon's servers' data to validate applications without having to build OI displays.

**Engineering Interface (EI):** OIディスプレイを構築する必要なしに、サーバ・アプリケーションを有効にするためにParagonサーバのデータをブラウズします。

Using EI, you can produce displays for all runtime elements. EI layouts can be saved and restored. It is an effective tool for learning about Paragon's extensive data type support, and how clients view the server data. You can also analyze DM history collections.

EIを使用すると、すべてのランタイム要素のためのディスプレイを形成することができます。EIレイアウトは保存し、復元することができます。Paragonの広範囲なデータ・タイプのサポートと、クライアントがどうサーバデータを見るかに関して学ぶための有効なツールです。さらに、DM履歴コレクションを分析することができます。

## To browse the data points for this application do the following:

データ・ポイントをブラウズするには、このアプリケーションのために下記を行ってください：

1. Highlight the **Engineering Interface** line and click on **Start Selected**. The Engineering Interface will appear.

**Engineering Interface**のラインをハイライトして、**Start Selected**をクリックしてください。Engineering Interfaceがあらわれます。

2. Click on **Search»Browse**, select the following PIO points, then click on **Add** after each one:

**Search»Browse**をクリックして、次のPIOポイントを選択し、そして、それぞれの後に**Add**をクリックします：

0 PIO.firstappa.LIC100.WriteOut.

0 PIO.firstappa.LIC100.ReadIn

0 PIO.firstappa.LIC102.WriteOut

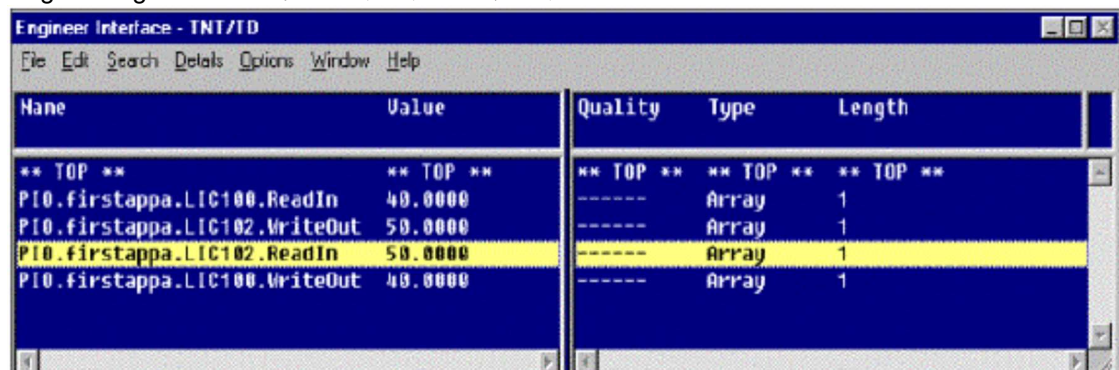
0 PIO.firstappa.LIC102.ReadIn

3. Select **File»Save**. Name the application **Firstappa.EIC**.

**File»Save**を選択。アプリケーションを**Firstappa.EIC**と名付けます。

The Engineering Interface should look like the following:

Engineering Interfaceは次のように見える筈です：



The screenshot shows a window titled "Engineer Interface - TNT/TD" with a menu bar (File, Edit, Search, Details, Options, Window, Help). The main area contains a table with four columns: Name, Value, Quality, and Type. The table lists four PIO points, with the third one highlighted in yellow.

Name	Value	Quality	Type	Length
** TOP **	** TOP **	** TOP **	** TOP **	** TOP **
PIO.firstappa.LIC100.ReadIn	40.0000	-----	Array	1
PIO.firstappa.LIC102.WriteOut	50.0000	-----	Array	1
PIO.firstappa.LIC102.ReadIn	50.0000	-----	Array	1
PIO.firstappa.LIC100.WriteOut	40.0000	-----	Array	1

**Name Field:** lists the PIO points you selected.

**Name領域：**選択したPIOポイントがリストされます。

**Value Field:** reflects the numbers you entered for the Value 1 and Value 2 data entry fields. The WriteOut values represent the data that is entered into the data entry fields. The ReadIn value collects data from the data entry fields and displays it in the appropriate data display field.

**Value領域：**Value 1およびValue 2のデータ・エンタリー・フィールドに登録した数を反映します。WriteOutの値は、

データ・エントリー・フィールドに入力されるデータを表わします。ReadInの値はデータ・エントリー・フィールドからデータを集めて、適切なデータ表示領域にそれを表示します。

**Quality Field:** displays the quality bits set for the selected point.

**Quality領域:** 選択したポイントの品質ビットセットを表示します。

**0 Fail (F):** indicates that an error has occurred reading a value from the device.

**Fail (F):**装置から値を読んでエラーが生じたことを示します。

**0 Unavailable (N):** indicates that the requested data is not available.

**Unavailable (N):**要求されたデータが利用可能ではないことを示します。

**0 Alarm (A):** indicates that the value is in alarm.

**Alarm (A):**値がアラーム範囲にあることを示します。

**0 Manual (M):** indicates that the data is in the initial value state and has not been read from the hardware.

**Manual (M):**データがデフォルト値状態でハードウェアから読まれていないことを示します。

**0 User (U):** can be set by user and developed using the Client Development Kit (CDK) or Block Development Kit (BDK). Can also be set by interactive trends.

**User (U):**Client Development Kit(CDK)もしくはBlock Development Kit(BDK)を使用して開発することでユーザがセットできます。また、interactive trendsで設定することができます。

**0 Start (S):** is generated by the Data Manager. Identifies the first value in a data collection.

**Start (S):**はData Managerによって生成されます。データ・コレクションの最初の値を識別します。

Since there are no errors, hyphens appear in this field.

エラーが無ければ、ハイフンがこの領域にあらわれます。

**Type領域:** displays the data type of the highlighted parameter. In this case Array.

データ・タイプのハイライトされたパラメータを表示します。この場合はArray (配列)。

**0 Length Field:** for the highlighted parameter it displays the array field length or the number of possible bites for a string parameter.

ハイライトされたパラメータに関しては、Stringパラメータのための配列フィールド長あるいは可能なビット数を表示します。

## Chapter Summary

### 章のまとめ

Congratulations! You now know how to create a basic display using the Process I/O, Operator Interface, and Application Manager. You also learned how to use the Application Wizard to create expandable processes and functions. In the next chapter you will use the Continuous Strategy to perform calculations on the values from the Process I/O subsystem.

おめでとう！いまあなたは、**Process I/O**、**Operator Interface**、そして **Application Manager**を使用して、基礎的なディスプレイを作成する方法を知っています。また、あなたは拡張可能なプロセスと関数を作成するのに**Application Wizard**を使用する方法を学びました。次の章では**Process I/O**サブシステムからの値の計算を行うのに**Continuous Strategy**（連続戦略）を使用します。



# Using The Continuous Strategy

## Continuous Strategyの使用について

### In This Chapter:

この章には :

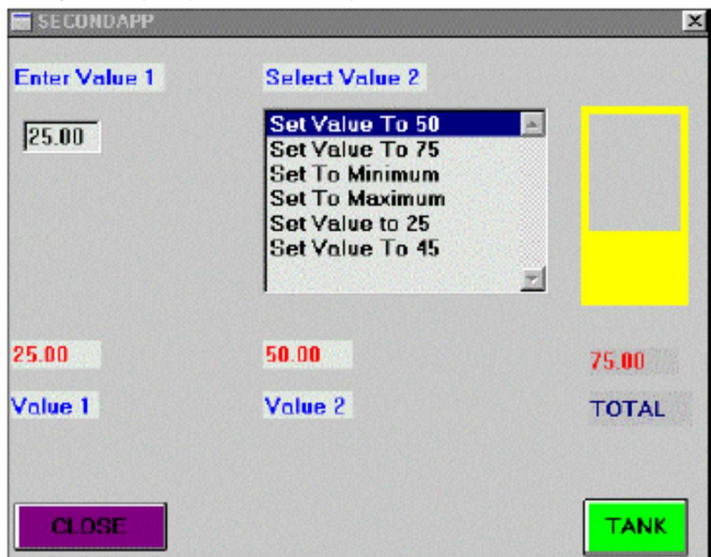
ä Continuous Strategy 77

ä Operator Interface 88

ä Chapter Summary 96

In this chapter you are going to create a Control strategy with the Continuous Strategy builder. The strategy will sum the values from the Process I/O and add a bar chart to your Firstapp display to animate those values. Finally, you will create push buttons that close the Firstapp display and load other displays that you will create in the following chapters. When you are finished your display will look like the following:

本章ではContinuous Strategy builderを使って制御戦略(Control strategy)を作成します。戦略は、Process I/Oからの値を合計して、それらの値を活気づけるためにFirstappディスプレイに棒グラフを加えます。最後には、Firstappディスプレイを閉じて、次章で作成する他のディスプレイをロードする押しボタンを作成します。完了すると、ディスプレイは次のように見えるでしょう :



In the completed display, the Total field will sum the values from the Value 1 and Value 2 data display fields. The bar chart above the total field will rise and fall according to that value.

完成したディスプレイでは、Total領域は、Value 1およびValue 2のデータ表示領域からの値を合計します。Total領域の上の棒グラフはその値により上昇そして下降します。

# Continuous Strategy

The Continuous Strategy subsystem is used to graphically create your strategy. It executes real-time, continuous control of your process, guaranteeing the on-time execution of critical process algorithms. High-level process information is generated from Process I/O values by the use of:

**Continuous Strategy** (連続戦略) サブシステムはグラフィカルに戦略を作成するために使用されます。それは、重要なプロセス・アルゴリズムが時間通りに実行されることを保証して、あなたのプロセスのリアルタイムの連続制御を実行します。高レベルのプロセス情報は次のものを使用して**Process I/O**値から生成されます：

Math

Boolean

Direct and supervisory control

ASCII string functions

Similar to the Process I/O subsystem, the Continuous Strategy is configured in a two-stage process within the CS Builder.

Process I/Oサブシステムと同様に、Continuous StrategyはCS Builderで2ステージの過程で構成されます。

## Creating A Strategy Display Window

### 戦略ディスプレイ・ウィンドウの作成

Each strategy is designed in its own display window and can consist of multiple layers. A strategy is used to perform calculations on a data source.

各戦略はそれ自身のディスプレイ・ウィンドウで設計されており、複数のレイヤーから成ります。戦略はデータソース上で計算を行なうために使用されます。

#### To create a new strategy:

##### ▶ 新しい戦略の作成：

1. From the Application Manager Main menu, select **Builders**▶**Continuous Strategy**.

Application Managerのメイン・メニューから、**Builders**▶**Continuous Strategy**を選択してください。

2. From the Main menu select **File**▶**New**.

メイン・メニューから、**File**▶**New**を選択します。

3. When the New Strategy dialog box appears, type Firstappa in the **Process Name** field. Leave the other parameters at their default values, then press **OK**. This creates the first process block on your display labeled Firstappa.

**New Strategy**ダイアログ・ボックスがあらわれたら、**Process Name**領域にFirstappaをタイプしてください。他のパラメータにはデフォルト値を残して、**OK**を押してください。これはFirstappaとラベルされたディスプレイに最初のプロセス・ブロックを作成します。

The CS Builder is similar to the OI Builder, and it contains the following:

CS BuilderはOI Builderに似ています。また、それは下記を含んでいます：

Title Bar

Main Menu

Status Line

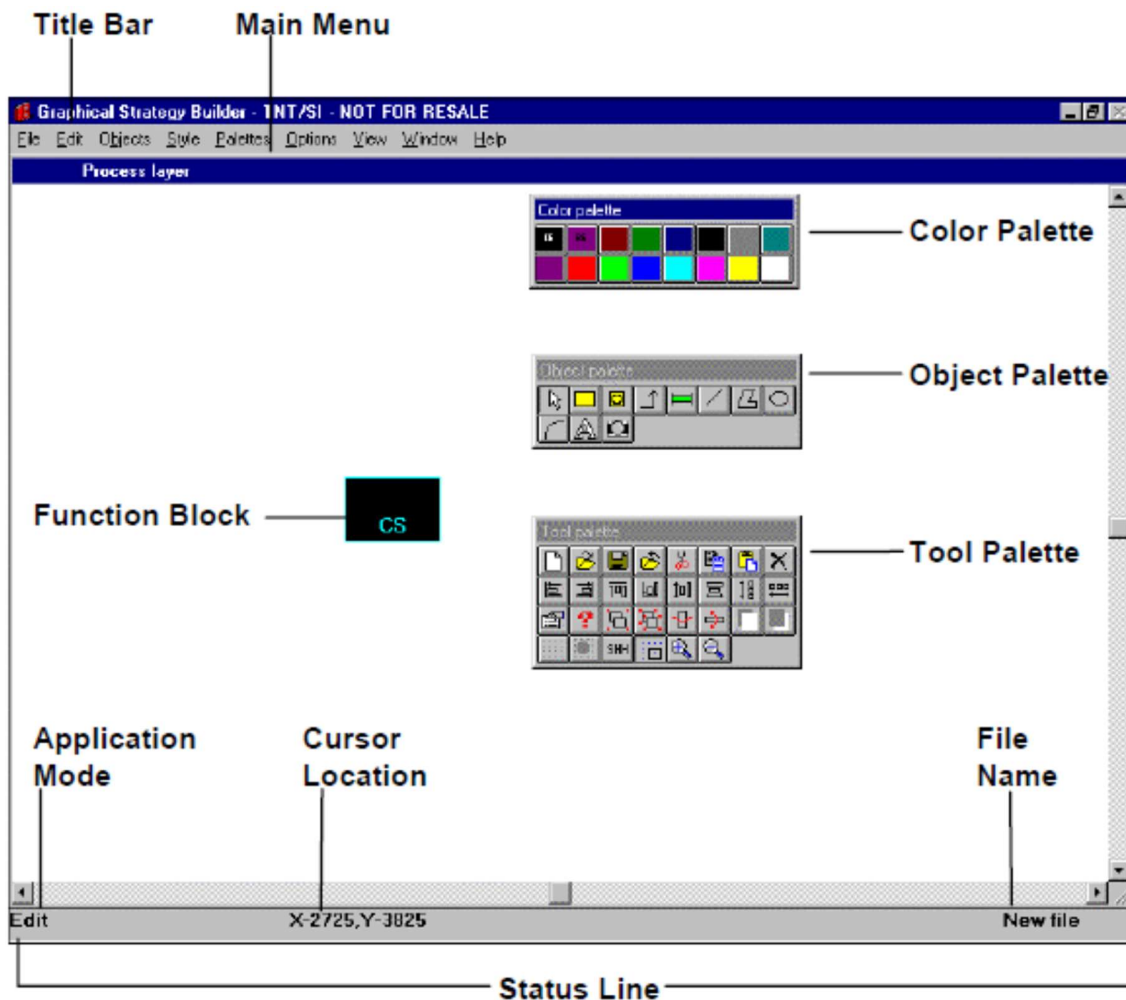
Object Palette

## Color Palette

**Strategy Objects:** are used to place components that make up your strategy.  
**Strategy Objects:** あなたの戦略を構築するコンポーネントを置くのに使用されます。

**Annotation Objects:** graphically illustrate your strategy.  
**Annotation Objects:** 戦略をグラフィカルに説明します。

The Object Palette consists of **Strategy Objects** and **Annotation objects**. Strategy objects are used to place components that make up your strategy. Annotation objects graphically illustrate your strategy. Object Paletteは、Strategy ObjectsとAnnotation objectsから成ります。Strategy objectsは、あなたの戦略を構築するコンポーネントを置くために使用されます。Annotation objectsは、あなたの戦略をグラフィカルに説明します。



The display window only shows a portion of the area in which you can build your strategy. Use the scroll bars to view different areas of the display. By using the **Options & Window Parameters** command you can define the location, size, and background color of the window.

ディスプレイ・ウィンドウは戦略を構築可能なエリアの一部のみを示します。ディスプレイの異なるエリアを見るにはスクロール・バーを使用してください。**Options > Window Parameters**の命令を使用して、ウィンドウの位置、サイズ、および背景色を定義できます。

## The Object Palette

The Object Palette contains objects that are used to configure your display. Objects are selected from the palette by clicking on the appropriate icon and then clicking on the display window where you want to place the object.

Object Paletteは、ディスプレイを構成するために使用するオブジェクトを持っています。オブジェクトはパレットから適切なアイコンをクリックして、オブジェクトを置きたいと思うディスプレイ・ウィンドウをクリックすることで選択されます。

**Edit Tool:** Select move and resize objects.



**Edit Tool:** 選択したオブジェクトの移動とサイズ変更

**Function Block tool:** Place and configure function blocks.



**Function Block tool:** 関数ブロック (function block) の設置と設定

**Terminator tool:** Connect processes and subsystems between layers.



**Terminator tool:** レイヤー間のプロセスとサブシステムを接続

**Connection tool:** Connect function blocks.



**Connection tool:** 関数ブロック (function block) の接続

**Conduit tool:** Connect process blocks.



**Conduit tool:** プロセス・ブロック (process block) の接続

**Polyline tool:** Draw line segments.



**Polyline tool:** 線分の描画

**Polygon tool:** Draw multi-sided objects.



**Polygon tool:** 複数面を持つオブジェクトの描画

**Ellipse tool:** Draw circles and ellipses.



**Ellipse tool:** 円と楕円の描画

**Arc tool:** Draw open or closed arcs.



**Arc tool:** 開いた弧あるいは閉じた弧の描画

**Text tool:** Create a text label.



**Text tool:** テキストラベルの作成

**Bitmap tool:** Place pre-designed bitmap objects.



**Bitmap tool:** デザイン済みのビットマップ・オブジェクトの設置

## Process Blocks

**Process Blocks:** are used to organize your strategy into processes.

**Process Blocks:** プロセスに戦略を編成するのに使用されます。

**Process Blocks** are used to organize your strategy into processes. A process can consist of:

**Process Blocks** はプロセスに戦略を編成するのに使用されます。プロセスは次のものから成ります:

A specific task.

特定のタスク

A relatively independent manufacturing unit of your plant.

工場の比較的独立した生産ユニット

Equipment that operates somewhat independently from the rest of your plant.

工場の他から多少独立して作動する設備

Process blocks help to optimize your strategy in the following ways:

プロセス・ブロックは、次の方法で戦略を最適化するのを支援します：

Logically partition and organize your strategy, which aids in future modification, expansion, and troubleshooting.

戦略を論理的に切り分けて編成することで、将来の修正、拡張、およびトラブルシューティングを支援します。

Build complex strategies by organizing your strategy in ways that mirror your process.

プロセスを反映する方法で戦略を編成して、複雑な戦略を構築します。

Distribute Paragon functionality across multiple computers connected by a local area network (LAN). Processes can be assigned to run on a network station. Stations only load and execute function block processes that are assigned to that station. A process can be easily reassigned to a different node without changing the strategy or renaming any functions. Multiple process layers can execute on the same node.

ローカル・エリア・ネットワーク(LAN)で接続した複数のコンピュータにParagon機能を分散してください。プロセスはネットワーク・ステーションで実行するために割り当てることができます。ステーションは、そのステーションに割り当てられる関数ブロック・プロセスをロードし実行するだけです。プロセスは戦略の変更やどの関数の名前変更もなしに、簡単に別のノードに再割り当てすることができます。複数プロセス・レイヤーが同じノード上で実行できます。

Duplicate functionality without changing all the function names internal to the process block. As in other server subsystems, functions are accessed by **Subsystem.Process.Function.Element**. Therefore, a unique process name gives the function a unique name.

機能の複製はプロセス・ブロック内の関数名を変更しません。他のサーバー・サブシステムでのように、関数は**Subsystem.Process.Function.Element**によりアクセスされます。したがって、ユニークなプロセス名は、ユニークな名前を関数に与えます。

Facilitate troubleshooting by allowing individual processes to be shut down during runtime to isolate potential problems. Process blocks can only be placed in the Process Layer, the initial window of your Strategy Builder.

各プロセスが潜在的な問題を分離しランタイムの間にシャットダウンしてトラブルシューティングを促進します。プロセス・ブロックはプロセス・レイヤー（Strategy Builderの初期ウィンドウ）にしか設置できません。

## Building The Strategy

### 戦略を組み立てる


The first step in building your strategy is to create the **Firstappa** process block. To connect to the Process I/O subsystem functions you must place a second process block.

あなたの戦略を構築する第一歩は**Firstappa**プロセス・ブロックを作成することです。Process I/Oサブシステムの関数に接続するには、別のプロセス・ブロックを置く必要があります。

#### To place a second process block:

▶ 2つめのプロセス・ブロックを設置するには：


1. Click on the **Function Block Tool**:

**Function Block Tool:**  をクリックします。

2. Move the mouse to the right of the first block and paste another.

最初のブロックの右にマウスを動かして、別のものを貼ってください。

3. Align the blocks by selecting the **Edit Tool**: and then clicking on the block to select it. Hold the mouse button down and drag the block to its new location.

**Edit Tool:**  を選択して、ブロックを選択するためにクリックして、ブロックを整列します。マウスボタンを押したまま、新しい位置へブロックをドラッグしてください。

4. Double-click on the first Process Block to open its **Configure Function Block** dialog. Here you enter the specific parameters for the block including the name, process type, and alarm destinations.

最初のプロセス・ブロックをダブルクリックして、**Configure Function Block**ダイアログを開いてください。ここで名前(name)、プロセス・タイプ(process type)、および警報先(alarm destinations)のブロックに含まれる特定パラメータを登録します。

5. From the **Type Topic** list box make sure **Continuous Strategy** is selected. Click on **OK**.

**Type Topic** リストボックスから、**Continuous Strategy**が選択されるのを確認してください。**OK**をクリックします。

6. Double-click on the second block and configure it by doing the following:

2つ目のブロックをダブルクリックして、次の設定を行ってください：

0 From the **Type Topic** scroll box select **Process I/O**. This block represents the Process I/O functions you created in the last exercise.

**Type Topic** スクロールボックスから、**Process I/O**を選択します。このブロックは、最後の課題で作成したProcess I/O関数をあらわします。

0 Select the PIO library file by clicking on the **Chevron:** next to the **Data Source** field. Select the file **Firstappa.PID**, then click on **OK**. If you do not see the Library file **Firstappa.PID** then double-click on the two periods (..) in the directories box. Scroll through the list and double-click on **Firstapp**. The library file **Firstappa.PID** should appear in the Files box:

**Data Source**領域の横の**Chevron:**をダブルクリックして、PIOライブラリ・ファイルを選択してください。ファイル**Firstappa.PID**を選択して、**OK**をクリックしてください。ライブラリ・ファイル**Firstappa.PID**が見つからない場合、ディレクトリ・ボックス内の2つのピリオド(..)をダブルクリックします。リストをスクロールして、**Firstapp**をダブルクリックしてください。ライブラリ・ファイル**Firstappa.PID**がファイル・ボックスにあらわれるに違いありません：

0 Click on the **Chevron:** next to the name field and select **Firstappa**. Process blocks can have the same name if they are different types. In this case you have a Firstappa process for the CS and another for the PIO. Within the same process type, a process block must have a unique name.

**Name**領域の横の**Chevron:**をダブルクリックして、**Firstappa**を選択してください。それらが異なるタイプならば、プロセス・ブロックは同じ名前を持つことができます。この場合CS のための**Firstappa**プロセスおよびPIOのための別のものがあります。同じプロセス・タイプ内では、プロセス・ブロックはユニークな名前を持っている必要があります。

Configure function block

Name:  >> Type:  >>

Description:  Scan rate (Sec):

Type  
Topic:   EXTERNAL CS

Data source:  >> X

Alarm destination:  >>

Collection ID:

DM.{LocalNode}.Alarm.InPort >>

DM.{LocalNode}.Alarm.InPort >>

DM.{LocalNode}.Alarm.InPort >>

DM.{LocalNode}.Alarm.InPort >>

DM.{LocalNode}.Alarm.InPort >>

DM.{LocalNode}.Alarm.InPort >>

DM.{LocalNode}.Alarm.InPort >>

DM.{LocalNode}.Alarm.InPort >>

DM.{LocalNode}.Alarm.InPort >>

Page 1 of 1

Process

OK Next Previous Show Expand Cancel Help

7. Click on **OK** again to exit the dialog box.

ダイアログ・ボックスを出るために再び**OK**をクリックしてください。

You have created two Firstappa process blocks. One for the CS and the other for the PIO. In order to pass the acquired data from the Process I/O process to the Continuous Strategy for calculation and control, you need to connect the blocks with a conduit.

あなたは2つのFirstappaプロセス・ブロックを作成しました。CSのための1つとPIOのためのもう片方です。計算とコントロール用の獲得したデータをProcess I/OプロセスからContinuous Strategyへ渡すために、ブロックを情報伝達ルート(conduit)に接続する必要があります。


## Connecting Process Blocks

### プロセス・ブロックの接続

**To connect the two process blocks:**

2つのプロセス・ブロックを接続するには：

1. Click on the **Conduit Tool** , and then on one of the process blocks.

**Conduit Tool:**  を、その後でプロセス・ブロックの一方をクリックしてください。

2. Slide the mouse outside the process block and begin to draw the connection line to the other process block (the order is insignificant). Click on it to complete the connection. Your display should look like the following:

プロセス・ブロックの外側にマウスを滑らせて、もう一方のプロセス・ブロック(順序は重要ではありません)に接続ラインを引いてください。接続を完了するために、それをクリックしてください。ディスプレイは次のように見える筈です：



## Strategy Layers

### 戦略レイヤー

Strategies consists of multiple layers, (windows) each adding more detail to the overall strategy. The initial layer (process layer) is where process blocks are defined. The second layer (expanded layer) is where functions are defined. CS blocks are the only expandable blocks.

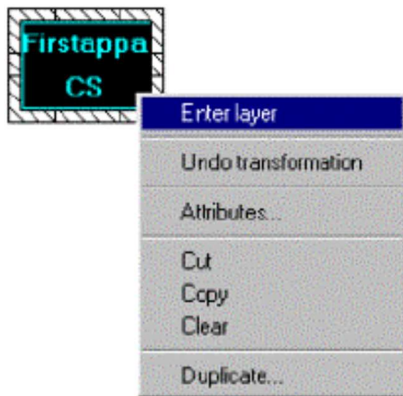
全体的な戦略により多くの詳細を追加して、戦略は複数のレイヤー（ウィンドウ）から成ります。初期レイヤー（プロセス・レイヤー）はプロセス・ブロックが定義される場所です。第2レイヤー（拡張レイヤー）は関数が定義される場所です。CSブロックは数少ない拡張可能なブロックです。

### To enter the Expanded layer

#### 拡張レイヤーの入力

1. Click on the **Edit Tool:** and enter the CS block by rightclicking on it and selecting **Enter Layer**. You are now in the second process layer.

**Edit Tool:**  をクリックして、その上で右クリックし **Enter Layer** を選択してCSブロックを入力してください。



**Note:** You can later return to the initial process layer, by selecting **View»Process Layer**.

注意：あとで**View»Process Layer**を選択して、初期プロセス・レイヤーに戻ることができます。

## Creating Function Blocks

### 関数ブロックの作成


Function blocks are used to define analog, discrete, and string handling functions. They are pictorial representations of algorithms that need to be executed during Runtime. They include math calculations, string manipulation, boolean logic, and PID loops. In this section you will sum the values from the Process I/O using an Add block.

関数ブロックはanalog, discrete, および string を取り扱う関数を定義するために使用されます。それはランタイムで実行される必要のあるアルゴリズムの図形による表現です。それらは数学計算、文字列処理、ブール論理およびPIDループを含みます。このセクションでは、Addブロックを使用して、Process I/Oからの値を合計します。


### To create the ADD block:

#### ADDブロックの作成：

1. Click on the **Function block Tool:** , then click on the center of the Strategy window.

**Function block Tool:**  をクリックして、次に戦略ウィンドウの中心をクリックしてください。

2. Click on the **Edit Tool:** , then click on the **Function block** to select it.

**Edit Tool:**  をクリックして、その後、それを選択するために**関数ブロック**をクリックします。



3. Move the cursor over the frame of the block. When the cursor becomes a double arrow, use it to resize the block to approximately 1 inch by 1 inch.

ブロック枠の上にカーソルを移動させます。カーソルが二重矢になるとき、それを使用して、ブロックをおよそ1インチ×1インチまでサイズ変更してください。

4. Double-click on the block to open its **Configure Function Block** dialog box.

ブロックをダブルクリックして**Configure Function Block**ダイアログボックスを開いてください。

5. In the **Name** field type Sum. To select the type of function block click on the **Chevron**: next to the **Type** field.

**Name**領域にSumとタイプします。関数ブロックのタイプを選ぶには、**Type**領域の横の**Chevron**: [ ] をクリックしてください。

Paragon provides over 60 types of function blocks. These are all organized into seven high-level categories:

Paragonは60以上のタイプの関数ブロックを提供します。これらはすべて7つの高位カテゴリへ編成されます：

0 Input/Output

0 Logic/Selection

0 Calculation

0 Supervisory Control

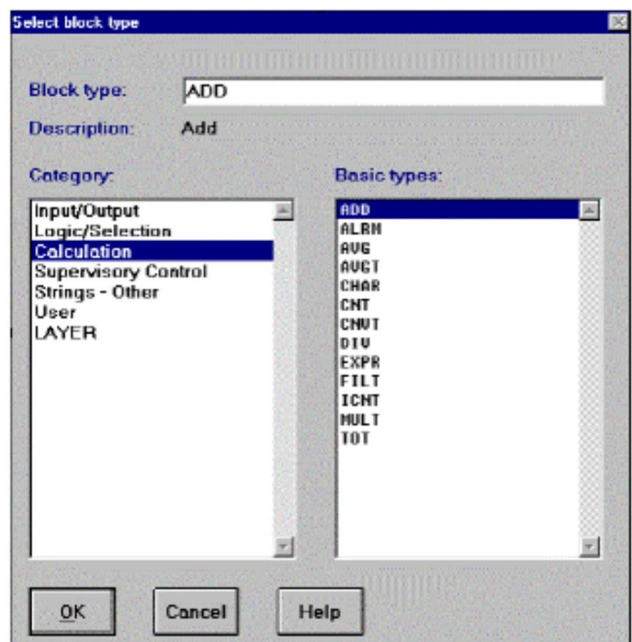
0 Strings-Other

0 Layer

0 User

6. From the **Category** list box select **Calculation**. All blocks contained in this category are under **Basic Types**.

**Category** リストボックスから、**Calculation** を選択します。このカテゴリに含まれるすべてのブロックが**Basic Types**の下にあります。



7. Click on **Add**, then on **OK**.

**Add**そして**OK**をクリックします。

## Configuring The ADD Block Dialog Box

### ADDブロック・ダイアログボックスの設定

#### To configure the Add Block Dialog box:

▶ Addブロックのダイアログボックスを設定するには：

1. From the **Scan Rate** list box select **0.25**. This is how often the specified points are sampled. The values are shown in seconds. Typically, scan rates are greater than one second.

**Scan Rate** リストボックスから**0.25**を選択します。これは指定ポイントが抽出される割合です。値は秒で示されています。通常、スキャン・レートは1秒より大きいです。

2. In the **High Range** field type 200. This field determines the block's maximum output value.

**High Range**領域に200をタイプします。この領域はブロックの最大出力値を定めます。

3. In the **High Limit** field type 200. This field determines the maximum quantity that the block output is allowed to achieve.

**High Limit**領域に200をタイプします。この領域はブロックの出力が達成することができる最大量を定めます。

4. Click on **OK**. The block changes color to indicate that it has been configured.

**OK**をクリックします。ブロックは、それが設定されたのを示すために色を変えます。

## Connecting Process I/O Functions

### Process I/O関数の接続

**Terminator**: allows you to connect a function block to an external block.

**Terminator**: 外部ブロックに関数ブロックを接続することを可能にします。


The next step is to connect the analog functions from the Process I/O to the Add block. This is done using a **Terminator**. A Terminator allows you to connect a function block to an external block. In this case, you are going to connect two process blocks from different layers of the same strategy.


次のステップは、Process I/OからAddブロックにアナログ関数を接続することです。これはTerminatorを使用して完成します。Terminatorは、外部ブロックに関数ブロックを接続することを可能にします。この場合、同じ戦略の異なるレイヤーからの2つのプロセス・ブロックを接続するつもりです。

#### To connect the Process I/O functions to the ADD block:

▶ **ADDブロックにProcess I/O関数を接続するには:**

1. Click on the **Terminator Tool:** and paste one to left of the **Add** block and another to the right.


**Terminator Tool:**  をクリックして、**Add**ブロックの左に1つ貼り付け、右に別のもう1つを貼り付けます。

2. Select the **Edit Tool:** , then double-click on the first Terminator and type Val1Term in the **Name** field. Click on **OK**  
**Edit Tool:**  を選択して、次に最初のTerminatorをダブルクリックして、**Name**領域でVal1Termをタイプしてください。**OK**をクリックしてください。

3. Double-click on the second Terminator and type Val2Term in the name field. Click on **OK**.

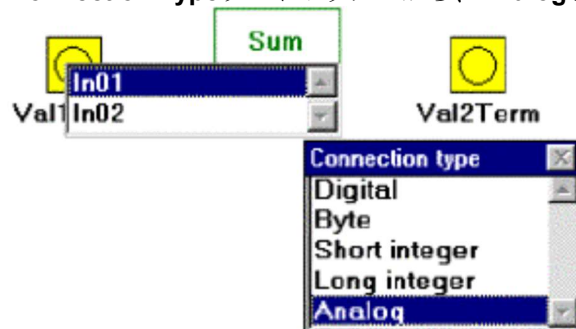
2つめのTerminatorをダブルクリックして、Name領域でVal2Termをタイプしてください。**OK**をクリックしてください。

4. Select the **Connection Tool:** and click on Val1Term. Slide the mouse to the **Add** block and click on it to complete the connection. When the list box appears click on **In01** to complete the connection.

**Connection Tool:**  を選択して、Val1Termをクリックします。**Add**ブロックにマウスを滑らせて、それをクリックして、接続を完了してください。リストボックスがあらわれたら、**In01**をクリックして、接続を完了してください。

5. From the **Connection Type** list box select **Analog**.

**Connection Type** リストボックスから、**Analog**を選択してください。



6. Click on **Val2Term** and draw another connection into the **Add** block. Select **In02** from the list box to complete the connection.

**Val2Term** をクリックして、別の接続を**Add**ブロックに引っ張ってください。接続を完了するためにリスト・ボックスから**In02**を選択してください。



## Specifying The Process I/O Functions

### Process I/O関数の指定

**To use terminators to specify Process I/O functions:**

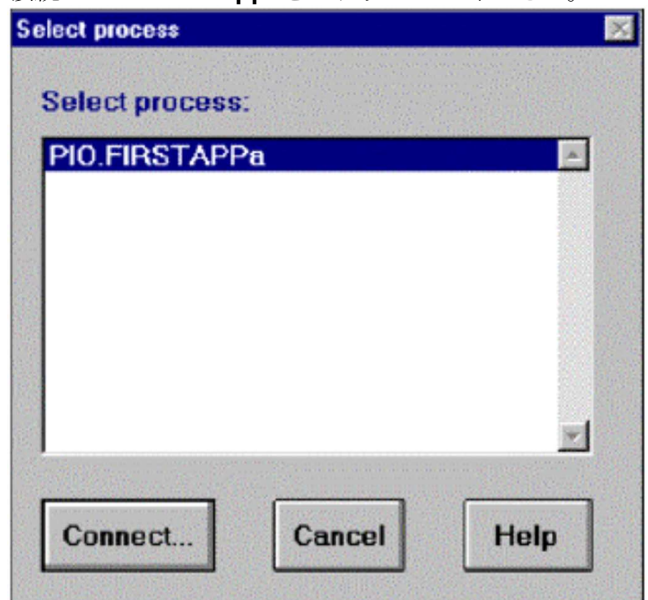
**Process I/O関数を指定するためにターミネータを使用するには :**

1. From the Main menu select **Objects&External Connections**. The **Select Process** dialog box appears showing all the process blocks that have been connected to the **Firstapp** process by a conduit.

メインメニューから **Objects** ▶ **External Connections** を選択します。 **Select Process** ダイアログ・ボックスが、情報伝達ルート (conduit) によって **Firstapp** プロセスに接続した全てのプロセス・ブロックを示してあらわれます。

2. Click on **PIO.Firstappa**, then on **Connect**.

接続上の **PIO.Firstappa** をクリックしてください。



3. Connect the first Terminator by selecting **Val1Term(Sum.In01)** from the **Incomplete Connections** list box, **LIC100** from the **Functions** list box, and **ReadIn** from the **Elements** list-box. Then click on **Connect**.

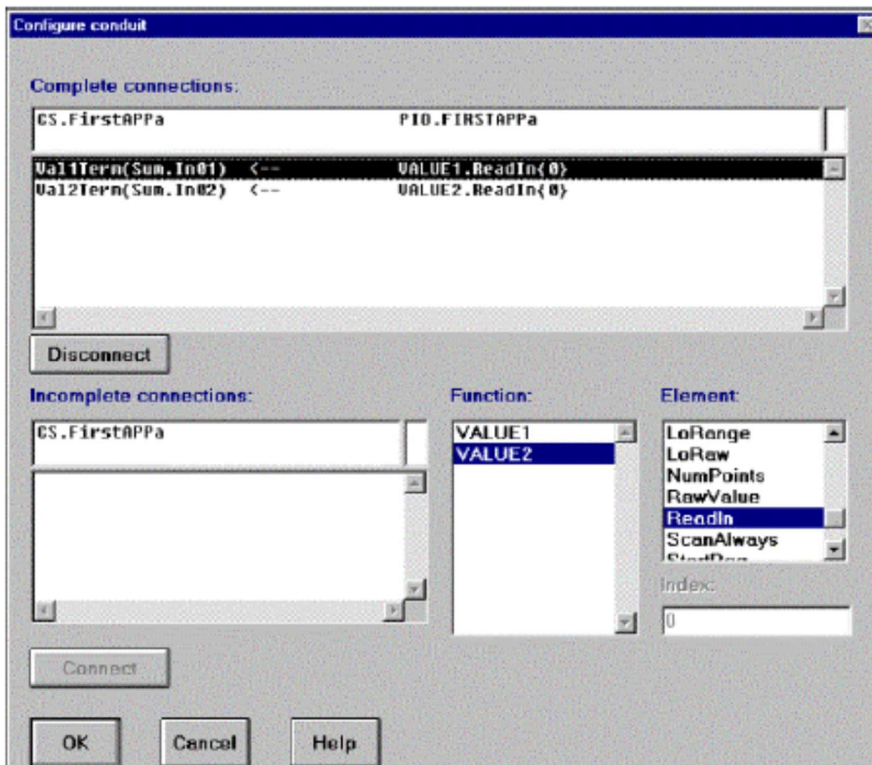
**Incomplete Connections** リストボックスから **Val1Term(Sum.In01)**, **Functions** リストボックスから **LIC100**, そして **Elements** リストボックスから **ReadIn** を選択して、最初のターミネータを接続してください。そして **Connect** をクリックしてください。

4. To connect the second terminator, select **Val2Term (Sum.In02)** from the **Incomplete Connections** list box, **LIC102** from the **Functions** list box, and **ReadIn** from the **Elements** list-box.

2つ目のターミネータを接続するには、**Incomplete Connections** リストボックスから **Val2Term (Sum.In02)**, **Functions** リストボックスから **LIC102**, そして **Elements** リストボックスから **ReadIn** を選択してください。

5. Click on **Connect**, then on **OK**. The connections you just created should appear in the **Complete Connections** list box.

**Connect** それから **OK** をクリックしてください。作成した接続が **Complete Connections** リストにあらわれます。



**Note:** If you want to review your configuration, select the Edit tool, right-click on one of the function blocks, and select Enter Layer. Double-click on either Terminator and the Configure Terminator dialog appears listing the destination and source connections. The GoTo button allows you to move directly to the referenced connection. This helps you view an off-screen connection.

**注意：**設定を再検討するには、Editツールを選択して、1つの関数ブロックを右クリックして、Enter Layerを選択してください。Terminatorをダブルクリックすると、Configure Terminatorダイアログが、destinationとsourceの接続をリストしてあらわれます。GoToボタンで、直接参照した接続に移動します。これは、画面の無い接続を見るのを助けます。

## Saving The Strategy

### 戦略の保存

#### To save your strategy:

戦略を保存するには：

1. From the Main menu select **File&Save**.

メインメニューから**File▶Save**を選択します。

2. Type Examples.CSB into the **File Name** field, then click on **OK**.

**File Name**領域にExamples.CSBをタイプして、次に**OK**をクリックしてください。

3. Close the Continuous Strategy builder by selecting **File&Exit**.

**File▶Exit**を選択して、Continuous Strategyビルダーをクローズしてください。

# Operator Interface

In this section you will edit your Firstapp display from the previous exercise and add the following to it:  
このセクションでは、前の課題からのFirstappディスプレイを編集し、それに次のものを加えます：

A display field that shows the sum of Value 1 and Value 2.  
Value 1とValue 2の合計を示す表示領域

A bar chart that rises and falls according to that sum.  
その合計により上昇および下降する棒グラフ

By doing this, you will create a display that graphically presents the data.  
これを行うことで、グラフィカルにデータを示すディスプレイを作成します。

## Opening Your Display ディスプレイを開始する

**To open your Firstapp display:**  
Firstappディスプレイをオープンするには：



1. From the Application Manager select **Builders&Operator Interface**.  
Application Managerから、**Builders**▶**Operator Interface**を選びます。
2. From the Main menu select **File&Open**. Double-click on **Firstapp.OIL**.  
メイン・メニューから**File**▶**Open**を選択します。**Firstapp.OIL**をダブルクリックしてください。

**Note:** If the Object and Color Palettes are in your way move them by holding the left mouse button down over the title bar, and dragging it to a new location.  
注意：ObjectとColorのパレットが邪魔な場合、タイトル・バーでマウスの左ボタンを押したまま新しい位置へドラッグして移動することができます。

## Placing Objects オブジェクトを設置する

The next step is to place a data display field that will show the summed values.  
次のステップは合計された値を示すデータ表示領域を置くことです。

**To place a data display field that will show the summed values:**

1. Select the **Data Display Tool**: .  
**Data Display Tool:**  を選択します。
2. Move to the Color palette and select **light gray** as the background color and **red** as the foreground.  
Colorパレットに移って、背景色(background color)に**薄い灰色**そして前景(foreground)に**赤**を選択してください。
3. Move the cursor to the right of the Data display labeled **Value 2** and paste the new data display field.  
**Value 2**とラベルされたデータ・ディスプレイの右にカーソルを動かして、新しいデータ表示領域を貼ってください。
4. Select the **Text Tool**: , and set **dark blue** as the foreground color. Click beneath the display and type TOTAL.  
**Text Tool:**  を選択して、前景色に**紺色**をセットしてください。ディスプレイの下をクリックしてTOTALとタイプしてください。


## Creating A Bar Chart

### 棒グラフの作成

#### To create a bar chart:

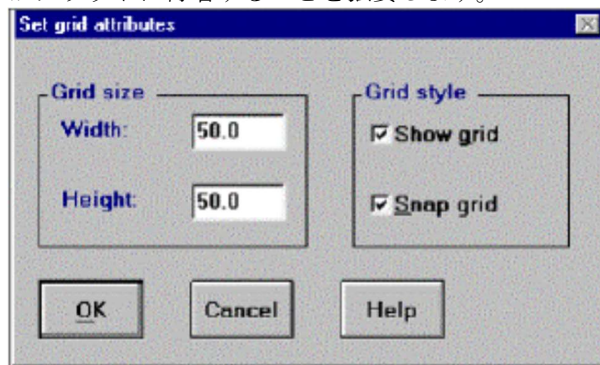
棒グラフを作成するには :

1. Select the **Edit Tool** :

**Edit Tool:**  を選択してください。

2. From the Main menu select **Options** > **Grid**. Select **50** for both the Width and the Height. Click on the **Show Grid** and **Snap Grid** checkboxes, then click on **OK**. Snap Grid forces the lines you draw to adhere to the grid.

メインメニューから**Options** > **Grid**を選択します。Width (幅) とHeight (高さ) の両方で**50**を選択します。**Show Grid**と**Snap Grid**のチェックボックスをクリックして、次に**OK**をクリックしてください。Snap Gridは、描画する線がグリッドに付着することを強要します。



3. To create the bar, select the **Polygon Tool** , then **yellow** as the foreground color.

棒を作成するために、**Polygon Tool:**  を選択して、次に前景色として**黄色**を選択してください。

4. Move to the display window, and use the grid as a guideline. To start the first line segment hold the shift key (to create right angles) and click next to the top of the **Value 2** scroll-box.

ディスプレイ・ウィンドウに移動して、ガイドラインとしてグリッドを使用してください。最初の線分を始めるために、シフトキーを押したまま (直角の作成) 、**Value 2** スクロールボックスの上端の横をクリックしてください。

5. Without holding the mouse button down, move to the right about four grid spaces and click to make a corner.

マウスボタンを押していない状態で、右方向にグリッド・スペースを約4マス移動し、角を作るためにクリックします。

6. Move the mouse down about eight grid spaces and click to make the next corner.

マウスを下方方向にグリッド・スペースを約8マス移動し、次の角を作るためにクリックします。

7. Move the mouse four grid spaces to the left, then double-click. Paragon finishes the polygon, making it a perfect rectangle.

マウスを左方向にグリッド・スペースを約4マス移動し、ダブルクリックします。Paragonはそれを完全な長方形にして、polygonを終了します。

**Note:** If you make a mistake while you are drawing the rectangle, right-click and select Remove Last Vertex. If you need to start over right-click and select Start Over.

**注意:** 長方形を描いていて間違ったら、右クリックしてRemove Last Vertexを選択してください。やり直しが必要ならば、右クリックしてStart Overを選択してください。


## Defining the Attributes 属性(Attribute)を定義する

Next, define the attributes for the data display field and bar. This determines the object's properties. In this case, you are going to configure the Total field to display the summed values of Value 1 and Value 2.

次に、データ表示領域および棒の属性を定義します。これはオブジェクトのプロパティを決定します。今回は、Value 1とValue 2の合計された値を表示するTotal領域を設定します。

**To define the objects attributes:**  
オブジェクトの属性を定義するには：


1. Select the **Edit Tool:** and double-click on the **Total** data display field. Configure its **Edit Object Attributes** dialog box by entering the following values:

**Edit Tool:**  を選択して、**Total**データ表示領域をダブルクリックしてください。以下の値を入力して、**Edit Object Attributes**ダイアログボックスを設定してください：

0 Type 7 for the **Length** and 2 for the **Decimal Places**.

**Length** (桁数) に7, そして**Decimal Places** (小数点以下の桁数) に2をタイプしてください。

0 Click on the **Chevron:** next to the **Connection** field to select the Out parameter of the Add block. This uses the strategy from the CS to sum the values from the Value 1 and Value 2 fields.

**Connection**領域横の**Chevron:**  をクリックして、**Add**ブロックのOutパラメータを選択してください。これは、Value 1とValue 2の領域からの値を合計するためにCSからの戦略を使用します。

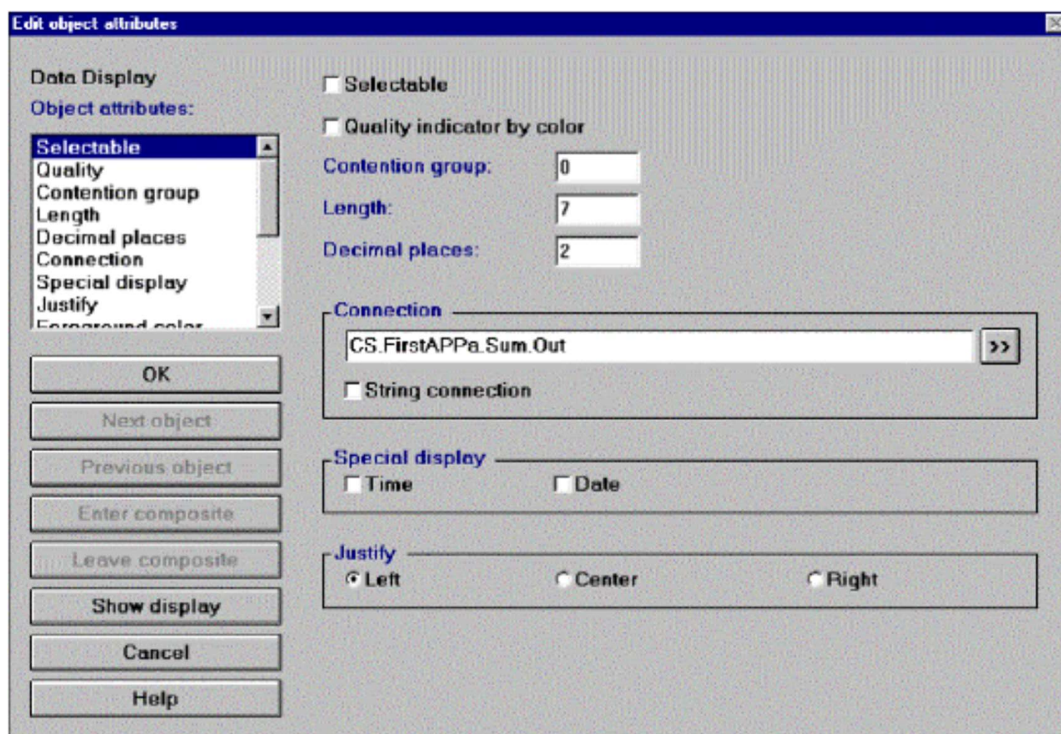
0 Select **Continuous Strategy** from the **Topic** scroll-box, **Examples.CSD** from the **Database** list box. **Firstappa** from the **Process** list box, **Sum** from the **Function** list box, and **Out** from the **Element** list box. This writes out the Total field's value. In the next section you will connect the bar chart to this value so that it rises and falls according to the sum.

**Topic**スクロールボックスから**Continuous Strategy**, **Database**リスト・ボックスから**Examples.CSD**, **Process**リストボックスから**Firstappa**, **Function**リストボックスから**Sum**, そして**Element**リストボックスから**Out**を選択してください。これはTotalフィールドの値を書きます。次のセクションでは、合計に応じてそれが上下するように、棒グラフをこの値に関連づけるでしょう。

2. Click on **OK**, then **OK** again to exit the dialog box.

**OK**そして再度**OK**をクリックしてダイアログボックスを抜けます。






3. Double-click on the bar to open its attributes dialog box. Scroll through the **Object Attributes** list box and click on **Percent Fill**. This defines how the bar will rise.

attributesダイアログ・ボックスを開くために棒をダブルクリックしてください。**Object Attributes**リストボックスをスクロールして、**Percent Fill**をクリックしてください。これは棒がどのように上昇するか定義します。

4. Click on the **Chevron:** next to the **Percent Fill Connection** field and select **Continuous Strategy** from the **Topic** scroll-box, **Examples.CSD** from the **Database** list box, **Firstappa** from the **Process** list box, **Sum** from the **Function** list box, and **Out** from the **Element** list box.

**Percent Fill Connection**領域横の**Chevron:**  をクリックして、**Topic**スクロールボックスから**Continuous Strategy**, **Database**リストボックスから**Examples.CSD**, **Process**リストボックスから**Firstappa**, **Function**リストボックスから**Sum**, **Element**リストボックスから**Out**を選択してください。

**CS.Firstapp.Sum.Out** is the out parameter of the **Add** block.

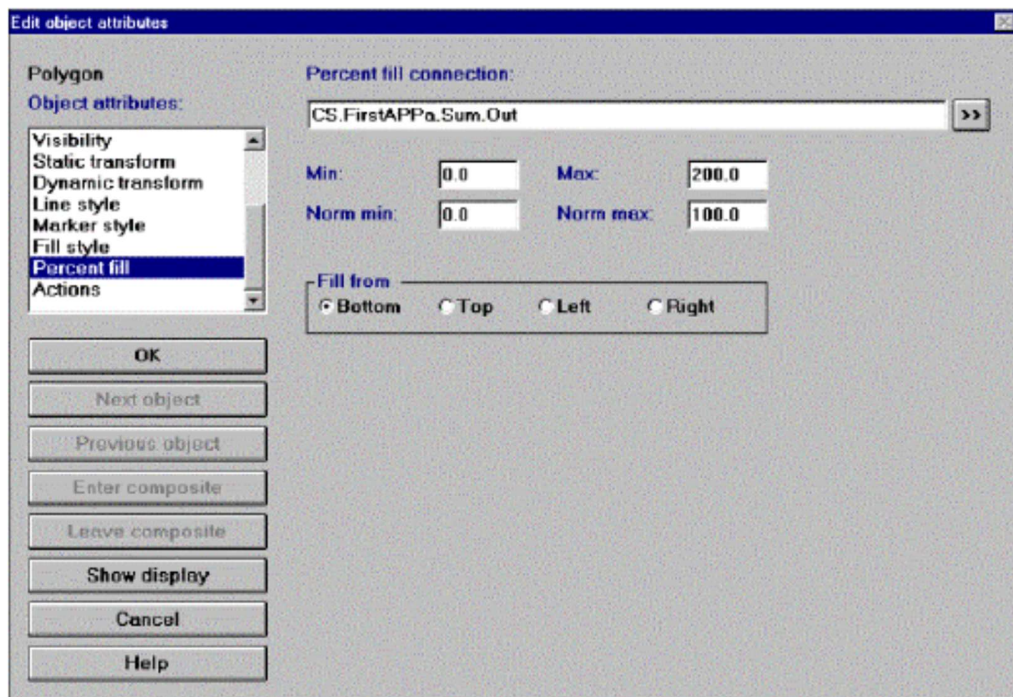
**CS.Firstapp.Sum.Out**は**Add**ブロックのoutパラメーターです。

5. Type 200 into the **MAX** field to correspond to the high range from the **Add** block.

**Add**ブロックからの高い範囲に対応させるために、**MAX**領域に200をタイプしてください。

6. Be sure that the **Bottom** radio button is selected in the **Fill From** field, so the bar will rise from the bottom. Click on **OK**.

棒が下部から上昇するように**Bottom**ラジオボタンがFill From領域で選択されるのを確認してください。**OK**をクリックします。



## Assigning Actions For Display Call-ups

### ディスプレイ呼び出しのための動作を割り当てる

The final step is to create two push buttons. One will call up the display you are going to build in the next chapter, and the other closes this one.

最終ステップは2つのプッシュボタンを作成することです。1つは次章で構築するディスプレイを呼び出し、もう1つはそれを閉じます。

#### To create two new push buttons:

2つの新しいプッシュボタンを作成するには：

1. Select the **Push Button Tool** :

**Push Button Tool:**  を選択します。

2. Select **green** as the background color and **black** as the foreground color.

背景色に**緑**そして前景色に**黒**を選択してください。

3. Paste a push button on the bottom-left corner of the display window. Select **purple** as the background color and paste another push button on the bottom-right corner of the display window.

プッシュボタンをディスプレイ・ウィンドウの左下の角に貼ってください。背景色に**紫**を選択して、別のプッシュボタンをディスプレイ・ウィンドウの右下の角に貼ってください。

4. Double-click on the **green** push button to bring up its **Edit Object Attributes** dialog box.

**緑**のプッシュボタンをダブルクリックして、**Edit Object Attributes**ダイアログボックスを出してください。

5. Select **Label** from the **Object Attributes** list box, then type TANK in the **Label** field. This is the description that will appear on the Tank button.

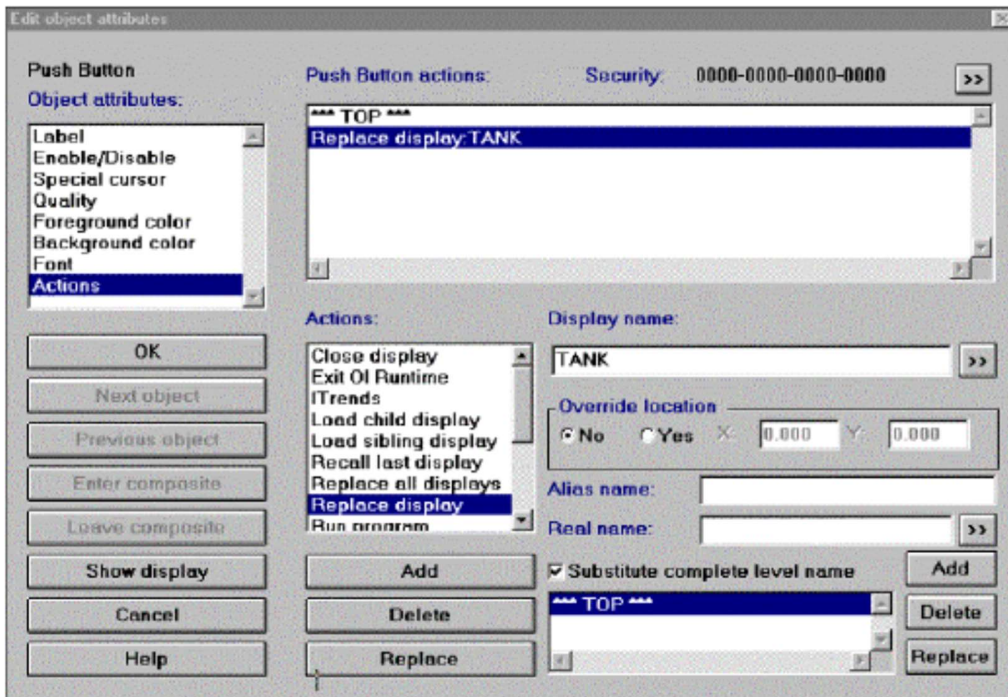
**Object Attributes**リストボックスから**Label**を選択して、次に**Label**領域へTANKとタイプしてください。これはTankボタンに表示される記述です。

6. In the **Objects Attributes** list box, click on **Actions**. In the **Actions** list box click on **Replace Display**. The Actions

attribute specifies what the display will do.

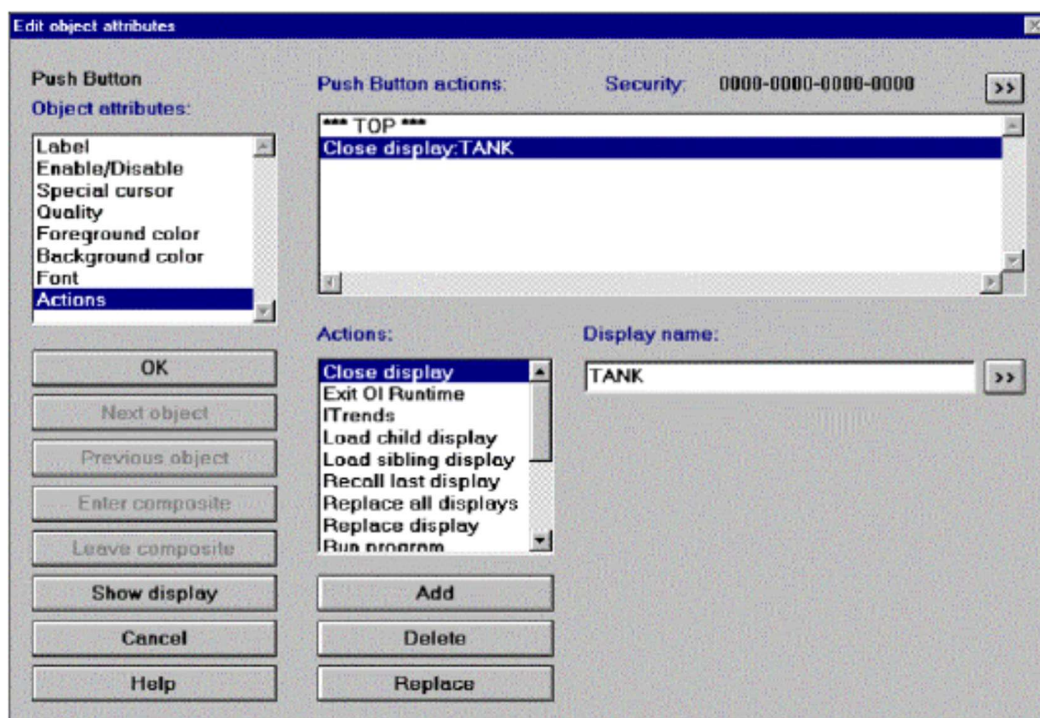
**Objects Attributes** リストボックスで、**Actions** をクリックしてください。 **Actions** リストボックスで **Replace Display** をクリックしてください。アクション属性は、ディスプレイが何を行うか指定します。

7. Type **TANK** into the **Display Name** field, then click on **Add**. The **Push Button Action** list should read "Replace display: Tank." Click on **OK**. During Runtime this will load the Tank display you are going to build in the next chapter. **Display Name** 領域に **TANK** とタイプして、次に **Add** をクリックしてください。 **Push Button Action** リストに "Replace display: Tank" が読める筈です。 **OK** をクリックします。これはランタイムにおいて、次章で構築する Tank ディスプレイをロードします。



8. Double-click on the **purple** button and type **CLOSE** in the **Label** field.  
紫のボタンをダブルクリックして、**Label** 領域へ **CLOSE** とタイプしてください。

9. Click on **Actions** in the **Object Attributes** list box. In the **Actions** list box select **Close Display**, then click on the **Add** button. When this button is selected during Runtime it will close the display. **Objects Attributes** リストボックスで、**Actions** をクリックしてください。 **Actions** リストボックスで **Close Display** を選択し、**Add** ボタンをクリックしてください。このボタンがランタイムにおいて選択される時、それはディスプレイを閉じるでしょう。



10. No other parameters are needed. Click on **OK**, then save your application.

他のパラメーターは必要ではありません。**OK** をクリックして、次にアプリケーションを保存してください。

## Application Manager

**Start-up File:** defines which processes execute during Runtime. It also specifies the station name, communication's port, and scan period.

**Start-up File:** プロセスがランタイムにおいて何を実行するか定義します。さらに、ステーション名、通信ポート、およびスキャン周期を指定します。

You must edit your Examples application file (Examples.APP) to add the start-up file for the Continuous Strategy. The **start-up file** is the Examples.CSC configuration file you created earlier in this chapter.

Continuous Strategyのスタートアップ・ファイルを加えるために、Examplesアプリケーション・ファイル (Examples.APP)の編集が必要です。スタートアップ・ファイルは本章の初めに作成したExamples.CSC設定ファイルです。

### To edit your Examples application file:

**Examplesアプリケーション・ファイルを編集するには：**

1. If the Control Panel is still open, click on it to bring it to the foreground. If not, you need to open your Examples application.

コントロールパネルがまだ開いている場合は、それを前面に出すためクリックしてください。そうでなければ、Examplesアプリケーションをオープンする必要があります。

2. From the Application Manager Main menu select **File**►**Open** then double-click on Examples.APP.

Application Managerのメインメニューから**File**►**Open**を選択して、Examples.APPをダブルクリックします。

**Note:** The start-up files you designated in the previous exercise for the Process I/O and the Operator Interface should be listed. You must now add the Continuous Strategy start-up file.

**注意：** 前の課題でProcess I/OおよびOperator Interfaceとして指定したスタートアップ・ファイルがリストされるべきです。いまContinuous Strategyのスタートアップ・ファイルを追加する必要があります。

3. Double-click on the Continuous Strategy line to designate its start-up file.

そのスタートアップ・ファイルを指定するためにContinuous Strategyラインをダブルクリックしてください。

4. Click on the Subsystem In Application Group field, then click on the Chevron: next to the Start-Up File field. Double-click on Examples.CSC, then click on **OK**.

Application Group領域でSubsystemをクリックして、次にStart-Up File領域の横のChevron:  をクリックしてください。Examples.CSCをダブルクリックして、**OK**をクリックしてください。

5. From the Application Manager Main menu select **File** > **Save**. The Operator Transcript window tells you that Examples.APP has been saved. You are ready to enter Runtime.

Application Managerのメインメニューから**File** > **Save**を選択します。Operator Transcript（操作記録）ウィンドウは、Examples.APPが保存されたことを伝えます。あなたはランタイムに入る準備ができました。

## Runtime

### ランタイム

#### To enter Runtime:

ランタイムに入るには：

1. Click on the Control Panel to bring it to the foreground.

コントロール・パネルをクリックして前面に出します。

2. If you shut down the Process I/O in the last exercise No is listed in the Running column. Select this line, then click on **Start Selected**.

シャットダウンした最後の実行番号のProcess I/OはRunning欄にリストされています。ラインを選択して、**Start Selected**をクリックしてください。

3. Double-click on the Continuous Strategy line, select the **Subsystem In Application Group checkbox**, then click on **OK**. Do the same thing for the Operator Interface line.

Continuous Strategyラインをダブルクリックして、**Subsystem In Application Group checkbox**を選択し、次に**OK**をクリックする。Operator Interfaceラインのために同じことを行ってください。

4. Click on **Start Group**, and your display will appear.

**Start Group**をクリックすると、あなたのディスプレイがあらわれます。

5. Enter a value into the Value 1 field. The Total field reflects this change and the bar display moves accordingly.

Value 1領域に値を入力してください。Total領域はこの変更を反映し、そして棒の表示は連動して動きます。

6. Select a new value for the **Value 2** field by double-clicking on an entry in the list box. The **Total** field and **Bar** change accordingly. Continue to experiment with different values.

リストボックスのエントリをダブルクリックして、**Value 2**領域に新しい値を選択してください。**Total**領域と棒は連動して変わります。異なる値で実験し続けてください。

The reason this happens is because you configured the Value 1 Data Display field to read the information that was written out from the Value 1 Data Entry field (PIO.Firstappa.LIC100.WriteOut). You then configured the Data Display field to read this information (PIO.Firstappa.LIC100.ReadIn).

これが起こる理由は、Value 1データ入力領域へ書き込まれた(PIO.Firstappa.LIC100.WriteOut)情報を読み取るようにValue 1データ表示領域を設定したからです。それから、この情報を読むためにデータ表示領域を設定しました(PIO.Firstappa.LIC100.ReadIn)。

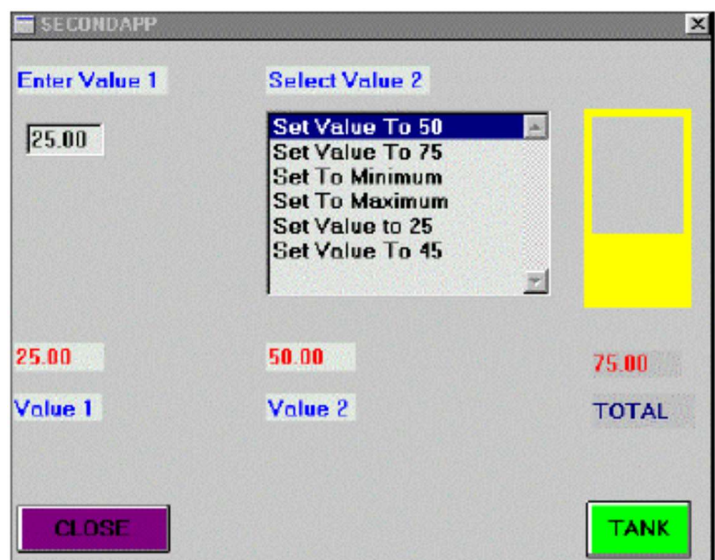
The Value 2 Data Display field was configured the same way except it reads the information from the Select Value 2 list box. The list box writes out the values that you associated with each entry.

Value 2データ表示領域は、Value 2リストボックスで選択した情報を読み取ることを除いて同じように設定されてい

ます。リストボックスは、各エントリに関連した値を書き込みます。

Using the strategy you created, the CS adds the values from the Value 1 and Value 2 fields and displays the sum in the Total field. The total field writes out its value to the bar chart, which rises and falls accordingly. This was done by assigning the percent fill attribute.

作成した戦略を使用して、CSはValue1とValue2の領域からの値を合算して、Total領域に合計を表示します。Total領域は値を棒グラフ（それは連動して上下します）に書き出します。これはpercent fill（パーセント充填）属性を割り当てることにより行われました。



## Closing the Display

### ディスプレイをクローズする

1. When you are finished, close the display by clicking on the **Close** button.

終了するには、**Close**ボタンをクリックしてディスプレイを閉じてください。

2. In the next exercise you will edit the configuration file; therefore, you must shut down the Continuous Strategy. Click on the **Continuous Strategy** line in the Control Panel, then click on **Stop Selected**.

次の課題では、設定ファイルを編集します；そのために、**Continuous Strategy**をシャット・ダウンする必要があります。コントロールパネルの**Continuous Strategy**ラインをクリックして、次に**Stop Selected**をクリックしてください。

## Chapter Summary

### 章のまとめ

Congratulations! You now know how to use the Continuous Strategy to create, configure and connect function blocks to perform basic calculations. In the next chapter you will create a display that simulates a tank filling.

おめでとう！いまあなたは基本的な計算を行なう関数ブロックを作成し、設定し、かつ接続するために**Continuous Strategy**（連続的な戦略）を使用する方法を知っています。次の章では、タンク充填物をシミュレートするディスプレイを作成します。

# Tank Fill Simulation

## タンク充填物のシミュレーション

### In This Chapter:

この章では：

• **Continuous Strategy 99**

• **Operator Interface 105**

• **Creating A PID Faceplate 114**

• **Display Call-up 116**

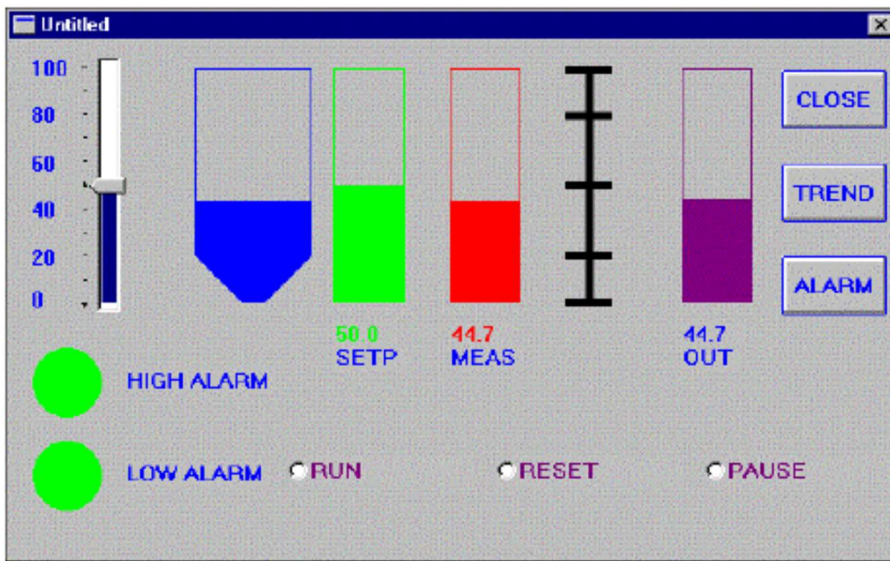
• **Runtime 118**

• **Chapter Summary 119**



In this exercise you will simulate the flow in and out of a tank. This is done by connecting a PID (Proportional Integral Derivative) block to a SIM (Simulation) block in the Continuous Strategy. You will design a tank in the Operator Interface and use a slider device to manipulate its setpoint. The tank will include digital animation to indicate alarms, and radio buttons to start, pause, or reset the process. When you are finished your display should look like the following:

この課題では、タンクの流入と流出をシミュレートします。これはContinuous StrategyのSIM(Simulation)ブロックにPID(Proportional Integral Derivative)ブロックを接続することにより行われます。Operator Interfaceでタンクを設計して、セットポイントの操作にスライダー・デバイスを使用します。タンクは、アラームを示すデジタル・アニメーション、そしてプロセスの開始(start)、停止(pause)、もしくはリセット(reset)のためのラジオボタンを持ちます。完了するとディスプレイは次のように見える筈です。



# Continuous Strategy

## 連続する戦略(Continuous Strategy)

For this exercise you will add another process block to the **Firstapp** strategy. This process block will contain a PID block connected to a SIM block to simulate the flow in and out of a Tank.

この課題では、**Firstapp** 戦略(strategy)に別のプロセス・ブロックを加えます。このprocessブロックは、タンクおよびタンクの流入と流出をシミュレートするためにSIMブロックに接続されたPIDブロックを含みます。

### To add a PID block to your strategy:


戦略にPIDブロックを追加するには：

1. Enter the Continuous Strategy builder. Open the **Firstapp** strategy by selecting **File**➤**Open** from the Main menu. Double-click on **Examples.CSB**.

Continuous Strategy builderを入力します。メイン・メニューから**File**➤**Open**を選択して**Firstapp**戦略をオープンします。**Examples.CSB**をダブルクリックしてください。

2. Click on the **Function Block Tool:** , then paste the block beneath the **Firstapp** process.

**Function Block Tool:**  をクリックして、次に**Firstapp**プロセスの下にブロックを貼ってください。

3. Select the **Edit Tool:**  , then double-click on the block to open its **Configure Function Block** dialog box.

**Edit Tool:** を選択して、次に、その**Configure Function Block**ダイアログ・ボックスを開くためにブロックをダブルクリックしてください。

4. Type Tank in the **Name** field. Make sure that the Continuous Strategy is selected as the **Type Topic**, then select the **Expand** button.

**Name**領域にTankとタイプしてください。Continuous Strategyが**Type Topic**として選択されていることを確かめて、次に**Expand**ボタンを選択してください。

**Note:** Unlike the previous exercise the Tank process will not receive any values from the Process I/O. You do not have to connect it to another process block.

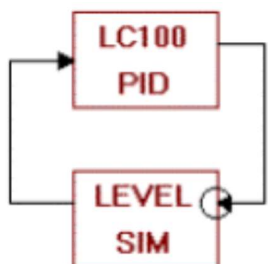
**注意：**前の課題と異なり、TankプロセスはProcess I/Oから値を受け取りません。別のプロセス・ブロックにそれを接続する必要はありません。

## Defining Function Blocks

### 関数ブロックを定義する

Next, the PID and SIM blocks have to be placed on the display screen. The PID block is the most sophisticated linear control block in Paragon. The PID block connects to the SIM block to emulate a process without physically connecting to the hardware. The following figure shows an example of the completed tank process.


次に、ディスプレイ・スクリーンにPIDとSIMのブロックを置かなければなりません。PIDブロックはParagonで最も洗練された線形の制御ブロック(linear control block)です。PIDブロックは、ハードウェアに物理的に接続しないでプロセスをエミュレートするためにSIMブロックに接続します。次の図は、完成したTankプロセスの例を示します。




### To place the blocks on your display:

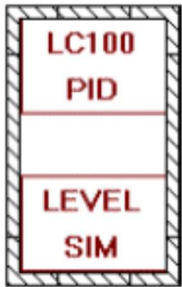
ディスプレイにブロックを設置するには：

1. Click on the **Function Block Tool:** , then paste a block in the middle of your screen. Paste another beneath the first.

**Function Block Tool:**  をクリックして、次にスクリーンの中央にブロックを貼ってください。そして1つ目の下にもう1つブロックを貼ってください。

2. Click on the **Edit Tool:** , then select both blocks by positioning the mouse outside their perimeter, holding the mouse button down, and dragging it until the frame surrounds both blocks. Release the mouse button.

**Edit Tool:**  をクリックして、そして次に、マウスを両方のブロック周辺から外れたところに置いて、マウスボタンを押したまま、ドラッグして両方のブロックを囲んで選択します。マウスボタンを離してください。



3. Double-click on the selection to open its Configure Function Block dialog. Here you set the objects properties.

選択をダブルクリックして、Configure Function Blockダイアログを開きます。ここでオブジェクトのプロパティ（特性）を設定します。

4. Configure the dialog box by entering the following values:

次の値を入力して、ダイアログ・ボックスを設定してください：

0 In the **Name** field type LC100.

**Name**領域にLC100をタイプする

0 Click on the **Chevron:** next to **Type**. Click on **Supervisory Control** in the **Category** list box, then click on **PID** in the **Basic Type** list box.

**Type**横の**Chevron:**  をクリックしてください。**Category**リストボックスで**Supervisory Control**をクリックして、次に**Basic Type**リストボックスの**PID**をクリックしてください。

0 Click on **OK**.

**OK**をクリックします。

5. Add the following values to the specified fields:

指定の領域に次の値を加えてください：

0 **Description:** Level Controller

0 **Scan Rate:** 0.25

0 **Proportional:** 90

0 **Integral:** 0.12

0 **Derivative:** 0.01

Configures function block

Name:  >> Type:  >>

Description:  Scan rate (Sec):

Scaling

High range:  High limit:   Ext

Low range:  Low limit:   Ext

Engineering unit:

Inputs

Feedback:  Y  N Remote setpoint:  Y  N

Algorithm control

Proportional (Pct):   Ext Setpoint:

Integral (Min):   Ext Initial output:

Derivative (Min):   Ext Inc/Inc:  Y  N  Ext

Bias:   Ext Track:  Y  N  Ext

KBias:  Manual:  Y  N

Page 1 of 3 Proportional Integral Derivative Controller

OK Next Previous Show Expand Cancel Help

6. Click on the right arrow on the bottom left of the dialog box to move to the second page. Add the following values to the specified fields:

2ページ目に移るために、ダイアログ・ボックスの左下の右向き矢印をクリックしてください。指定の領域に次の値を加えてください:

0 Inhibit: N

0 Priority: 1

0 High Alarm: 75.00

0 Low Alarm: 25.00

0 Alarm Deadband: 3.00

Configure function block

Name: LC100 Type: PID

Description: Level Controller Scan rate (Sec): 0.25

Alarming

Inhibit:  Y  N  Ext Priority: 1

High alarm: 75.00  Ext Deviation alarm: 100.00  Ext

Low alarm: 25.00  Ext Deviation deadband: 1.00

Alarm deadband: 3.00

Group:  1  2  3  4  5  6  7  8


Page 2 of 3 Proportional Integral Derivative Controller

OK Next Previous Show Expand Cancel Help

7. Click on the **Next** button and configure it by doing the following:  
**Next**ボタンをクリックして、次のことを行って設定してください：

0 Type Level in the **Name** field.  
**Name**領域にLevelとタイプする

0 Click on the **Chevron**: next to the **Type** field. Click on **Supervisory Control** in the category list box, then click on **SIM** in the **Basic Type** list box.

**Type**領域の横の**Chevron**:  をクリックしてください。categoryリストボックスで**Supervisory Control**をクリックして、次に**Basic Type**リストボックスで**SIM**をクリックしてください。

8. Click on **OK**.  
**OK**をクリックします。

9. Add the following values to the specified fields:  
指定の領域に次の値を加えてください：

0 **Description**: Tank Level Simulation

0 **Scan rate**: 0.25

0 **High limit**: 95.00

0 **Low limit**: 5.00

0 **Lag1**: 0.02

0 **Lag2**: 0.02

10. Click on **OK**.  
**OK**をクリックします。

**Configure function block**

Name:  >> Type:  >>

Description:  Scan rate (Sec):

**Scaling**

High range:  High limit:   Ext

Low range:  Low limit:   Ext

Engineering unit:

**Algorithm control**

Lag1:   Ext Initial output:

Lag2:  Integrate:  Y  N

Bias:   Ext Noise on:  Y  N

KBias:  Noise max:

Inc/Inc:  Y  N  Ext Manual:  Y  N

Page 1 of 3 Simulation

OK Next Previous Show Expand Cancel Help

## Connecting Function Blocks

### 関数ブロックを接続する

The next step is to connect the output of the PID block to the SIM block, and the output of the SIM block to the PID block.

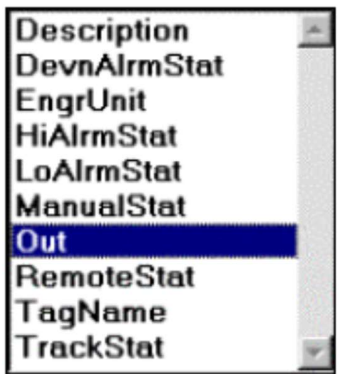
次のステップはSIMブロックへのPIDブロックの出力、およびPIDブロックへのSIMブロックの出力を接続することです。

#### To connect the output of the PID block to the SIM block:

SIMブロックにPIDブロックの出力を接続するには：

1. Select the **Connection Tool** , then click on the PID block to produce the **Output Connection Elements** list box. Click on **Out**.

**Connection Tool**:  を選択して、次に**Output Connection Elements** リストボックスを作成するためにPIDブロックをクリックしてください。 **Out** をクリックしてください。



2. Draw the connection:

接続を描きます：

0 Move the mouse to the right of the PID block and click to make a corner.

PIDブロックの右にマウスを移動させて、角を作るためにクリックしてください。

0 Move the mouse down until it is level with the SIM block. Click to make a corner, then bring the connection into the SIM block.

それがSIMブロックを含む段になるまで、マウスを下へ移動させてください。角を作るためにクリックして、次にSIMブロックへ接続を持って来ててください。

0 Click on the SIM block to open the **Input Connection Elements** list box.

**Input Connection Elements** リストボックスを開始するためにSIMブロックをクリックしてください。

3. Click on **In**.

**In** をクリックします。



4. With the Connection tool still active, click on the SIM block to produce its **Output Connection Elements** list box. Select **Out**, and connect it to the PID block.

Connection toolがまだアクティブな状態で、SIMブロックをクリックして、**Output Connection Elements** リストボックスを作り出してください。 **Out** を選択して、それをPIDブロックに関連づけてください。

5. Select **In** from the **Input Connection Elements** list box. The following message appears:

**Input Connection Elements** リストボックスから **In** を選択してください。次のメッセージがあらわれます：

**Loop detected. Left click on loop Master block or press ESC to abort the connection.**

(ループを検知しました。ループのマスター・ブロックを左クリックするか、**ESC**を押して接続を中止できます)

During Runtime, Paragon automatically processes blocks in order of signal flow. However, if you construct an internal loop into your control strategy, Paragon cannot decide which block should execute first. You must specify which block within the loop executes first.

ランタイムにおいて、Paragonはシグナル・フローの順に自動的にプロセス・ブロックを行います。しかしながら、制御戦略(control strategy)に内部ループを構築すると、Paragonはどのブロックを最初に実行するか決定することができません。ループ内のどのブロックを最初に実行するか明示する必要があります。

6. Click on **OK**. Move the cursor to the SIM block and click on it. The following message appears:

**OK**をクリックします。SIMブロックにカーソルを移動して、それをクリックしてください。次のメッセージがあらわれます：

**Warning. Quality propagation for the looped connection is disabled.**

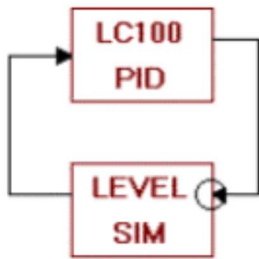
(警告。ループ状の接続のための性質:Qualityの伝播は無効になります)

Quality tags prevent control blocks from executing until a valid signal is input. If an input value has a quality tag of unavailable, then it is ignored by the control block. Therefore, the control algorithm will not execute. This ensures that bad values do not affect the internal history parameters maintained within the control block.

Qualityタグは、有効な信号が入力されるまで、コントロール・ブロックが実行するのを防ぎます。入力値を利用できないQualityタグがあると、それはコントロール・ブロックによって無視されます。したがって、コントロール・アルゴリズムは実行されません。これは、悪い値がコントロール・ブロック内に保持された内部履歴(internal history)パラメータに影響しないことを保証します。

7. Click on **OK**. Paragon marks the SIM input with a circle.

**OK**をクリックします。ParagonはSIM入力に円マークをつけます。



8. Return to the Process layer by selecting **View>Process Layer**.

**View>Process Layer**の選択によりプロセス・レイヤーに戻ってください。

## Saving Your Strategy

### 戦略を保存する

**To save the changes to your strategy.**

戦略の変更を保存するには：

1. From the Main menu select **File>Save**.

メイン・メニューから、**File>Save**を選択します。

2. Close, move or minimize the Continuous Strategy builder.

Continuous Strategy builderを閉じるか、移動させるか、最小化してください。



# Operator Interface

The next step is to create the Tank display. The display will include a slider, 3 radio buttons, a PID faceplate, and digital animation. First you must define the display area.

次のステップはTankディスプレイを作成することです。ディスプレイは、スライダー、3つのラジオボタン、PIDフェースプレート、そしてデジタル・アニメーションを持ちます。最初に、ディスプレイ・エリアを定義する必要があります。

## To define the display area:

ディスプレイ・エリアを定義するには：

1. Enter the Operator Interface builder and select **File** **▶** **New**. Configure the window by entering the following values: Operator Interface builderに入り、**File** **▶** **New**を選択してください。次の値を入力してウィンドウを設定してください：

0 In the **Horizontal** field type 0.4250

**Horizontal**（水平）領域に0.4250をタイプする

0 In the **Vertical** field type 0.0950.

**Vertical**（垂直）領域に0.0950をタイプする


0 In the **Width** field type 0.8500.

**Width**（幅）領域に0.8500をタイプする

0 In the **Height** field type 0.6000.

**Height**（高さ）領域に0.6000をタイプする

0 Click on the **Chevron:** next to **Background Color** and select **light gray**. Click on **OK**.

**Background Color**（背景色）の横の**Chevron:**  をクリックして、**light gray**（薄い灰色）を選択してください。**OK** をクリックしてください。

0 Under **Frame Style** click on **Double**.

**Frame Style**で、**Double**をクリックしてください。

2. Click on **OK**.

**OK**をクリックしてください。

3. Activate the grid by selecting **Options** **▶** **Grid**. Under grid size, set the Width and Height to 50. Under grid style select **Show Grid** and **Snap Grid**. Click on **OK**.

**Options** **▶** **Grid**の選択によりグリッドを活性化してください。grid sizeで、WidthとHeightに50をセットしてください。grid styleで、**Show Grid**と**Snap Grid**を選択します。**OK**をクリックしてください。

## Designing The Tank

### タンクをデザインする

The next step is to draw the tank. Your tank can be any shape. Paragon is capable of filling irregular shapes.

次のステップはタンクを描くことです。タンクはどんな形でもあり得ます。Paragonは充填している不恰好な形にすることもできます。

## To draw the tank:

タンクを描くには：


1. Select the **Polygon Tool:** and set **blue** as the foreground color.

**Polygon Tool:**  を選択して、前景色(foreground color)に**青(blue)**を設定してください。


2. Click in the upper-left corner of the screen and draw the tank in any shape you desire. Keep the size around eight grid spaces tall and 6 grid spaces wide. Remember to click whenever you want to make a corner and double-click when you have finished the object.

スクリーンの左上の角をクリックして、あなたの望む任意の形でタンクを描いてください。6つのグリッド・スペースの幅と8つのグリッド・スペースの高さのサイズを保持してください。角を作りたいときはいつでもクリックして、オブジェクトを終了するときはダブルクリックすることを覚えておいてください。

3. Select the **Edit Tool:** and double-click on the **Tank** to open its **Object Attributes** dialog box.

**Edit Tool:**  を選択して、**Object Attributes** ダイアログボックスを開くためにタンク(**Tank**)をダブルクリックしてください。

4. From the **Object Attributes** list box select **Percent Fill**. Click on the **Chevron:** next to the **Percent Fill Connection** field. Select **Continuous Strategy** from the **Topic** scroll-box, **Examples.CSD** from the **Database** list box, **Tank** from the **Process** list box, **LC100** from the **Function** list box, and **In** from the **Element** list box. This will read the information from the strategy you just created.

**Object Attributes** リストボックスから、**Percent Fill** を選択します。**Percent Fill Connection** 領域の横の **Chevron:**  をクリックしてください。**Topic** スクロールボックスから **Continuous Strategy**, **Database** リスト・ボックスから **Examples.CSD**, **Process** リスト・ボックスから **Tank**, **Function** リスト・ボックスから **LC100**, および **Element** リスト・ボックスから **In** を選択してください。これは、ちょうど作成した戦略から情報を読みます。

**Note:** If you click on Continuous Strategy in the Topic field and the Examples.CSD file does not appear, click on Browse. When the Select Library window appears double-click on the two periods (..) in the directories list box. Then select Firstapp. Examples.CSD should appear in the Files list box. Select it, then click on OK.

**注意:** Topic 領域で Continuous Strategy をクリックしても、Examples.CSD のファイルがあらわれなければ、Browse をクリックしてください。Select Library ウィンドウがあらわれたら directories リストボックス内の2つのピリオド(.)をダブルクリックしてください。次に、Firstapp を選択してください。Examples.CSD がファイル・リストボックスにあらわれるに違いありません。それを選択して、次にOKをクリックしてください。

## Adding A Slider To Your Display

### ディスプレイにスライダを追加する


The next step is to add a slider to your display that will change the Tank's setpoint. If you are using Windows NT the slider must be horizontal.

次のステップはタンクのセットポイントを変更するスライダをディスプレイに追加することです。あなたが Windows NT を使用しているなら、スライダは水平であるに違いありません。

#### To add a slider:

スライダを追加するには:

1. Select the **Slider Tool:** and place it beneath the tank.

**Slider Tool:**  を選択して、タンクの下にそれを置いてください。

2. Double-click on it to open its **Object Attributes** dialog box. Configure it by entering the following values:

それをダブルクリックして **Object Attributes** ダイアログボックスを開いてください。次の値を入力して、それを設定してください:

0 From the **Object Attributes** list box select **Type**, then click on the **Vertical** radio button. This produces a vertical slider.

**Object Attributes** リストボックスから **Type** を選択し、次に **Vertical** ラジオボタンをクリックします。これは垂直スライダを作成します。

0 From the **Connection** Field click on the **Chevron:** and select **CS.Tank.LC100.Setp**. Click on **OK**.

**Connection**領域から**Chevron:**をクリックして、**CS.Tank.LC100.Setp**を選択してください。**OK**をクリックしてください。

0 From the **Object Attributes** list box select **Format**.

**Object Attributes** リストボックスから**Format**を選択します。

0 Under **Layout**, set **Tick Intervals** to 10 and **Value Intervals** to 5. Click on **OK**. The Value Intervals option will not appear in Windows NT and will be the same as the Tick Interval's value.

Layoutでは、**Tick Intervals**に10をセットして、**Value Intervals**に5をセットしてください。**OK**をクリックしてください。Value IntervalsオプションはWindows NTではあられず、Tick Intervalの値と同じになります。

0 Under slide direction select **Bottom To Top**. This numbers the slider from the bottom to the top.

Slide direction (スライド方向)では**Bottom To Top**を選択します。これは下部から上部へとスライダーに番号を付けます。

0 Under scale position select **Left**. This places the scale number to the left of the slider.

scale positionでは**Left**を選択します。これはスライダーの左側に目盛りの番号を置きます。

0 Click on the **Chevron:** next to Text Color and set it to **Blue**.

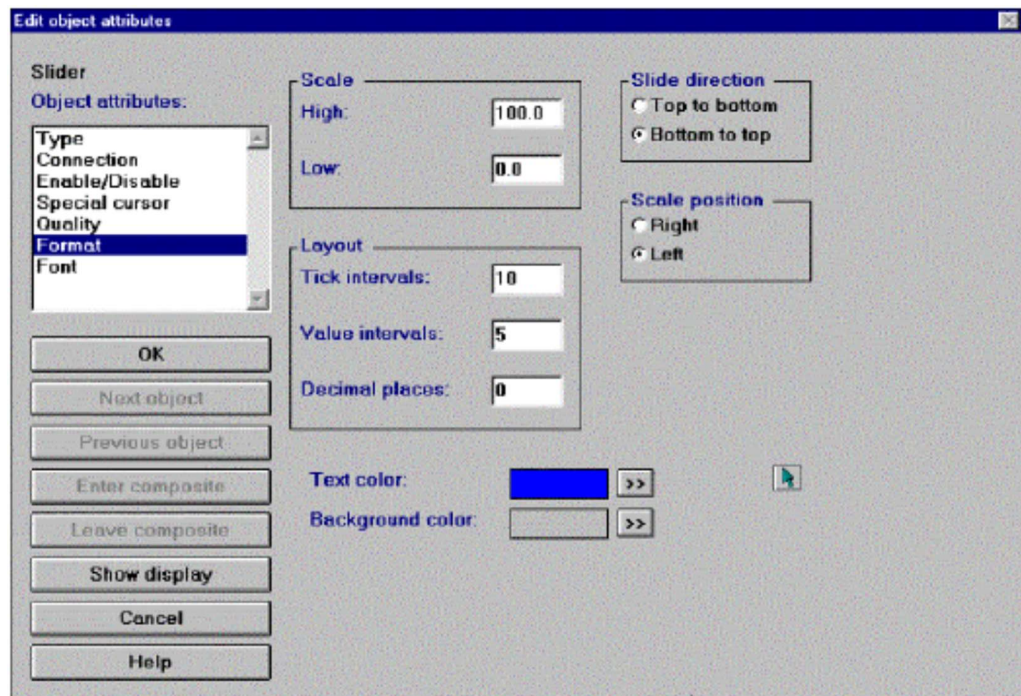
Text Colorの**Chevron:**  をクリックして、**青(Blue)**をセットしてください。

0 Set the background color to **Gray**.

Background color (背景色)に**灰色(Gray)**をセットしてください。

**Note:** The options under Scale Position, and the Value Intervals field will not appear on Windows NT.

**注意:** Scale PositionのオプションおよびValue Intervals領域はWindows NTではあられません。




3. Click on **OK**.

**OK**をクリックしてください。

4. The slider may appear flattened since it originally was in a horizontal position. Select the slider with the **Edit Tool:**

and resize it so it is the same height as the tank. Make it wide enough so you can read its values. Place it to the left of the tank.

スライダーは、もとは水平位置にあったので平らになったように見えるかもしれません。 **Edit Tool:**  でスライダーを選択して、タンクと同じ高さになるようにサイズ変更してください。その値を読み取れるように、それを十分な幅にしてください。タンクの左側にそれを置いてください。

## Digital Animation

### デジタル・アニメーション

**Digital:** is a binary value equal to 1 or 0. Examples include high/low, on/off, true/false, yes/no. Digital values are often referred to as “discrete values.”

**Digital:** デジタルは1または0に等しい2進法の値です。例はhigh/low, on/off, true/false, yes/noを含んでいます。デジタル値はしばしば“discrete values”と呼ばれます。


In this section you will place two circles that represent the high and low alarms. When the block goes into alarm, the circles and their text labels will blink and change color. This is done using **digital** animation. Digital refers to a binary value equal to 1 or 0. Examples include high/low, on/off, true/false, and yes/no. Digital values are often referred to as “discrete values.”

このセクションでは、上限と下限のアラームを表わす2つの円を置きます。ブロックがアラームに入る時、円およびそれらのテキスト・ラベルは明滅し、色を変更するでしょう。これはデジタル・アニメーション(**digital animation**)を使用して完成しています。デジタルは1または0に等しい2進法の値を参照します。例はhigh/low, on/off, true/false, yes/noを含んでいます。デジタル値はしばしば“discrete values”と呼ばれます。

#### To create the high and low alarm circles:

上限と下限の警報 (アラーム) を行う円を作成するには :


1. Select the **Ellipse Tool:** and set **green** as the foreground color.

**Ellipse Tool:**  を選択して、foreground color (前景色) に**緑**をセットしてください。

2. Click beneath the slider to start the circle. Hold the shift key and slide the mouse down and to the right until it is about a half inch in diameter. Click again when finished.

円を始めるためにスライダーの下をクリックしてください。シフトキーを押したまま、円が直径0.5インチ位になるまでマウスを右下に滑らせてください。終了するために、もう1度クリックしてください。

3. Click on the **Edit Tool:** and select the circle. From the Main menu select **Edit&Copy**.

**Edit Tool:**  をクリックして、円を選択してください。メイン・メニューから**Edit&Copy**を選択します。

4. From the Main menu select **Edit&Paste**. Move the mouse cursor beneath the first one, then click to paste it.

メイン・メニューから**Edit&Paste**を選択します。最初の円の下にマウスカーソルを動かし、クリックして貼ってください。

5. Select the **Edit Tool:** and open the first circle's attributes dialog box. Configure the window by entering the following values:

**Edit Tool:**  を選択して、1つ目の円のAttributesダイアログボックスを開いてください。


0 From the Object Attributes list box select foreground color.

Object Attributesリストボックスから、foreground colorを選択します。

0 Under Color Connection Type click on 1 Digital.

Color Connection Typeでは、1 Digitalをクリックします。

0 In the Digital Color Selector Map field, click on the **Chevron:** to select the high alarm parameter of the PID block:

Digital Color Selector Map領域で、PIDブロックの上限アラームのパラメーターを選択するために**Chevron:**  をクリックしてください :

CS.Tank.LC100.HiAlarmStat.

You set the PID/SIM loop's, high alarm value to 75. The display will go into alarm when the value reaches 75 or higher. PID/SIMループのhigh alarm value (上限警報値) に75をセットします。値が75以上に達するとディスプレイはアラームに入ります。

6. To the left of the Chevron there is a two colored rectangle. The left side should be green and the right side should be red. If not do the following:

Chevronの右側に、色の付いた2つの四角形があります。左側は緑、右側は赤であるべきです。そうでなければ、下記を行ってください：

0 Double-click on the left side of the rectangle, select **green**, then click on **OK**.

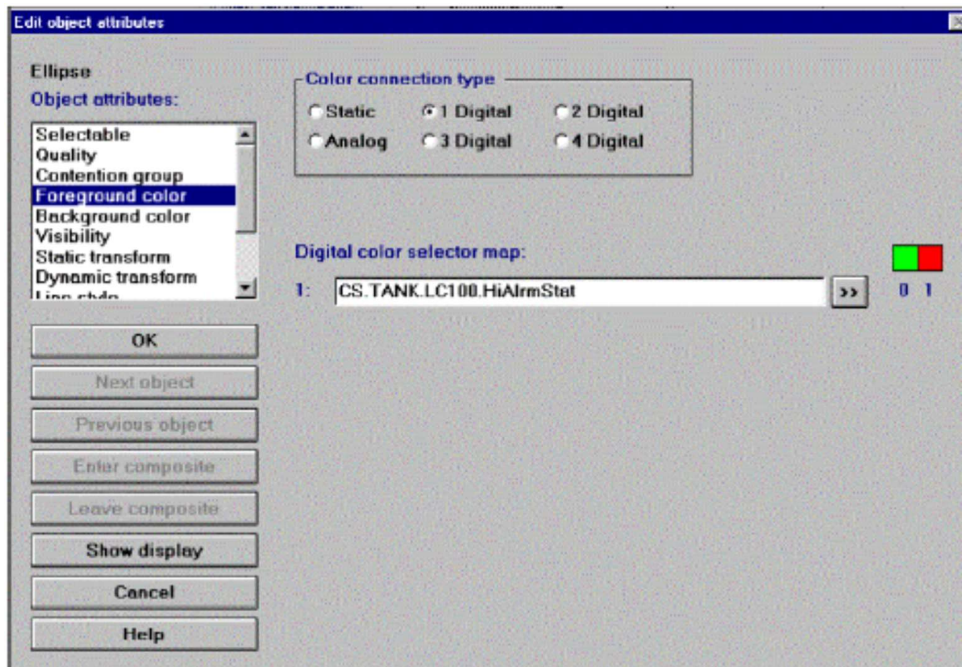
左側の四角をダブルクリックして、**緑**を選択し、**OK**をクリックします。

0 Double-click on the right side of the rectangle, select **red**, then click on **OK**.

右側の四角をダブルクリックして、**赤**を選択し、**OK**をクリックします。

The 0 color is green. This is the color that is displayed when the process is not in alarm. The 1 color (1 digital) is the color associated with the HighAlarmStat Parameter. When the tank's value reaches 75 or above, the circle will blink and turn red.

0の色は緑です。これはプロセスがアラームにない場合、表示される色です。1の色(1 digital)はHighAlarmStatパラメータに関連した色です。タンクの値が75以上に達する時、円は明滅し、赤くなるでしょう。



7. Exit the dialog box, by clicking on **OK**.

**OK**をクリックして、ダイアログ・ボックスを出てください。

8. Repeat the same procedure for the other circle, except enter the low alarm parameter as the connection:

**CS.Tank.LC100.LoArmStat.**

もう1つの円のために、low alarm（下限警報）パラメータのconnection（接続）に：**CS.Tank.LC100.LoArmStat**を入力する以外は、同じ手続きを繰り返してください。

In the PID/SIM loop you set the low alarm value to 25. Therefore, when the tank's value drops to 25 or below, the circle will blink and turn red.

PID/SIMループでは、low alarm value（下限警報値）に25をセットします。したがって、タンクの値が25以下に落ちる時、円は明滅し、赤くなるでしょう。

## Labeling The Circles

円にラベルを付ける


The next step is to create text labels for the circles that blink and change color when they go into alarm.

次のステップは、それらがアラームに入る場合、明滅する円および変更色のためのテキスト・ラベルを作成することです。

#### To create text labels:

テキスト・ラベルを作成するには：


1. Select the **Text Tool**: and set the foreground color to **blue** and the background color to **gray**.

**Text Tool:**  を選択して、foreground color（前景色）に青そしてbackground color（背景色）に灰色をセットしてください。

2. Click to the right of the first circle and type High Alarm. Click to the right of the second circle and type Low Alarm.

1つ目の円の右側をクリックして、High Alarmとタイプしてください。2つ目の円の右側をクリックして、Low Alarmとタイプしてください。

3. Select the **Edit Tool**: and double-click on the High Alarm text field to enter its attributes. Configure the window by entering the following parameters:


**Edit Tool:**  を選択して、その属性を入力するためにHigh Alarmテキスト領域をダブルクリックしてください。次のパラメーターを入力してウィンドウを設定してください：

0 From the Object Attributes box select Foreground, then select 1 Digital.

Object AttributesボックスからForegroundを選択し、次に1 Digitalを選択します。

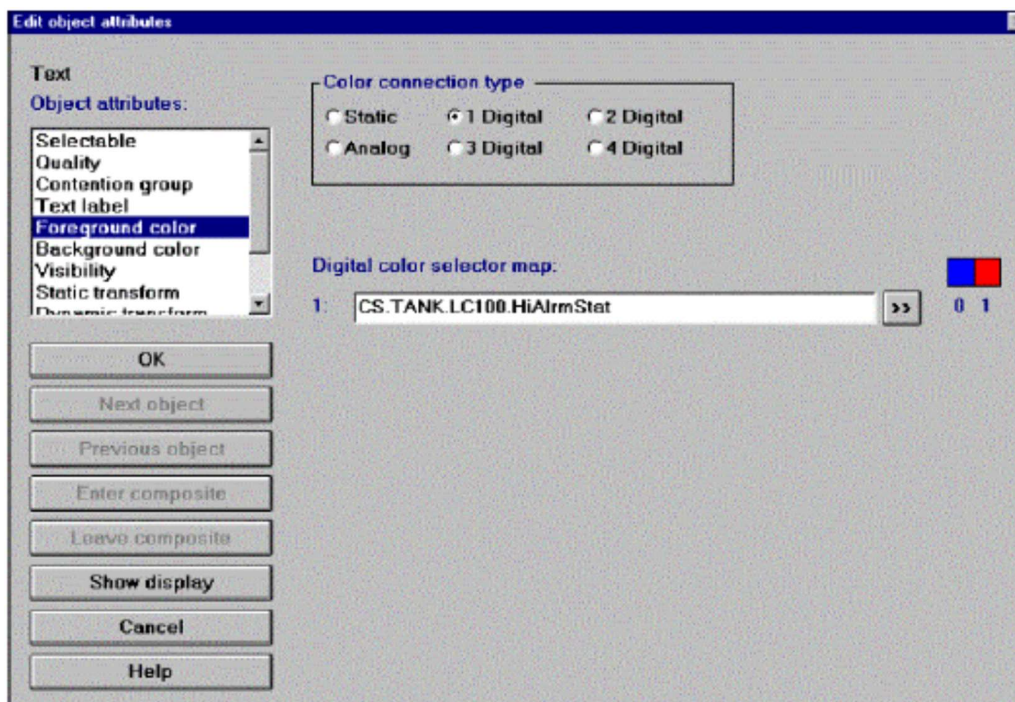
0 In the Connection field click on the Chevron: to set the high alarm parameter of the PID block:

CS.Tank.LC100.HiArmStat. This is the same parameter that you used for its corresponding circle.

**Chevron:**  をクリックしてConnection領域でPIDブロックのhigh alarm（上限警報）パラメーターをセットする：CS.Tank.LC100.HiArmStat. これはテキストに対応した円で使用したのと同じパラメーターです。

0 Select blue as the 0 color and red as the 1 color.

0の色に青そして1の色に赤を選択してください。



0 Select Visibility from the Object Attributes list box.

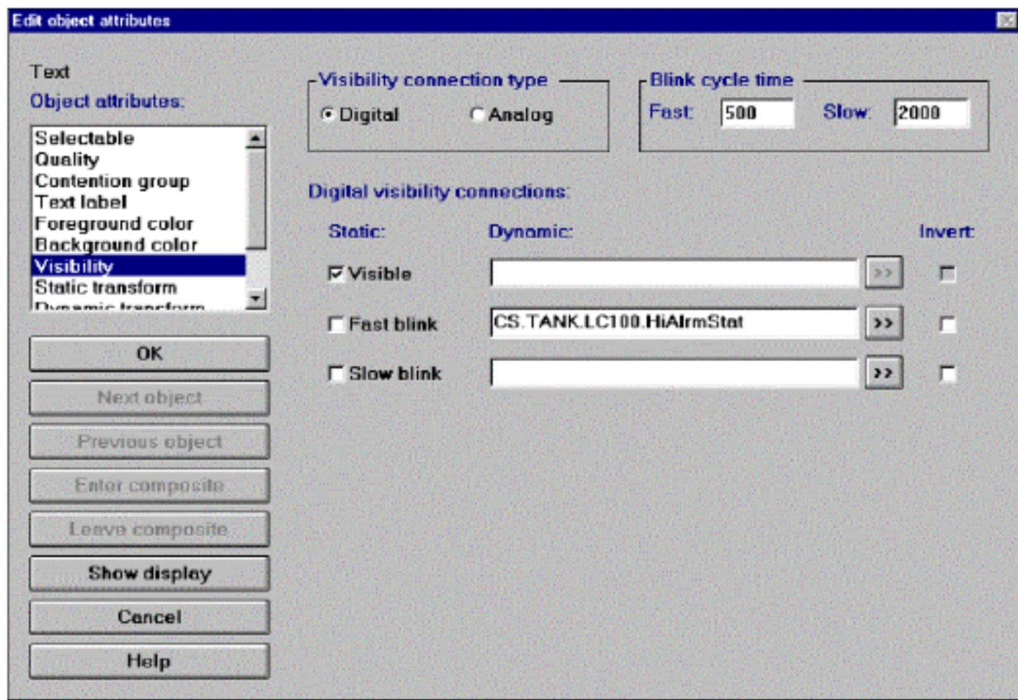
Object Attributes リストボックスから Visibility を選択してください。

0 Choose Digital as the Visibility Connection Type. Under Static make sure that Visible is checked.

Visibility Connection Type で Digital を選びます。Static の下で Visible がチェックされているのを確かめてください。

0 Click on the Chevron: next to the second Dynamic field and select: CS.Tank.LC100.HiAlrmStat

2つ目の Dynamic 領域の Chevron をクリックして、次のものを選択してください : CS.Tank.LC100.HiAlrmStat



4. To exit the dialog box, click on **OK** then **OK** again.

ダイアログ・ボックスを出るために、**OK** をクリックして、もう 1 回 **OK** をクリックしてください。

5. Repeat the above procedure for the Low Alarm label, except select the low alarm parameter:

**CS.Tank.LC100.LoAlrmStat** for both the **Color** and **Visibility** connection fields.

Low Alarm (下限警報) ラベルのために、low alarm パラメーター : **Color** と **Visibility** の connection 領域の両方のために **CS.Tank.LC100.LoAlrmStat** を選択する以外は、上記の手続きを繰り返してください :

## Adding Radio Buttons

### ラジオボタンを追加する


The next step is to add radio buttons that cause the process to run, pause, and reset when selected.

次のステップは、プロセスの **実行(run)**、**停止(pause)**、そして **リセット(reset)** を選択するためのラジオボタンを追加することです。

#### To add radio buttons:

ラジオボタンを追加するには :

1. Select the **Radio Button Tool**: and set **purple** as the foreground color.

**Radio Button Tool**:  を選択して、foreground color (前景色) に **紫** をセットしてください。

2. Place three buttons to the right of Low Alarm. Resize them if needed.



Low Alarmの右側に3つのボタンを置いてください。必要ならばボタンをサイズ変更してください。

3. Select all three radio buttons, right-click on **gray** and set it as the background color.

3つのボタンをすべて選択して、右クリックしてbackground color（背景色）に**灰色**をセットしてください。

### Making A Composite

組み合わせを作る

For the radio buttons to work together, you need to make a composite out of them.

ラジオボタンが相互に動作するには、それらの組み合わせ(composite)を作成する必要があります。

#### To make a composite:

組み合わせを作るには：

1. With all three radio buttons still selected, select **Objects**▶**Make Composite** from the Main menu. The status line at the bottom of your screen should read "3 Objects in Composite."

3つのラジオボタンすべてが選択されている状態のまま、メイン・メニューから**Objects**▶**Make Composite**を選んでください。スクリーンの下部の状態表示行は、"3 Objects in Composite."（組み合わせの3つのオブジェクト）と読める筈です。

2. Double-click on the composite to open its attributes dialog box. Click on the **Enter Composite** push button. The first radio button's attributes are shown. Objects in a composite are cycled through an order based on which object was placed on the display first.

そのattributesダイアログ・ボックスを開くためにcomposite（組み合わせ）をダブルクリックしてください。**Enter Composite**プッシュボタンをクリックしてください。1つ目のラジオボタンの属性が示されます。組み合わせされたオブジェクトはどのオブジェクトが最初にディスプレイに置かれたかに基づく順番で循環します。

3. For the first radio button enter the following parameters:

1つ目のラジオボタンのために、次のパラメーターを入力してください：

0 In the **Label** field type Run.

**Label**領域にRunとタイプしてください

0 From the **Objects Attributes** list box select **Check Actions**. This assigns the appropriate action to occur when this button is checked.

**Objects Attributes** リストボックスから、**Check Actions**を選択します。これは、このボタンがチェックされた時に発生させる適切なアクションを割り当てます。

0 Click on **Set Value** from the **Actions** list box.

**Actions** リストボックスで、**Set Value**をクリックしてください。

0 Under **Type** select **Digital**.

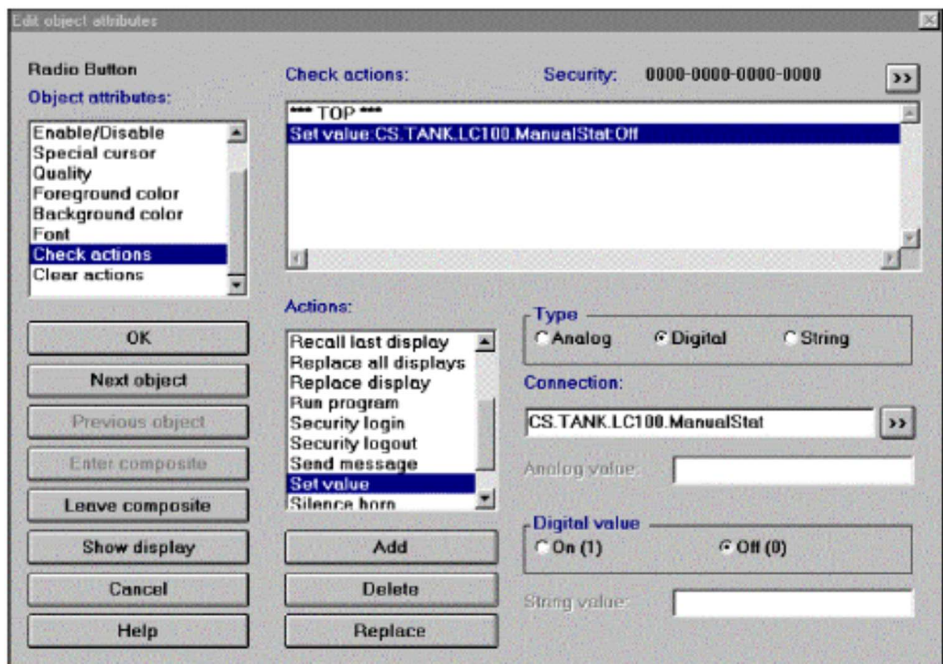
**Type**では**Digital**を選択してください。

0 From the **Connection Field** click on the **Chevron**: and select **CS.Tank.LC100.ManualStat**.

**Connection Field**の**Chevron**:  をクリックして、**CS.Tank.LC100.ManualStat**を選択してください。

0 Under **Digital Value** select **Off(0)**, then press **Add**.

**Digital Value**では**Off(0)**を選択して、**Add**を押してください。



0 From the **Object Attributes** list box select **Clear Actions**. This turns this button off when another is selected.

**Object Attributes** リストボックスから、**Clear Actions** を選択します。これは他のボタンが選択されている時に、このボタンをオフにします。

0 Use the same parameters that were used for **Check Actions**, except select **On(1)** for the **Digital Value**. Then click on **Add**. This turns the process on when it is selected.

**Digital Value** で **On(1)** を選択する以外は、**Check Actions** で使用したのと同じパラメーターを使用してください。次に、**Add** をクリックしてください。これは選択された時に、プロセスをオンにします。

4. To enter the second radio button's attributes, click on the **Next Object** button and enter the following parameters:

2つ目のラジオボタンの属性を入力するために、**Next Object** ボタンをクリックして、次のパラメーターを入力してください：

0 Select the **Label** attribute and type **Pause** into the **Label** field.

**Label** 属性を選択して、**Label** 領域に **Pause** とタイプしてください。

0 Enter the same parameters that you used for the previous button, except select the **On(1)** radio button for the **Digital Value**.

**Digital Value** で **On(1)** のラジオボタンを選択する以外は、前のボタンに使用したのと同じパラメータを入れてください。

0 Retain the same parameters except set the Digital Value to **Off(0)**. Then click on **Add**.

**Digital Value** に **Off(0)** をセットする以外は、同じパラメーターを保持してください。その次に **Add** をクリックしてください。

5. To enter the third radio button's attributes, click on the **Next Object** button and enter the following parameters:

3つ目のラジオボタンの属性を入力するために、**Next Object** ボタンをクリックして、次のパラメーターを入力してください：

0 Select the **Label** attribute and type **Reset** into the **Label** field.

**Label** 属性を選択して、**Label** 領域に **Reset** とタイプしてください。

0 Select the **Check actions** attribute.

**Check actions** 属性を選択してください。

0 From the **Actions** list-box select **Set Value**.

**Actions** リストボックスから、**Set Value** を選択します。

0 Under **Type** select **Analog**.

**Type** では **Analog** を選択します。

0 From the **Connection Field** select: **CS.Tank.LC100.Setp**.

**Connection** 領域で次のものを選択します：**CS.Tank.LC100.Setp**

0 Type **50** into the **Analog Value** field, then click on **Add**.

**Analog Value** 領域に **50** をタイプして、**Add** をクリックします。

0 With **Set Value** still selected, select **Digital** as the type.

まだ **Set Value** を選択したまま、type として **Digital** を選択してください。

0 From the **Connection field** select: **CS.Tank.LC100.Manual.Stat**.

**Connection** 領域で次のものを選択します：**CS.Tank.LC100.Manual.Stat**

0 Under digital value select **Off(0)**, then click on **Add**.

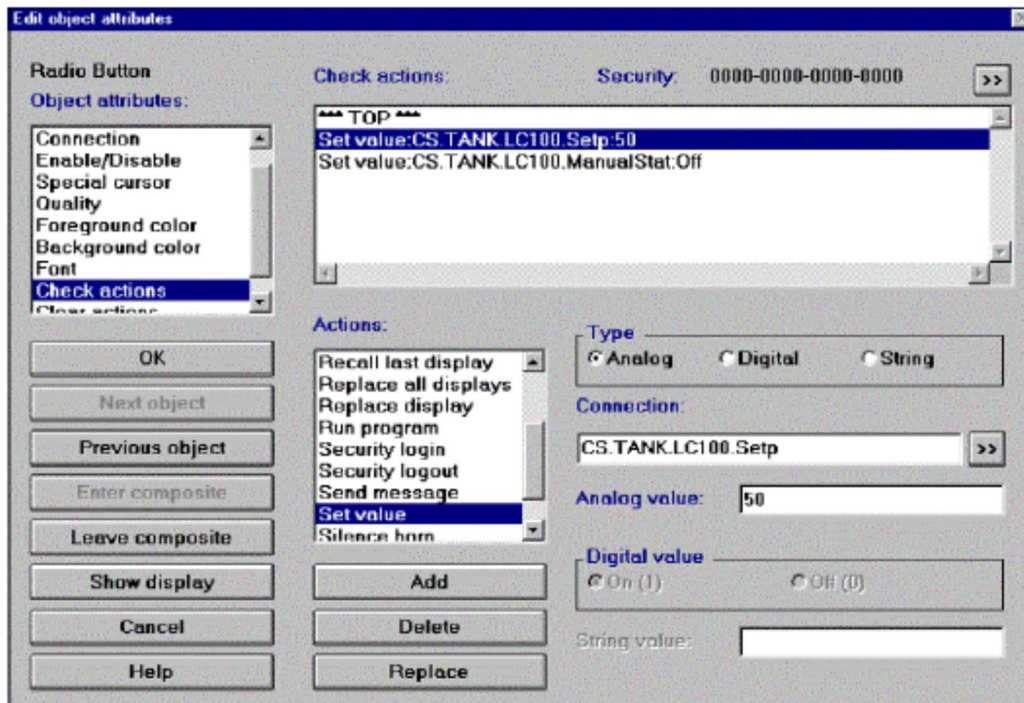
digital value で **Off(0)** を選択して、**Add** をクリックします。

0 Select the **Clear Actions** attribute.

**Clear Actions** 属性を選択します。

0 Retain the same parameters except set **Digital Value** to **On(1)**, then press **Add**.

**Digital Value** に **On(1)** をセットする以外は、同じパラメーターを保持してください。その次に **Add** をクリックしてください。



Click on the **Leave Composite** push button to return to the initial attributes dialog.

最初のattributesダイアログに戻るために**Leave Composite**プッシュボタンをクリックしてください。

6. Click on **OK** to exit out of the dialog box.

ダイアログ・ボックスから出るために**OK**をクリックしてください。

# Creating A PID Faceplate

## PIDフェースプレートの作成


The next step is to create a PID faceplate. First you need to create the bars that correspond to the setpoint, measure, and output functions.

次のステップはPIDフェースプレートを作成することです。最初にセットポイント、測定と出力の関数に対応する棒を作成する必要があります。

### To create the the bars for the setpoint, measure and output functions:

セットポイント、測定と出力の関数のための棒を作成するには：

1. Use the **Polygon Tool**: to create a rectangle to the right of the tank (about 3/4" away) that is the same height as the tank and 3 grid spaces wide. Select **green** as the foreground color.

**Polygon Tool**: を使用して、タンクと同じ高さでグリッド・スペース3つ分の幅を持つ長方形をタンクの右側（およそ3/4インチ離れたところ）に作成してください。foreground color（前景色）に**緑**を選択してください。

2. Copy the bar and paste it directly to the right of the first. Paste the second a little farther to the right so there is room to place a measuring line in between the second and third bar.

棒をコピーして、最初のもの右側にペーストして（貼り付けて）ください。さらにもう少し右側に2つ目と3つ目の棒の間に測定線の余地をおいて2回目のペーストをしてください。

3. Select the second bar and change its foreground color to **red**.

2つ目の棒を選択して、foreground color（前景色）を**赤**に変更してください。

4. Select the third bar and change its foreground color to **purple**.

3つ目の棒を選択して、foreground color（前景色）を**紫**に変更してください。

5. Select the **Polyline** tool, then **Line Attributes**, from the Main menu. Type **15** into the **Weight** field, then click on **OK**.  
メイン・メニューから、**Polyline** tool、次に**Line Attributes**を選んでください。**Weight**領域で**15**とタイプして、**OK**をクリックしてください。

6. Select black as the foreground color and draw a vertical line between the second and third bar that is the same height as the bars. Add horizontal lines that are two grid spaces wide at the top, bottom, midpoint, and quarter points of the line.

foreground color（前景色）に黒を選択して、2つ目と3つ目の棒の間に棒と同じ高さの垂直線を引いてください。線の上端、下端、中間点、および1/4地点に、グリッド・スペース2つ分の幅の水平線を加えてください。

7. Select the **Edit** tool then enter the green bar's **Attributes dialog** box.

**Edit** toolを選択して、次に緑の棒の**Attributes**ダイアログボックスに入ってください。


8. Select the **Percent Fill** attribute by clicking on the **Chevron**: next to the **Percent Fill Connection** field and selecting: **CS.Tank.LC100.Setp**. This is the setpoint parameter for the PID block. Exit the dialog box.

**Percent Fill Connection**領域の**Chevron**: をクリックし**CS.Tank.LC100.Setp**を選択することで、**Percent Fill**属性を選択してください。これはPIDブロックのセットポイント・パラメーターです。ダイアログ・ボックスを出てください。


9. Do the same for the other two bars. Set the **red bar's** connection to the measure parameter: **CS.Tank.LC100.In**, and the **purple bar's** connection to the out parameter: **CS.Tank.LC100.Out**.

他の2本の棒のために同じことをしてください。赤の棒のmeasure（測定）パラメーターに**CS.Tank.LC100.In**の接続、紫の棒のout（出力）パラメーターに**CS.Tank.LC100.Out**の接続をセットしてください。

10. Select the **Data Display Tool**: , and set **gray** as the background color and **green** as the foreground color. Place a data display field underneath the green bar. With the **Data Display** tool still active, select red as the foreground color and place one beneath the red bar. Place another beneath the purple bar with purple as the foreground color.

**Data Display Tool:**  を選択して、background color (背景色) に灰色そしてforeground color (前景色) に緑をセットしてください。緑の棒の真下にデータ表示領域を置いてください。Data Display toolがアクティブな状態のまま、赤の棒の真下にも1つ置いてforeground color (前景色) に赤を選択してください。紫の棒の真下にもforeground color (前景色) が紫で置いてください。

11. Select the **Edit Tool:** and enter the green data display field's **Attributes Dialog** box. Set the **Length** to 5 and **Decimal Places** to 1. Click on the **Chevron:** next to the **Connection** field and select: **CS.Tank.LC100.Setp**. This is the setpoint parameter for the PID block (desired tank level).

**Edit Tool:** を選択して、緑のデータ表示領域の **Attributes** ダイアログ・ボックスに入ってください。Length に5そして **Decimal Places** に1をセットしてください。Connection 領域の **Chevron:**  をクリックして、**CS.Tank.LC100.Setp** を選択してください。これはPIDブロック(要求タンクレベル) のセットポイント・パラメータです。

12. Configure the other data display fields in the same manner, setting the **red's** connection to **CS.Tank.LC100.In** and the **purple's** connection to: **CS.Tank.LC100.Out**. The **In** parameter is the measure parameter (tank's actual level); the **Out** specifies the needed change to bring the tank to its setpoint level.

同じ方法で他のデータ表示領域を設定してください。赤には**CS.Tank.LC100.In**の接続、紫には**CS.Tank.LC100.Out**の接続を設定してください。Inパラメータは測定パラメータ(タンクの実際のレベル)です; Outはタンクをそのセットポイント・レベルにもたらすために必要とされる変更を指定します。

13. Using **blue** as the foreground color, label the **green bar** as Setp, the **red bar** as Meas, and the **purple bar** as Out. foreground color (前景色) に青を使用して、緑棒のラベルをSetp, 赤棒のラベルをMeas, 紫棒のラベルをOutとしてください。

# Display Call-up

## ディスプレイの呼び出し

**Child Display:** overlays the parent display, but cannot be moved outside of its boundaries. If you close the parent display, the child display will close too.

**Child Display (子ディスプレイ) :** 親ディスプレイ (parent display) の上に表示されますが、親の境界より外に移動することはできません。親ディスプレイを閉じると、子ディスプレイ (child display) も閉じます。

**Sibling Display:** is independent of the parent display. It can be moved outside the parent display, and will not close if the parent is shut down.

**Sibling Display (兄弟ディスプレイ) :** 親ディスプレイに依存しません。親ディスプレイの外に移動することができ、親がシャット・ダウンされても閉じません。


The last step is to add push buttons that will call-up the displays that you are going to build in the next exercise. The first button will close the display. The second loads a **Child display**, which overlays the Parent display, but cannot be moved outside of its boundaries. If you close the Parent display the Child display will close too. The third display is called a Sibling display. A **Sibling display** is independent of the parent display. It can be moved outside the parent display, and will not close if the parent is shut down.

最後のステップは、次の課題で構築するディスプレイを呼び出すためのプッシュボタンの追加です。1つ目のボタンはディスプレイをクローズします。2つ目は**子ディスプレイ**（それは親ディスプレイを覆います）をロードしますが、親の境界の外に移動させることはできません。親ディスプレイを閉じると子ディスプレイも閉じます。3つ目のディスプレイは兄弟ディスプレイと呼ばれます。兄弟ディスプレイは親ディスプレイに依存しません。それは親ディスプレイの外に移動することができ、親がシャット・ダウンされても閉じないでしょう。

### To add the push buttons to your display:

ディスプレイにプッシュボタンを追加するには :

1. Select the **Push Button Tool:** and place a push button on the top right of the display window. Paste two more buttons beneath the first.

**Push Button Tool:**  を選択して、ディスプレイ・ウィンドウの右上にプッシュ・ボタンを置いてください。1つ目の下に、さらに2つのボタンを貼りつけてください。

2. Select all three buttons and double-click on the selection to enter the attributes dialog box. The first button's Object Attributes dialog box will be displayed.

3つのボタンをすべて選択して、attributesダイアログ・ボックスに入るために選択をダブルクリックしてください。1つ目のボタン・オブジェクトのAttributesダイアログ・ボックスが表示されます。

0 Select the **Label** attribute and type Close into the **Label** field.

**Label**属性を選択して、**Label**領域にCloseとタイプしてください。

0 Select the **Actions** attribute.

**Actions**属性を選択してください。

0 From the **Actions** list box select **Close Display**, then click on **Add**. When this button is selected, the display will close.

**Actions** リストボックスから、**Close Display** を選択し、次に**Add** をクリックしてください。このボタンが選択されると、ディスプレイは閉じます。

3. Click on the **Next Object** button to open the second button's Object Attributes dialog box.

2つ目のボタン・オブジェクトのAttributesダイアログ・ボックスを開くために、**Next Object** ボタンをクリックしてください。

0 Select the **Label** attribute and type Trend in the **Label** field.

**Label**属性を選択して、**Label**領域にTrendとタイプしてください。

0 Select the **Actions** attribute.

**Actions**属性を選択してください。

0 In the Actions list box, click on **Load Child Display**. Type Trend in the Display Name field, then click on **Add**. This will load the Trend display that you are going to build in the next chapter as a child display.

Actionsリストボックスで、**Load Child Display**をクリックしてください。Display Name領域にTrendとタイプして、次に**Add**をクリックしてください。これは、子ディスプレイとして次章で構築するTrendディスプレイをロードします。

4. Click on the **Next Object** button to open the third button's Object Attributes dialog box.

3つ目のボタン・オブジェクトのAttributesダイアログ・ボックスを開くために、**Next Object**ボタンをクリックしてください。

0 Select the **Label** attribute and type Alarm into the **Label** field.

**Label**属性を選択して、**Label**領域にAlarmとタイプしてください。

0 Select the **Actions** attribute.

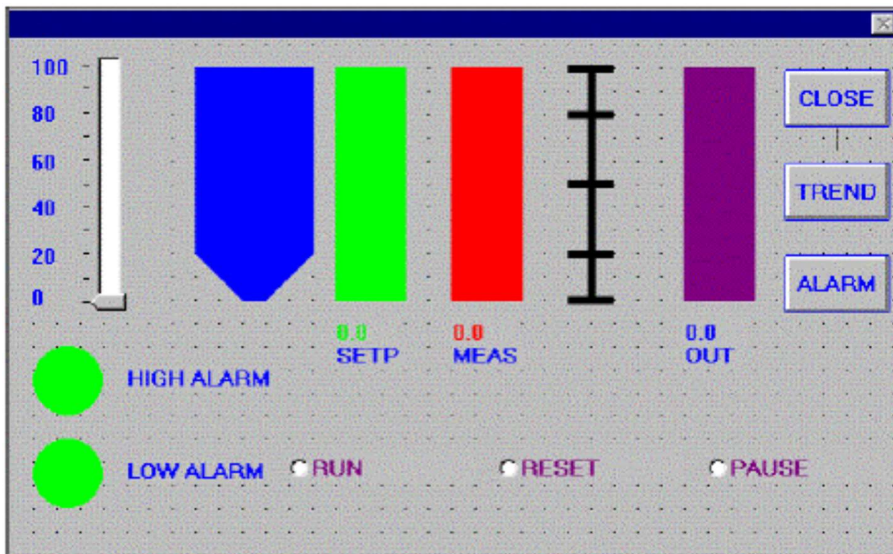
**Actions**属性を選択してください。

0 From the **Actions** list box, select **Load Sibling Display** then type Alarm as the **Display Name**. Click on **Add**. This will load the Alarm display that you are going to build in the next chapter as a sibling display.

**Actions**リストボックスから、**Load Sibling Display**を選択して、**Display Name**にAlarmとタイプしてください。**Add**をクリックしてください。これは、兄弟ディスプレイとして次章で構築するAlarmディスプレイをロードします。

5. Press **OK** to exit the dialog box. Your display should look like the following figure:

ダイアログ・ボックスを出るために**OK**を押してください。ディスプレイは次の図のように見える筈です：



6. Save the display as **Tank.OIL**, then exit, move, or minimize the OI builder.

**Tank.OIL**としてディスプレイを保存して、OI builderを終了するか、移動するか、または最小化してください。

You are now finished with this portion of the Tank exercise.

いまTankの課題のこの部分を完了しました。



# Runtime

## ランタイム

### To enter Runtime:

ランタイムに入るには：

1. Bring the Control Panel to the foreground by clicking on it. Start the Process I/O, the Continuous Strategy and the Operator Interface in this order.

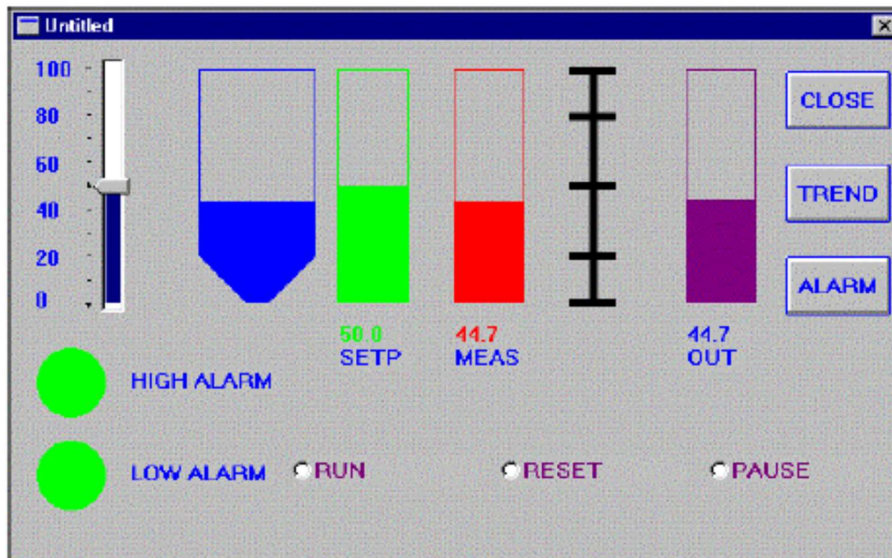
コントロールパネルをクリックして前面に出してください。Process I/O、Continuous StrategyおよびOperator Interfaceをこの順にスタートさせてください。

2. The **Firstapp** display appears. Click on the **Tank** button to bring up the Tank display.

**Firstapp**ディスプレイがあらわれます。Tankディスプレイを表に出すために**Tank**ボタンをクリックしてください。

The current process is running with the setpoint set at 50. The low alarm will flash if it falls below 25, and the high alarm will flash if it goes above 75.

現在のプロセスは50にセットされたセットポイントで実行しています。それが25以下に下降するとLOW ALARMが点滅します。また75以上に上昇するとHIGH ALARMが点滅します。



3. Move the slider arm to change the setpoint to **80**. The high alarm should go off when the output reaches **75**.

スライダーを動かしてセットポイントを**80**に変更してください。出力が**75**に達すると、HIGH ALARMが出る筈です。

4. Move the slider arm to **35**. As the output decreases, click on the **Pause Radio** button to halt the process. Click on the **Run** button to start it up again.

スライダーを**35**に動かしてください。出力を減少させるために、**Pause**ラジオボタンをクリックしてプロセスを中断させてください。再開するために**Run**ボタンをクリックしてください。

5. Once the output reaches **35**, select the **Reset** radio button. The setpoint is reset back to **50**.

一旦出力が**35**に達したら、**Reset**ラジオボタンを選択してください。セットポイントは**50**にリセットされます。

6. When you are through experimenting, close the display.

検証作業を終えるには、ディスプレイを閉じてください。

7. In the Control Panel stop all subsystems by clicking on **Stop Group**.

コントロールパネルから**Stop Group**をクリックしてすべてのサブシステムを停止してください。



## Chapter Summary

### 章のまとめ

Congratulations! You have created your first simulation. In the next chapter you will use the Data Manager to access the data from the Tank process and display it using an Alarm log and a Trend.

おめでとう！最初のシミュレーションを作成しました。次の章では、**Tank**プロセスからデータにアクセスし、かつアラームログおよびトレンドを使用して、それを表示するためにデータ・マネージャーを使用します。

# Trend And Alarm Displays

## トレンドとアラームのディスプレイ

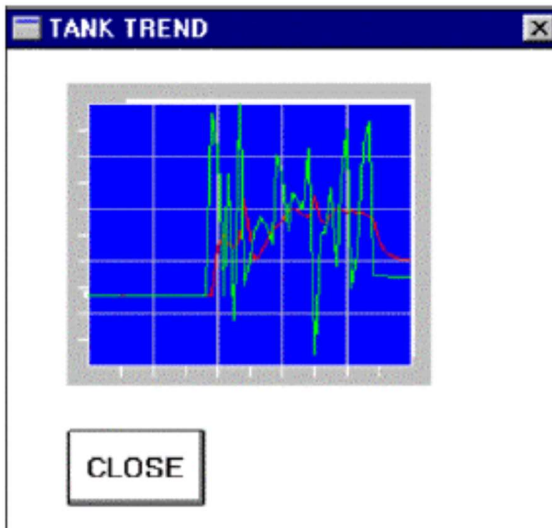
### In This Chapter:

この章には :

- [Data Manager 123](#)
- [Continuous Strategy 129](#)
- [Creating A Trend Display 131](#)
- [Creating An Alarm Log 134](#)
- [Chapter Summary 140](#)

In this exercise you will use the Data Manger to access trend and alarm information from the Tank process. The Trend display will show variations and consistencies in the Tank data. The Alarm display logs high and low alarms from the tank process in realtime.

この課題では、Tankプロセスからトレンド（傾向）およびアラーム（警報）の情報にアクセスするためにデータ・マネージャを使用します。トレンド・ディスプレイは、Tankデータの変化と一貫性を示します。アラーム・ディスプレイ・ログは、リアルタイムにTankプロセスからの上限と下限を警報します。



The screenshot shows a window titled "TANK ALARMS". It features a table with columns for Name, Type, State, Time, Date, and Descriptor. Below the table are several control buttons and checkboxes.

ALARM	All states				
Name	Type	State	Time	Date	Descriptor
CS.TANK.LC100.Out	High Alarm	Active	17:00:43	10/14/97	Level Controlle
CS.TANK.LC100.Out	Low Alarm	Inacti	16:53:36	10/14/97	Level Controlle

Below the table, there are checkboxes for "Disable horn", "Sort by time" (checked), and "Disable update". There are also buttons for "Acknowledge", "Acknowledge all", "Silence", and "Remove". A "CLOSE" button is located at the bottom left of the window.

# Data Manager

## データ・マネージャ

Data Manager is event driven, managing alarm lists, event lists, journals, trend and historical data. It contains periodic sampling and compression functions used to build large historical data collections. History, trends, alarms and events can be automatically moved to relational databases by adding an SQL command to the collection. The movement can be periodic or triggered by an event.

データマネージャはイベントドリブンで、アラームリスト、イベントリスト、ジャーナル、トレンドと履歴のデータを管理します。それは大きな履歴データコレクションを構築するのに使用する周期的なサンプリングと圧縮の機能を持っています。履歴、トレンド、アラームおよびイベントは、コレクションにSQLコマンドを加えることでリレーショナル・データベースに自動的に移動することができます。その移動は周期的もしくはイベントをトリガーに行うことができます。

**To enter the Data Manager builder:**  
**Data Manager builder**に入るには：

From the Application Manager Main menu, select **Builders**▶**Data Manager**.  
Application Managerのメインメニューから、**Builders**▶**Data Manager**を選択してください。

Similar to the Process I/O and Continuous Strategy subsystems, the Data Manager is configured in a two-stage process. You need to:

Process I/OとContinuous Strategyサブシステムと同様に、Data Managerは2ステージの過程で構成されます。次のことが必要です：

Build the Configuration File (\*.DMC) containing the processes to execute during runtime, the nodes they are assigned to, and their default sample rates.

ランタイムで実行するプロセス、それらが割り当てられるノード、およびデフォルトのサンプル・レートを持つ設定ファイル(\*.DMC)を構築してください。

Build the Library File (\*.DMD) which defines the processes and their data management functions, including alarms, trends, and history.

アラーム、トレンドおよび履歴を含み、プロセスおよびそれらのデータ管理関数を定義するライブラリー・ファイル(\*.DMD)を構築してください。

## Creating The library File

### ライブラリ・ファイルの作成

The next step is to create the library file.

次のステップはライブラリ・ファイルを作成することです。

**To create the Library file:**

**ライブラリ・ファイルを作成するには：**

1. From the Main menu select **File**▶**New File**▶**Library**.

メインメニューから**File**▶**New File**▶**Library**を選択してください。

2. Under **Process**, double-click on the **\*\*\*EMPTY\*\*\*** field to bring up the Edit Process Entry dialog box. Configure it by entering the following values:

**Process**の**\*\*\*EMPTY\*\*\***と表示されている領域をダブルクリックして、Edit Process Entryダイアログボックスを出してください。次の値を入力して、それを設定してください：

0 Type TankTrend in the **Name** field, then click on **OK**.

**Name**領域にTankTrendとタイプして、**OK**をクリックしてください。

0 Highlight the **TankTrend** process and click on the Description field. Type Collects Trend Values in the Process Name field, then press the Enter key.

**TankTrend**プロセスをハイライトして、**Description**領域をクリックしてください。**Process Name**領域に*Collects Trend Values*とタイプして、**Enter**キーを押してください。

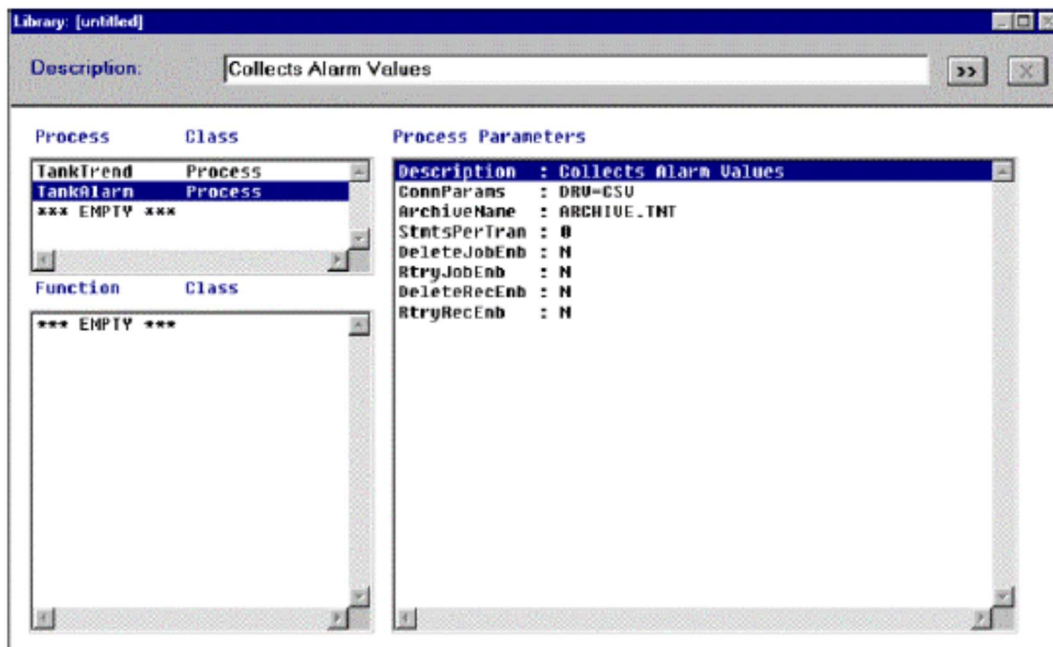
3. Under **Process**, double-click on the **\*\*\*EMPTY\*\*\*** field. To create the TankAlarm process enter the following values:  
**Process**の**\*\*\*EMPTY\*\*\***と表示されている領域をダブルクリックしてください。TankAlarmプロセスを作成するためには、次の値を入力してください：

0 Type TankAlarm in the **Name** field, then click on **OK**.

**Name**領域にTankAlarmとタイプして、**OK**をクリックしてください。

0 Highlight the TankAlarm process and click on the Description field. Type Collects Alarm Values in the **Process Name** field, then press the **Enter** key.

**TankAlarm**プロセスをハイライトして、**Description**領域をクリックしてください。**Process Name**領域に*Collects Alarm Values*とタイプして、**Enter**キーを押してください。



## Configure Trend Functions

### トレンド関数の設定

**To configure the Trend functions:**

トレンド関数を設定するには：

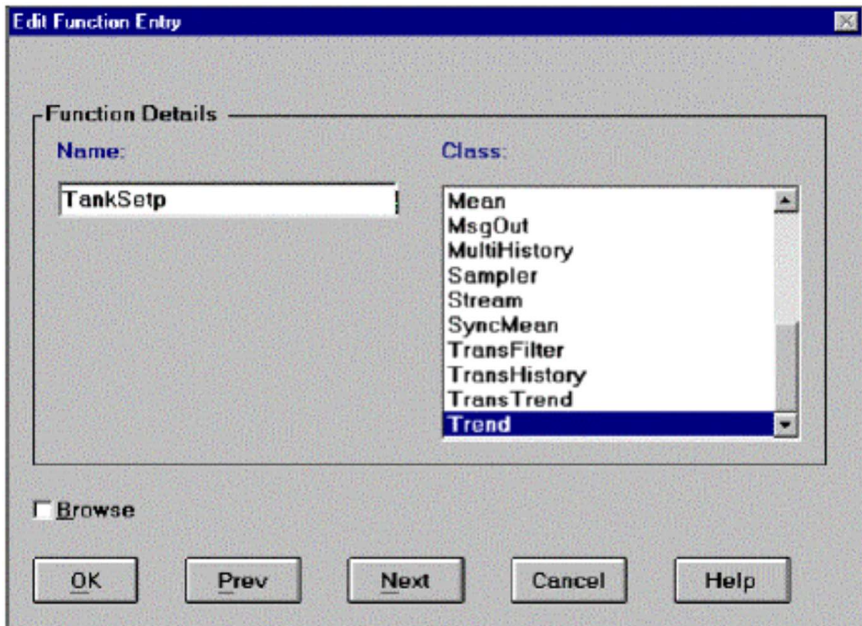
1. Highlight the **TankTrend** process by clicking on it.

クリックして**TankTrend**プロセスをハイライトにしてください。

2. Double click on the **\*\*\*EMPTY\*\*\*** field under **Function Name** to bring up the **Edit Function Entry** dialog box.

**Function Name**の**\*\*\*EMPTY\*\*\***と表示されている領域をダブルクリックして、**Edit Function Entry**ダイアログボックスを出してください。

3. In the **Function Name** field type **TankSetp**, then scroll through the **Class list** box and select **Trend**. Click on **OK**.  
**Function Name**領域にTankSetpとタイプして、**Class list**ボックスをスクロールし**Trend**を選択してください。**OK**をクリックしてください。



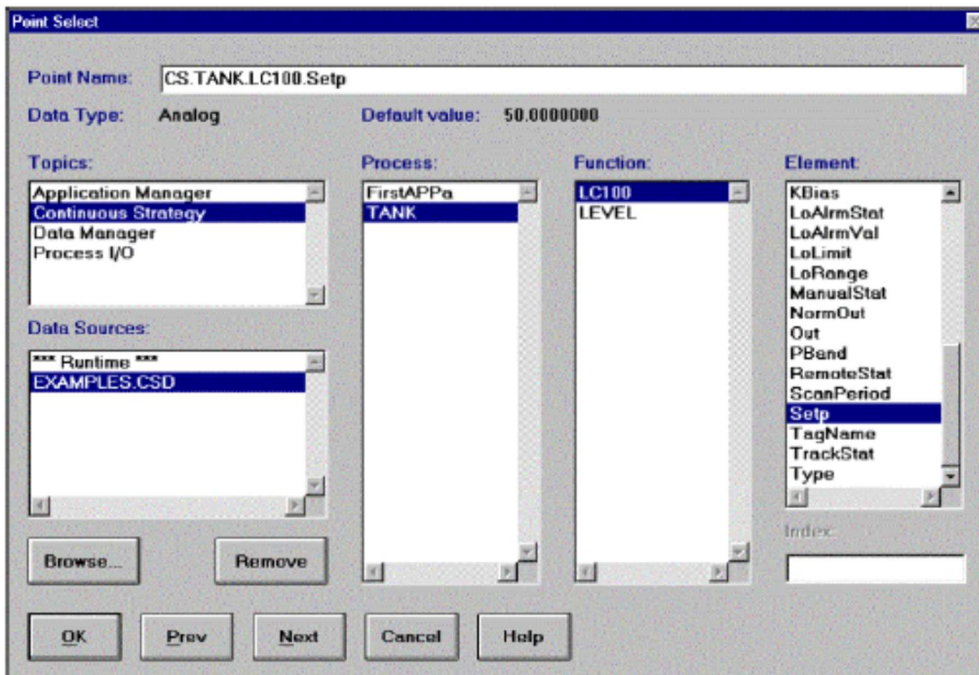
4. Highlight the **TankSetp** function then click on the **Description** field. Type **Collects Trend From Tank Setpoint**, then press **Enter**.

**TankSetp**関数をハイライトして、**Description**領域をクリックしてください。Tank Setpointに**Collects Trend**とタイプして、**Enter**キーを押してください。

5. Double-click on the **SrcConnName** field and select **CS.Tank.LC100.Setp**. Click on **OK**. This field is used to select the parameter in the Tank process that scans for trend information. In this case, the setpoint parameter of the PID block. This parameter will be used to plot the tank's setpoint on the trend.

**SrcConnName** 領域をダブルクリックして、**CS.Tank.LC100.Setp** を選択してください。**OK** をクリックしてください。この領域はトレンド情報のためにスキャンする Tank プロセスのパラメータを選ぶのに使用されます。この場合 PID ブロックのセットポイントのパラメータです。このパラメータは、トレンドのタンクのセットポイントをプロットするのに使用されます。





0 The **ScanPeriod** field defines how often to sample the connected point, and is expressed in milliseconds. Leave the value at 1000.

**ScanPeriod**領域は、接続ポイントのサンプリング周期をミリ秒単位で定義します。1000を値として残してください。

0 The **SyncOffset** field denotes the period in milliseconds after 00:00:00 (midnight) to wait before activating a particular function. This may be used to cause trends to commence at a certain time, such as after the start of a certain shift or batch. Leave the value at 0.

**SyncOffset**領域は、特定の関数をアクティブにする前の待ち時間を00:00:00(真夜中)の後にミリ秒の期間で示します。これはトレンドが一定の時刻に、例えば特定の変化やバッチが始まった後で開始するように使用されるかもしれません。0の値を残してください。

6. The **Size** field denotes the number of samples to include in the trend. Click on the field and type **60** in the **Parameter Value** field. Press **Enter**.

**Size**領域は、トレンドに含めるサンプル数を表示します。**Parameter Value**領域をクリックして**60**とタイプしてください。**Enter**を押してください。



7. Click on the **TankSetp** function, then select **Edit&Insert Entry After**.

**TankSetp**関数をクリックして、**Edit▶Insert Entry After**を選択してください。

8. Enter the next function in the Insert Function Entry dialog box. Configure it by entering the following parameters:  
Insert Function Entryダイアログ・ボックスに次の関数を入力してください。次のパラメータを入力して、それを設定してください：

0 In the **Name** field type TankMeas.

**Name**領域にTankMeasとタイプしてください。

0 Select **Trend** as the **Class**.

**Class**として**Trend**を選択してください。

0 In the **Description** field, type Collects Trend Of Tank Measure.

**Description**領域にCollects Trend Of Tank Measureとタイプしてください。

0 Double-click on the **SrcConnName** field and select **CS.Tank.LC100.In**. This plots the TankMeas function on the trend.

**SrcConnName**領域をダブルクリックして、**CS.Tank.LC100.In**を選択してください。これはトレンドでTankMeas関数をプロットします。

0 Type 60 in the **Size** field, and press **Enter**.

**Size**領域に60とタイプして、**Enter**を押してください。

9. Insert another entry after this function. Configure it by entering the following parameters:

この関数の後に他のエントリーを挿入してください。次のパラメータを入力して、それを設定してください：

0 In the **Name** field type TankOut.

**Name**領域にTankOutとタイプしてください。

0 Select **Trend** as the **Class**.

**Class**として**Trend**を選択してください。

0 In the **Description** field type Collects Trend Of Tank Output.

**Description**領域にCollects Trend Of Tank Outputとタイプしてください。

0 Double-click on the **SrcConnName** field and select **CS.Tank.LC100.Out**. This plots the TankOut function on the trend.

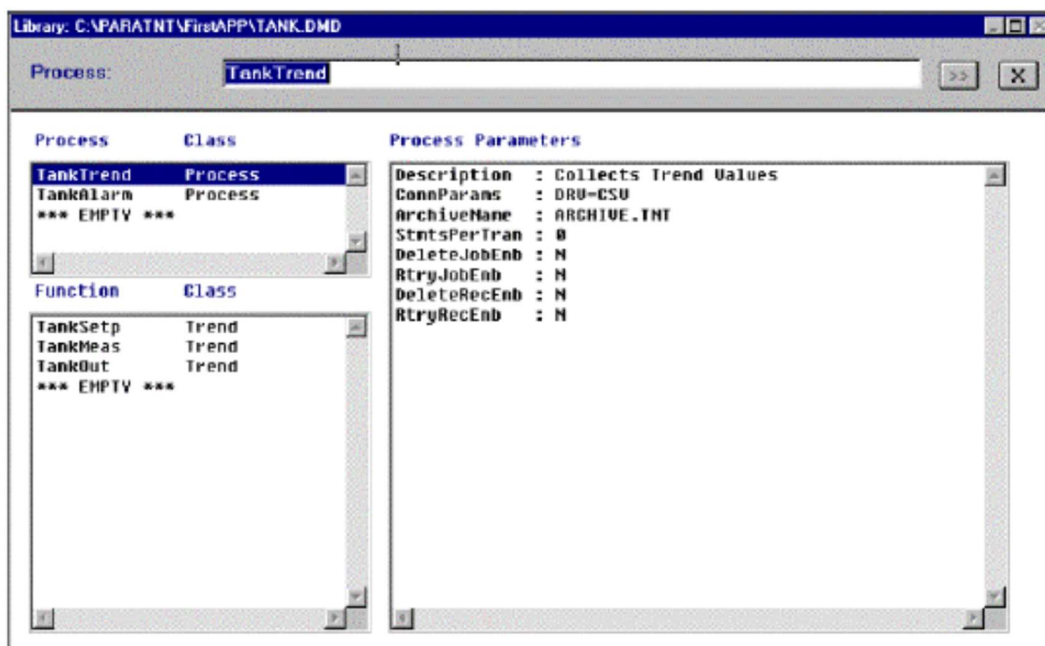
**SrcConnName**領域をダブルクリックして、**CS.Tank.LC100.Out**を選択してください。これはトレンドでTankOut関数をプロットします。

0 Type 60 in the **Size** field, then press **Enter**.

**Size**領域に60とタイプして、**Enter**を押してください。

Your library file window should look like the following:

ライブラリー・ファイル・ウィンドウは下記のように見える筈です：





## Creating The Configuration File

### 設定ファイルの作成

#### To create the Configuration file

設定ファイルを作成するには：

1. From the Main menu select **File**→**New File**→**Configuration**.

メインメニューから**File**→**New File**→**Configuration**を選択してください。

2. Double-click on the **\*\*\*EMPTY\*\*\*** field to bring up the Edit Process Entry dialog box.

**\*\*\*EMPTY\*\*\***と表示されている領域をダブルクリックして、Edit Process Entryダイアログボックスを出してください。

3. You need to define two processes for this configuration file, **TankTrend** and **TankAlarm**.

設定ファイルには**TankTrend**と**TankAlarm**の2つのプロセスを定義する必要があります。

4. To define the **TankTrend** process, configure the window by entering the following values:

**TankTrend**プロセスを定義するには、次の値を入力してウィンドウを設定してください：


0 Type **TankTrend** in the **Process Name** field.

**Process Name**領域に**TankTrend**とタイプしてください。

0 Type **Local** in the **Station Name** field.

**Station Name**領域に**Local**とタイプしてください。

0 Click on the **Chevron:** next to the **Parameter Set** field, and select **Standard**

**Parameter Set**領域の**Chevron:**  をクリックして、**Standard**を選択してください。

0 Type **Tank.DMD** into the **Library Name** field. You will create the **Tank.DMD** file in the next section.

**Library Name**領域に**Tank.DMD**とタイプしてください。次のセクションで**Tank.DMD**ファイルが作成されます。



5. To define the **TankAlarm** process, click on **Next**. Configure the window by entering the following values:

**TankAlarm**プロセスを定義するには、次の値を入力してウィンドウを設定してください：

0 Type **TankAlarm** in the **Process Name** field.

**Process Name**領域に**TankAlarm**とタイプしてください。

0 Type Local in the **Station Name** field.  
**Station Name**領域にLocalとタイプしてください。

0 Click on the **Chevron:** next to the **Parameter Set** field and select **Standard**.  
**Parameter Set**領域の**Chevron:**  をクリックして、**Standard**を選択してください。

0 Type Tank.DMD in the **Library File** field, then click on **OK**. TankTrend and TankAlarm collect their values from the Tank process in the Continuous Strategy.  
**Library Name**領域にTank.DMDとタイプして、**OK**をクリックしてください。TankTrendとTankAlarmは、Continuous Strategy（連続的な戦略）のTankプロセスからそれらの値を集めます

6. Select **File**▶**Save File**▶**Configuration**. Type Tank.DMC into the **File Name** field then click on **OK**.  
**File**▶**Save File**▶**Configuration**を選択してください。**File Name**領域にTank.DMCとタイプして、**OK**をクリックしてください。

## Configuring Alarm Functions

### アラーム関数の設定

#### To configure the Alarm functions:

アラーム関数を設定するには：

1. Switch over to the Library File by selecting **Window&Library**.

**Window▶Library**を選択して、ライブラリ・ファイルに切り替えてください。

2. Click on the **TankAlarm** process to highlight it and then doubleclick on the **\*\*\*EMPTY\*\*\*** field under **Function**.

Configure it by entering the following parameters:

**TankAlarm**プロセスをクリックしてハイライトにして、**Function**の**\*\*\*EMPTY\*\*\***領域をダブルクリックしてください。次のパラメータを入力して、それを設定してください：

0 In the **Name** field type MeasAlarm.

**Name**領域にMeasAlarmとタイプしてください。

0 In the **Class** field select **Alarm**.

**Class**領域で**Alarm**を選択してください。

0 In the Description field type Collects Alarm Data For Tank, then press **Enter**.

Description領域にCollects Alarm Data For Tankとタイプして、**Enter**を押してください。

3. Save the file by selecting **File&Save File&Library**. Type Tank.DMD into the **File Name** field. Click on **OK**.

**File▶Save File▶Library**を選択してファイルを保存してください。**File Name**領域にTank.DMDとタイプして、**OK**をクリックしてください。

4. Close, minimize, or move the Data Manager. To close it select **File&Exit**.

データ・マネージャを閉じるか、最小化するか、移動してください。閉じるには**File▶Exit**を選択してください。

# Continuous Strategy

The next step is to configure the alarms in the Tank process so that they are sent to the functions you just created with the DataManager.

次のステップは、データマネージャと共にいま作成した関数に送られるようにTankプロセスのアラームを設定します。

## To configure the alarms in the Tank process:

Tankプロセスのアラームを設定するには：

1. Enter the Continuous Strategy builder. Select **File** and **Open**, then double-click on **Examples.CSB**. This opens the strategy you created for the Tank process.

Continuous Strategy builderに入ってください。File▶Openを選択して、Examples.CSBをダブルクリックしてください。これは、Tankプロセスのために作成した戦略を開きます。

2. Double-click on the Tank process block to open it. The fields listed under Alarm Destination are where the alarms and events are sent. Alarms and events can be sent to a function within the Data Manager. You can specify eight destinations.

Tankプロセス・ブロックをダブルクリックして開いてください。Alarm Destinationにリストされた領域はアラームとイベントが送られる場所です。アラームとイベントは、データ・マネージャ内の関数に送ることができます。8つのdestinationsを指定できます。

3. To send alarms to the MeasAlarm function you just created in the Data Manager, you must do the following:

MeasAlarm関数にアラームを送るには、いま作成したデータマネージャで、次のことをしてください。

0 Click on the **Chevron**: next to the first **Alarm Destination** field.

1つ目の**Alarm Destination**領域の**Chevron**:  をクリックしてください。

0 Click on **Tank.DMD** and select **TankAlarm** as the **Process**, **MeasAlarm** as the **Function**, and **InPort** as the **Element**:

Tank.DMDをクリックして、ProcessにTankAlarm、FunctionにMeasAlarm、そしてElementにInPortを選択します：

**DM.TankAlarm.MeasAlarm.InPort**

If you don't see **Tank.DMD**, click on browse to locate it.

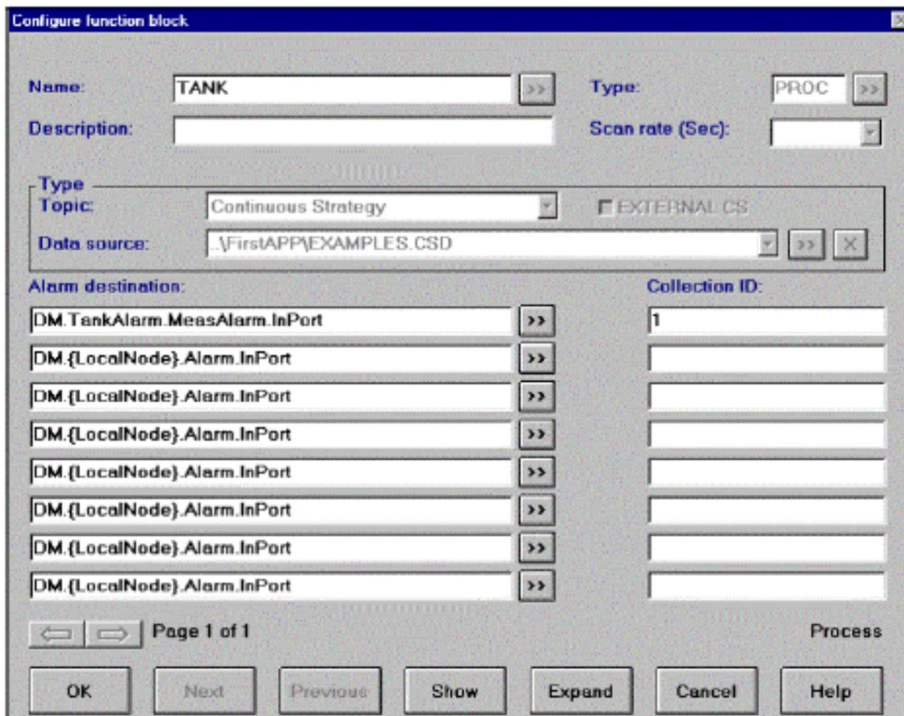
Tank.DMD が見つからなければ、それを見つけるためにbrowseをクリックしてください。

0 Click on **OK**.

OKをクリックしてください。

0 Click in the **Collection ID** field and type 1. Your dialog box should look like the following figure:

Collection ID をクリックして1をタイプしてください。ダイアログ・ボックスは次の図のように見えます：



4. Click on the **Expand** button, then double-click on the PID block to open it. Click on the **Right** arrow to move to the second page.

**Expand** ボタンをクリックして、次にPIDブロックをダブルクリックして開いてください。

You need to set each block's Group parameters in the Tank process to reference this destination. The eight Group parameter checkboxes represent the eight Alarm Destination fields. To send an alarm to the first destination, you would select the first checkbox.

Tankプロセスの各ブロックのGroupパラメタに、このdestination（目的地）を参照するように設定する必要があります。8つのGroupパラメタのチェックボックスが、8つのAlarm Destination領域を意味します。1つめのdestination（目的地）にアラームを送るには、1つ目のチェックボックスを選択します。

5. For the **Group** parameter click on checkbox 1, then on **OK**.

**Group** パラメタのチェックボックス1をクリックして、**OK**をクリックします。

6. Open the SIM block's **Configure Function Block** dialog box and click on the same checkbox for its **Group** parameter. Click on **OK**.

SIMブロックの**Configure Function Block**ダイアログ・ボックスを開いて、その**Group**パラメタの同じチェックボックスをクリックしてください。**OK**をクリックしてください。

7. Save your changes to the **Examples** strategy.

この例の戦略への変更を保存してください。

8. Minimize, move, or exit the Continuous Strategy builder.

Continuous Strategy builderを最小化、移動、もしくは終了してください。



# Creating A Trend Display

## トレンド・ディスプレイの作成

The trend display will load as a child display. A child display overlays its parent display (Tank). It can be moved within the borders of the parent, but not outside of it. Therefore, the Trend display must be built in a location that overlays the Tank display, and is smaller than it.

トレンド・ディスプレイは子ディスプレイとしてロードします。子ディスプレイはその親ディスプレイ(Tank)にかぶさってあらわれます。子は親の境界内で移動可能ですが、その外には移動できません。したがって、トレンド・ディスプレイはTankディスプレイにかぶさり、それより小さな配置で構築される筈です。

### To create a trend display:

トレンド・ディスプレイを作成するには：

1. Enter the OI Builder and create a new display window by selecting **File** > **New**. Configure the window by doing the following:

OI Builderに入って、**File** > **New**を選択して新しいディスプレイ・ウィンドウを作成してください。次のことを行って、ウィンドウを設定してください：

0 Under Window Location and Size, type 0.3575 for **Horizontal**, 0.0500 for **Vertical**, 0.4600 for **Width**, and 0.4400 for **Height**.

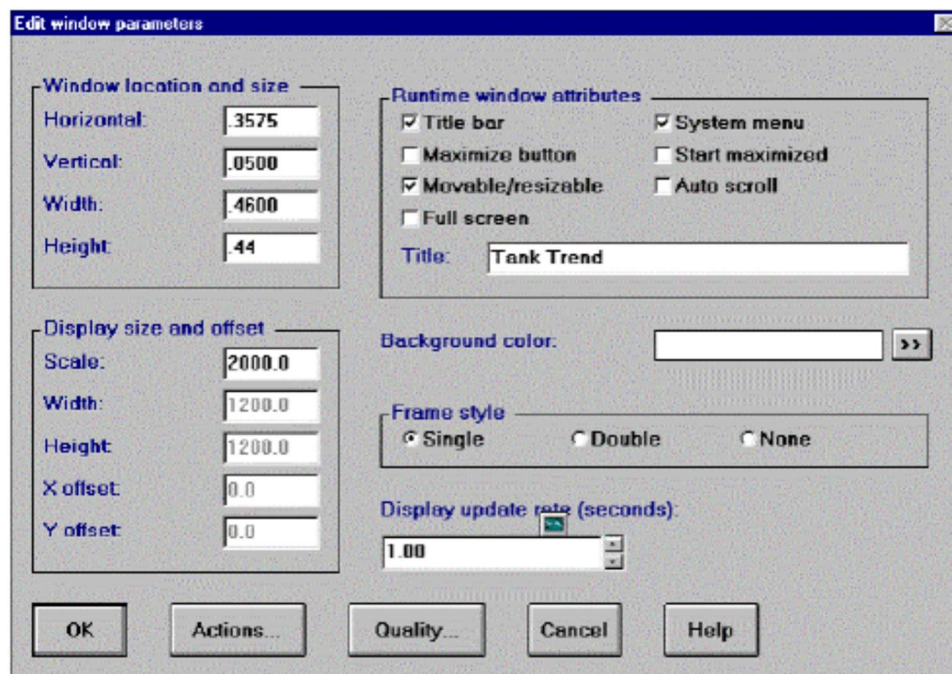
Window Location and Sizeで、**Horizontal**に0.3575、**Vertical**に0.0500、**Width**に0.4600、そして**Height**に0.4400をタイプしてください。

0 Type Tank Trend as the **Title**.


**Title**にTank Trendとタイプしてください。

0 Select white as the background color, then click on **OK**.

background color (背景色) に白を選択して、**OK**をクリックしてください。



2. Select the **Plot Tool**: , then set black as the foreground color and white as the background color. Place a plot on the display window about 1/2" up from the bottom frame.

**Plot Tool:**  を選択して、foreground color（前景色）に黒そしてbackground color（背景色）に白をセットしてください。ディスプレイ・ウィンドウの下枠から1/2インチ程度上にplotを置いてください。

3. Enter its attributes dialog box and select **Type** from the **Object Attributes** list box. Select the **Trend Plot** radio button as the Type.

attributesダイアログボックスに入って、**Object Attributes**リストボックスから**Type**を選択してください。タイプとして**Trend Plot**ラジオボタンを選択してください。

4. Select the **Format** attribute to format the appearance of the trend. Type 1.0 in the **Time Span** field and 1.0 in the **Refresh Rate** field. Leave everything else at their default values.

トレンドの外観の形式設定を行うため**Format**属性を選択してください。**Time Span**領域に1.0、**Refresh Rate**領域に1.0とタイプしてください。ほかの領域はデフォルト値のまま残してください。

5. Select the **Pens** attribute to format the three trend pens. For the first pen:

3本のトレンド・ペンを形式設定するために**Pens**属性を選択してください。1つ目のペンのために：

0 Click on the **First** radio button (1).

**First**ラジオボタン(1)をクリックしてください。

0 In the **Name** field enter the Measure function from the Data Manager: **DM.TankTrend.TankMeas.KeyedDataOut{0}**.

**Name**領域にData ManagerからMeasure function（測定関数）を入力してください：

**DM.TankTrend.TankMeas.KeyedDataOut{0}**

0 For the pen color, click on the **Chevron:** next to the Color field and select **red**.

ペンの色のために、Color領域の**Chevron:**  をクリックして**赤**を選択してください。

0 Leave the other fields at their default values.

ほかの領域はデフォルト値のまま残してください。

6. For the second pen:

2つ目のペンのために：

0 Click on the **Second** radio button (2).

**Second**ラジオボタン(2)をクリックしてください。

0 In the **Name** field enter the Output function from the Data Manager: **DM.TankTrend.TankOut.Keyed DataOut**.

**Name**領域にData ManagerからOutput function（出力関数）を入力してください：**DM.TankTrend.TankOut.Keyed DataOut**

7. For the third pen:

3つ目のペンのために：

0 Click on the **Third** radio button (3).

**Third**ラジオボタン(3)をクリックしてください。

0 In the **Name** field, enter the Setpoint function from the Data Manager: **DM.TankTrend.TankSetp.KeyedDataOut**.

**Name**領域にData ManagerからSetpoint function（セットポイント関数）を入力してください：

**DM.TankTrend.TankSetp.KeyedDataOut**

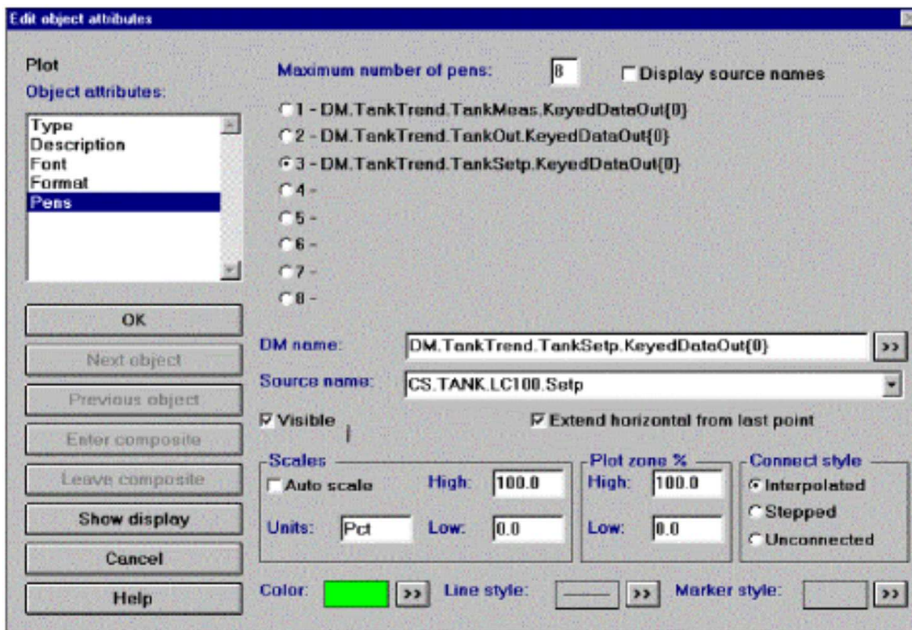
0 Select **green** as the color.

Colorに**緑**を選択してください。

0 Press **OK** to exit the dialog box.

**OK**を押して、ダイアログボックスを終了してください。

Your screen should look like the following:  
スクリーンは次のように見える筈です：




## Adding A Close Button Closeボタンの追加

The final step is to add a button to close the display.  
最終ステップはディスプレイを閉じるためにボタンを加えることです。

**To add a close button:**  
closeボタンを追加するには：

1. Select the **Push Button Tool**. Set white as the background color and black as the foreground color, then place it under the trend.

**Push Button Tool:**  を選択してください。background color（背景色）に白そしてforeground color（前景色）に黒をセットして、トレンドの下に置いてください。

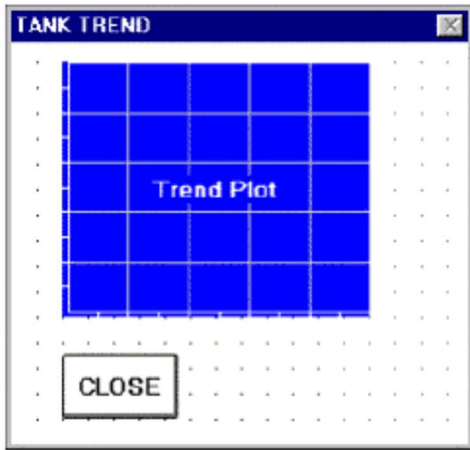
2. Open its attributes dialog box by double-clicking on it. Configure it by doing the following:  
それをダブルクリックしてAttributesダイアログ・ボックスを開いてください。次を行って、それを設定してください：

0 Select the **Label** attribute and type Close in the **Label** field.  
**Label**属性を選択して、**Label**領域にCloseとタイプしてください。

0 Select the **Actions** attribute and chose **Close Display** from the Actions list box.  
**Actions**属性を選択して、Actionsリストボックスから**Close Display**を選択してください。

0 Click on **Add**, then on **OK**.  
**Add**それから**OK**をクリックしてください。

Your trend display is now finished, and should look like the following:  
トレンド・ディスプレイはいま完成して、次のように見える筈です：



3. Save the display as Trend, then close it.

Trendとしてディスプレイを保存して、次にそれを閉じてください。

# Creating An Alarm Log

## アラーム・ログの作成

In this section you will use the Operator Interface to create an alarm log that logs alarms as they occur during runtime. このセクションでは、ランタイム中に生じるアラームを記録するアラームログを作成するためにOperator Interfaceを使用します。

### To create an alarm log:

アラーム・ログを作成するには :

1. From the Main menu select **File** & **New**. Configure the window by doing the following:

メイン・メニューから、**File** & **New**を選択します。次を行ってウィンドウを設定してください :

0 In the **Horizontal** field type 0.0250.

**Horizontal**領域に0.0250とタイプしてください。

0 In the **Vertical** field type 0.3600.

**Vertical**領域に0.3600とタイプしてください。

0 In the **Width** field type 1.1500.

**Width**領域に1.1500とタイプしてください。

0 In the **Height** field type 0.5000.

**Height**領域に0.5000とタイプしてください。

0 Click on the **Auto Scroll** check box, and type Tank Alarms in the **Title** Field.

**Auto Scroll**チェックボックスをクリックして、**Title**領域にTank Alarmsとタイプしてください。

0 Click on **Double** for frame style, then on **OK**.

frame styleで**Double**、そして**OK**をクリックしてください。

2. Select the **Event Tool**: and place it in the lower-left corner of the display window. Resize it until it fills up most of the display window.

**Event Tool**:  を選択して、ディスプレイ・ウィンドウの左下の角にそれを置いてください。ディスプレイ・ウィンドウを覆うように、それをサイズ変更してください。

3. Double-click on it to open its attributes box. Configure the window by entering the following values:

そのAttributesボックスを開くため、それをダブルクリックしてください。次の値を入力して、ウィンドウを設定してください。

0 Select the **Type** attribute, and then **Current** as the type.

**Type**属性を選択して、タイプとして**Current**を選択してください。

0 Select the **Assign** attribute to enter the alarm function from the Data Manager.

Data Managerからアラーム関数を入力するために**Assign**属性を選択してください。

0 In the function name field click on the **Chevron**: and select **DM.TankAlarm.MeasAlarm.Out**.

function name領域の**Chevron**: をクリックして、**DM.TankAlarm.MeasAlarm.Out**を選択してください。

0 In the **Description** field type Alarm, then click on **Add**.

**Description**領域にとタイプして、**Add**をクリックしてください。

0 Click on the **All States** radio button for the Initial Filter to display all alarm log states.

初期フィルタがアラームログ状態をすべて表示するように**All States**ラジオボタンをクリックしてください。

4. Select the **Format** attribute. Configure the window by doing the following:

**Format**属性を選択してください。下記を行うことでウィンドウを設定してください：

0 Under **Sort List By**, click on **Time**. This sorts alarms by their time of occurrence.

**Sort List By**で**Time**をクリックしてください。これは発生時間でアラームをソートします。

0 Under **Include Buttons**, click on all three selections to include: Acknowledge, Silence Horn, Disable Update, Remove, and Disable Horn.

**Include Buttons**では、Acknowledge（承認）、Silence Horn（警笛の沈黙）、Disable Update（更新の無効）、Remove（消去）、そして Disable Horn（警笛の無効）を含む3つの選択をすべてクリックしてください。

The ALL selection enables the Acknowledge, Silence Horn, and Disable Update buttons. During Runtime these button do the following:

すべてを選択すると、Acknowledge、Silence Horn、およびDisable Updateのボタンが有効になります。ランタイムの間、これらのボタンは次のことを行います：

**Acknowledge:** Acknowledges the selected active alarm.

**Acknowledge:** 選択されたアクティブなアラームを承認します。

**Silence Horn:** Silences the alarm horn.

**Silence Horn:** アラームの警告音を沈黙させます。

**Disable Update:** Causes the alarm log to stop updating alarms.

**Disable Update:** アラームログにアラームの更新をやめさせます。

**Remove:** Removes the current alarm from the log.

**Remove:** ログから現在のアラームを消去します。

**Disable Horn:** Stops the horn from sounding.

**Disable Horn:** 警告音が鳴るのを停止します。

4. Select the **Columns** attribute to format which fields are placed in the left and right columns of the alarm log. The default places **Name**, **Type**, and **State** in the left column, and the rest of the fields in the right column. Leave this at its default values, then click on **OK** to exit the dialog box.

どの領域をアラームログの左右の列に置くか形式設定のため **Columns** 属性を選択してください。デフォルトでは左の列に **Name**（名前）、**Type**（タイプ）、そして **State**（状態）、そして右の列に残りの領域が置かれます。そのデフォルト値を残して、次にダイアログ・ボックスを出るために **OK** をクリックしてください。

## Adding A Close Button To Your Display

### ディスプレイにCloseボタンを加える


The final step is to add a button to close your display.

最終ステップはあなたのディスプレイを閉じるためにボタンを加えることです。

**To add a close button:**

**close**ボタンを追加するには：

1. Select the **Push Button Tool**: Set white as the background color and black as the foreground color, then place it under the Alarm Log.

**Push Button Tool:**  を選択して、background color（背景色）に白そしてforeground color（前景色）に黒をセットして、アラームログの下に置いてください。

2. Open its attributes dialog box by double-clicking on it. Configure it by entering the following parameters:

それをダブルクリックしてAttributesダイアログ・ボックスを開いてください。次のパラメータを入力して設定してください：

0 Select the **Label** attribute and type **Close** in the **Label** field.

**Label**属性を選択して、**Label**領域に**Close**とタイプしてください。

0 Select the **Actions** attribute and choose **Close Display** from the Actions list box.

**Actions**属性を選択して、Actionsリストボックスから**Close Display**を選択してください。

0 Click on **Add**, then on **OK**.

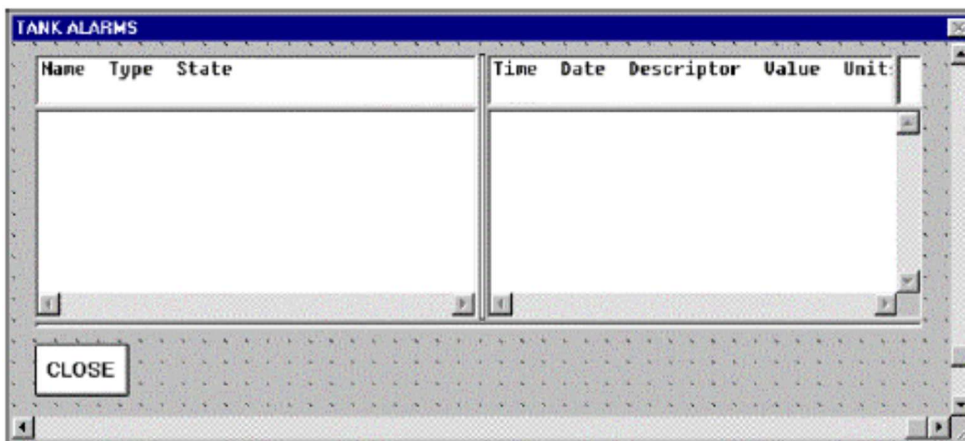
**Add**それから**OK**をクリックしてください。

**Note:** You can size the display window so that it covers or hides a portion of the alarm log (the display is larger than the display window). In Runtime, you can resize the window or use the scroll bars to view hidden parts of the alarm log.

注意：ディスプレイがアラームログの部分に被さるか隠すように、ディスプレイ・ウィンドウをサイズすることができます(表示はディスプレイ・ウィンドウより大きい)。ランタイムでは、ウィンドウをサイズ変更するか、あるいはアラーム・ログの隠れた部分を見るためにスクロール・バーを使用することができます。

Your display should look like the following:

あなたのディスプレイは下記のように見える筈です：



5. Save the display as **Alarm.OIL**, then return to the Application Manager.

**Alarm.OIL**としてディスプレイを保存して、次にApplication Managerに戻ってください。

## Runtime

### ランタイム

**To enter Runtime:**

ランタイムに入るには：

1. In the Control Panel, double-click on the **Data Manager** line. Click on the **Subsystem In Application Group** checkbox.

Control Panelで、**Data Manager**のラインをダブルクリックしてください。**Subsystem In Application Group**チェックボックスをクリックしてください。

2. Use the **Chevron**: to select the Start-up File **Tank.DMC**. Click on **OK**, then **OK** again. The Application Overview window of the control panel should look like the following:

**Chevron**:を使ってスタートアップ・ファイル**Tank.DMC**を選択してください。**OK**を2回クリックしてください。

Control PanelのApplication Overviewウィンドウは次のように見える筈です：

Application overview			
Subsystem	In Group	Running	Startup File
Data Manager	Yes	Yes	TANK.DMC
Process I/O	Yes	Yes	FirstAPP.PIC
Continuous Strategy	Yes	Yes	EXAMPLES.CSC
Operator Interface	Yes	Yes	FirstAPP.OID
DDE Services	No	No	(no startup file)
Engineer Interface	No	No	(no startup file)

3. Select **File**→**Save** to save **Examples.APP**.

**Examples.APP**を保存するために**File**→**Save**を選択してください。

4. Click on **Start Group**, to start all subsystems. If you want to start them individually you must start them in the following order:

**Start Group**をクリックして、すべてのサブシステムを開始してください。サブシステムを個別に開始したければ、次の順で開始しなければなりません：

1. Data Manager
2. Process I/O
3. Continuous Strategy
4. Operator Interface

You can open the Trend and Alarm displays from either the Firstapp display or the Tank display. For this exercise, open it from the Tank display so you can manipulate the setpoint and view the results on the trend.

FirstappディスプレイあるいはTankディスプレイのいずれかからトレンドとアラームのディスプレイを開くことができます。この課題では、セットポイントを操作して、トレンドに関する結果を見ることができるよう、Tankディスプレイからそれを開いてください。

## Using the Display

### ディスプレイの使用

#### To use the display:

ディスプレイを使用するには：

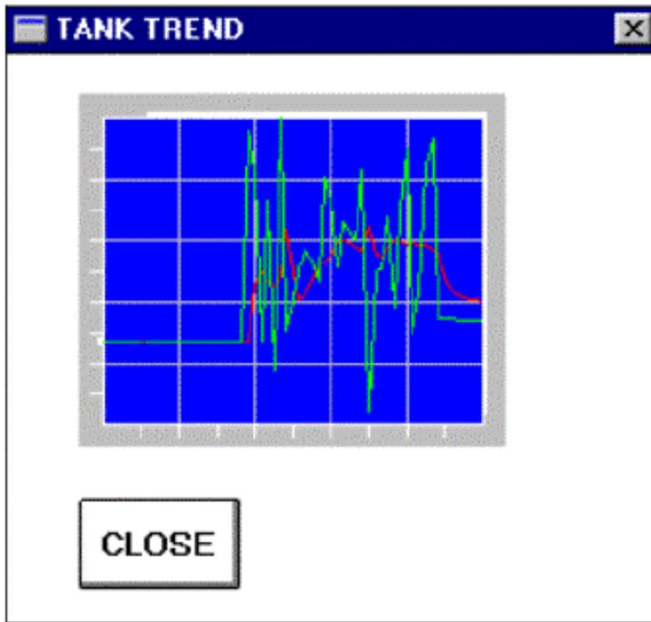
1. Select **Tank** from the **Firstapp** display. Select **Trend** from the Tank display. The Trend display should overlay the bottomright of the Tank display..

**Firstapp**ディスプレイから**Tank**を選んでください。Tankディスプレイから**Trend**を選んでください。Trendディスプレイは、Tankディスプレイの右下を覆ってあらわれます。

2. Move the slider in the Tank display to change the setpoint. Watch the results on the trend. Change the setpoint to cause high and low alarms.

Tankディスプレイのスライダーを動かして、セットポイントを変えてください。トレンドに関する結果を見てください。上限と下限のアラームを引き起こすためにセットポイントを変更してください。





3. Click on the Alarm display. Notice the logged alarms. Resize the window or use the scroll bars to view the entire alarm log. Highlight the current alarm and click on the Acknowledge button. The State changes from Active to Acknowledged.

Alarmディスプレイをクリックしてください。記録されたアラームに注意してください。ウィンドウをサイズ変更するか、あるいはアラームログ全体を見るためにスクロール・バーを使用してください。現在のアラームをハイライトして、Acknowledge（確認）ボタンをクリックしてください。State（状態）はActive（動作中）からAcknowledged（認知）に変化します。

Name	Type	State	Time	Date	Descriptor
CS.TANK.LC100.Out	High Alarm	Active	17:00:43	10/14/97	Level Controlle
CS.TANK.LC100.Out	Low Alarm	Inacti	16:53:36	10/14/97	Level Controlle

Disable horn       Sort by time       Disable update  
           

4. Click on the Tank display to bring it to the foreground. Change the setpoint to bring the process out of alarm. Notice that the alarm has been removed from the log.

それを前面に出すためにTankディスプレイをクリックしてください。アラームからプロセスをもたらすためにセットポイントを変更してください。アラームがログから消去されたことに注目してください。

5. Either resize or move the Alarm display so you can view both displays. Change the setpoint as desired to view changes to the alarm log.

サイズ変更するか、あるいは両方のディスプレイを見ることができるよう、アラーム・ディスプレイを移動させてください。アラームログの変化を見るために、必要なセットポイントの変更を行ってください。

6. When you are through, close the Alarm and Tank displays. The Trend display closes along with the Tank display.

終わったとき、AlarmとTankのディスプレイを閉じてください。TrendディスプレイはTankディスプレイと共に閉じます。

**7. Select Stop Group to close the subsystems.**

サブシステムを閉じるために**Stop Group**を選択してください。

## Chapter Summary

### 章のまとめ

Congratulations! You now know how to use the Data Manger to access trend and alarm information. The Trend display showed you variations and consistencies in the Tank data while the Alarm log tracked high and low alarms as they occurred.

おめでとう！あなたはいま、トレンドおよびアラームの情報にアクセスするためにData Mangerを使用する方法を知っています。トレンド・ディスプレイは、Tankデータの変化と一貫性を示し、その一方でそれらが起こると同時に、アラームログは上限と下限のアラームを追跡しました。

That's it! You should now be able to create a basic Paragon application on your own. To learn more about Paragon's capabilities refer to the User's Guide, the online Reference Manual or contact NemaSoft.

それで全てです！あなたは現在、基本的なParagonアプリケーションをあなた自身で作成することができるはずです。Paragonの能力に関してさらに学ぶには、User's Guide, online Reference Manualを参照するかNemaSoftにコンタクトしてください。

# Index

## A

**ActiveX, 19**

### Alarms

alarm log

creating, 134

Continuous Strategy (CS)

configuring, 129

Data Manager (DM)

configuring, 128

**Analog, 54**

**Anatomy Of A Paragon Application, 24**

### Application Files

Data Manger

internal storage file, 25

Paratnt.TEC, 25

Paratnt.USR, 25

**Application Manager (AM), 20, 68, 93**

application file, 25, 68

saving, 70

security, 29

application encryption, 29

custom rights, 29

passwords, 29

protecting database files, 29

system rights, 29

start-up file, 68, 93

**Application Wizard, 30**

application name, 31

application summary, 40

Comma Separated Variable (CSV), 34

Communication Port setup, 33

Data Manager (DM)

history, 37

collection points, 38

LAST.WIZ, 32, 41

Operator Interface (OI), 39

ActiveX Starter Set, 39

display setup, 39

OI Starter Set, 39

Process I/O setup, 33

Recipes, 34

starting, 31

tag names, 34

process I/O functions, 35

tag generation wizard, 34

### Attributes

line, 114

setting, 90

## B

**Block Development Kit (BDK), 73**

## C

**C, 19**

**C++, 19**

**CAUTION.TXT, 15**

**Chevron, 31, 48**

**Child Display:, 116**

**Client Development Kit (CDK), 73**

**Client Object Interface (COI), 19**

ActiveX, 19

C, 19

C++, 19

Delphi, 19

Java, 19

Visual Basic, 19

**Clients, 19, 22**

**Client-Server**

architecture, 19

**Comma Separated Variable (CSV), 34**

**Communications Port, 33**

**Composites**

making, 111

**CONFIG.SYS file, 14**

time zone setting, 14

**Configuration File, 47**

Continuous Strategy (CS), 24

Data Manger (DM), 127

**Continuous Strategy (CS), 20, 76, 77, 99, 129**

alarms

configuring, 129

**142 ì Paragon Getting Started**

annotation objects, 78

building a strategy, 80

configuration file, 24

display window

creating, 77

expanded layer, 82

function blocks

configuring, 83

connecting, 84, 103

creating, 82

defining, 99

graphics file, 24

library file, 24

object palette

arc tool, 79

bitmap tool, 79

conduit tool, 79

connection tool, 79

edit tool, 79

ellipse tool, 79

function block tool, 79

polygon tool, 79

polyline tool, 79

terminator tool, 79, 84

specifying Process I/O functions, 85

text tool, 79

object palette, 78

PID block, 99, 103

PID/SIM loop, 109

portable file, 24

process blocks, 79

connecting, 81

process layer, 80, 82

saving, 87, 104

SIM block, 99, 103

strategy objects, 78

viewing external connections, 86

**Control Panel, 53, 68**

## **D**

**Data Manager (DM), 20, 122, 123**

accessing, 123

configuration file, 24, 123

building, 127

history, 37

collection points, 38

internal storage file, 25

- library file, 24, 123
- configuring alarms, 128
- configuring functions, 124, 128
- creating, 123
- saving, 128
- using with SQL, 123

#### **Deleting**

- process names, 51

#### **Delphi, 19**

#### **Digital, 108**

#### **Directories Box, 80**

#### **Directory**

- application, 25
- creating for applications, 32
- Paragon
  - Installation icon, 13, 14
  - ODBC Administrator icon, 13, 14
  - Paragon icon, 13, 14
  - root, 14

#### **Discrete, 54**

#### **Drawing, 105**

#### **Driver Database, 30**

### **E**

#### **Engineering Interface (EI), 22, 72**

- data points
- browsing, 72

### **F**

#### **Function Blocks, 82**

- configuring, 83
- types, 83

#### **Functional Software Architecture, 18**

#### **Functions, 21, 47**

### **G**

#### **Grid, 89**

### **I**

#### **Installation**

- custom, 12
- OS/2, 13
- troubleshooting
  - Autorun.exe, 12
  - Windows 95, 12
  - Windows NT, 12

#### **Interactive Trends (ITrends), 22, 43**

#### **Internal Loop, 104**

#### **Internet technology**

#### **Index i 143**

- using with Paragon, 22

#### **Introduction, 5**

### **J**

#### **Java, 19**

#### **Java Tools**

- Borland JBuilder, 23
- IBM Visual Age for Java, 23
- Lotus Bean Machine, 23
- Symantec Visual Café, 23

#### **JavaBean, 19**

### **L**

#### **Labeling a Display, 60**

#### **Labels, 109**

#### **LAST.WIZ, 32, 41**

#### **Library File, 47**

- Continuous Strategy (CS), 24

Data Manager (DM), 24, 123

Process I/O (PIO), 24

**Line Attributes, 114**

**Local Area Network (LAN), 79**

## **M**

### **Manual Conventions**

menu commands, 9

Online help, 9

typeface styles, 9

using the keyboard, 8

using the mouse, 8

## **O**

### **Object Attributes, 62**

data display

setting, 66

data entry field

setting, 62

list box

setting, 64

### **ODBC Administrator, 13**

### **Open Database Connectivity (ODBC), 43**

### **OpenControl (OC), 20**

### **Operator Interface (OI), 22, 39, 54, 88, 105**

accessing, 54

alarm log

creating, 134

aligning objects, 60

assigning actions, 91

attributes

setting, 90

close button

adding, 133, 136

color palette, 55, 56

moving, 88

setting foreground colors, 58

compiled file, 24

creating a bar chart, 89

creating a display window, 57

data display

adding, 59

digital animation, 108

display setup, 39

drawing, 105

drawing right angles, 89

entering, 54

features, 54

grid, 89

labeling a display, 60

line attributes, 114

list box

adding, 58

resizing, 58

making composite objects, 111

object palette, 55

adding radio buttons, 111

arc tool, 56

bitmap tool, 56

data display tool, 56

data entry tool, 56

edit tool, 56

ellipse tool, 56

**event tool, 56**

**list box tool, 56**

moving, 88

- placing objects, 58
- plot tool**, 56
- polygon tool, 56
- polyline tool, 56
- push button tool, 56
- radio button tool, 56
- slider tool**, 56
  - adding to a display, 106
- text tool, 56
- opening a display, 88
- placing objects, 88
- repositioning and object, 59
- saving a display, 67, 137

## **144 i Paragon Getting Started**

### **Starter Sets**

- ActiveX, 39
- OI, 39
- text labels, 109
- tool palette, 57
- trends
  - creating, 131
  - formatting, 132
  - pens, 132
- uncompiled file, 24

## **P**

### **Packaging Files**

- Paratnt.TEC, 25
- Paratnt.USR, 25

### **Paragon**

- overview, 18

### **PID faceplate**

- creating, 114

### **Portable File**

- Continuous Strategy (CS), 24

### **Process Blocks, 79**

### **Process I/O (PIO), 20, 33, 45, 47**

- configuration file, 24
- configuring, 47
- creating, 51
- saving, 53
- configuring functions, 35
- data transformation, 47
- driver type, 33
- library file, 24, 36
- configuring, 47
- functions, 49
- saving, 51

## **Q**

### **Quality Tags, 72**

## **R**

### **RAD Tools, 22**

- Borland Delphi, 22
- PowerBuilder, 22
- PowerSoft, 22
- Visual Basic, 22

### **Recipes, 22, 34**

### **Repositioning Objects, 59**

### **Resizing Objects, 58**

### **Runtime, 51, 70, 94, 118, 137**

## **S**

### **Saving**

- Application Manager (AM)
  - application file, 70



Continuous Strategy (CS) file, 87  
Data Manager (DM)  
configuration file, 128  
library file, 128  
Operator Interface (OI) displays, 67, 137  
Process I/O (PIO)  
configuration file, 53  
library file, 51

**Security. See Application Manager (AM)**

**Servers, 19, 20, 21**

**Sibling Display, 116**

**Standard Query Language (SQL)**  
using with Data Manager (DM), 123

**Starting Paragon, 29**

**Stations**

Clients, 18

Servers, 18

**String, 54**

**System Requirements**

OS/2, 11

Windows 95, 11

Windows NT, 11

## **T**

**Tag Name, 30, 34**

element name, 30

function name, 30

process name, 30

subsystem name, 30

tag generation wizard, 34

**TCP/IP, 20**

**Test Drive**

important note, 7

**Trends**

creating a display, 131

formatting, 132

pens, 132

## **U**

**User Assumptions, 7**

## **V**

**Visual Basic, 19**

Paragon 5.3

# Getting Started

Copyright © NemaSoft Inc. 1998. All rights reserved.

NemaSoft, Paragon 500, Paragon 550, Paragon, Paragon 97 and paragon TNT are registered trademarks of NemaSoft. All other trademarks are the property of their respective holders.

Information in this document is subject to change without notice and does not represent a commitment on the part of NemaSoft. No part of this manual may be reproduced or transmitted in any form or by any mean for any purpose without the express written permission of NemaSoft Inc.

Document 5.00.03

55 West Street, Walpole, MA 02081 USA

Tel 508-668-7741 Fax 508-668-7126

# Technical Assistance

Paragon is designed for Application Developers, who are familiar with the technical details of Windows 95, Windows NT, and/or OS/2.

Details on these operating systems are not provided in this manual. For detailed information about Microsoft and IBM operating systems, we suggest you refer to the Reference Manuals for these systems.

If you need assistance using Paragon , please consult this manual, the online Reference Manual, the User's Guide or the Release Notes.

If you are unable to solve a problem after consulting the manuals or online documentation, our technical support staff is available. Feel free to contact us by fax, using the form provided on the following page. This gives us the opportunity to review the problem thoroughly and give you better customer support.

A cover sheet for Customer Service Request is provided on the following pages. Feel free to Photocopy this form and fax it to us, along with any other information you feel is appropriate.

## Customer Service Request

**FAX TO:** NemaSoft, Inc. 55 West Street, Walpole, MA 02081

**ATTN:**

**FAX No:** 508-668-7126 **PHONE:** 508-668-7741

**FROM: DATE:**

**COMPANY:**

**FAX No: PHONE: TIME ZONE:**

**TOTAL PAGES:**

### **HARDWARE CONFIGURATION**

Motherboard Memory Free Disk Space

Display Board & Resolution OS OS Version

Network Type Other Plug-in Cards

### **PARAGON CONFIGURATION**

Paragon Version Patches Applied Serial No.

List Engines Enabled

Options Enabled: Data Manager Continuous Strategy Network Services

Trending Supervisory TCP/IP

History String Blocks NETBIOS

Advanced User Block Limit Recipes

Alarm /Event Process I/O-64 Points DDE

RDB Support 256 Points

Engineering Interface 512 Points

Quick Reports 2048 Points

User Client Link Unlimited

### **PARAGON PROBLEM**

Suspect Subsystem

Suspect Function

Reproducible Yes No If yes, how is the problem reproduced?

### **DETAILED DESCRIPTION**

If the problem include a register dump (i.e., SYS3175),

What was the second line of the Register Information

Window? (e.g., TNTNLS.DLL 0001:00000337)

If the problem generated error or

Error messages, what were they?

Were there any related messages

in the System event Log?

## Contacting US By Phone

If you need to call NemaSoft Technical Support, hours are between 9 AM and 5 PM Eastern Standard Time. When you call, please be in front of your system and have the above information available. If you do not have this information ready, we may be unable to assist you. Our phone number is 508-668-7741.

Please note, your Customer Registration Card must be returned before our support staff can answer your questions. Completion of this card ensures that you can take advantage of Customer Support Services, mailings about software updates, revisions and release notes.