

ASTRA
Quick Start Manual





Manual Revisions

If you contact us in reference to this manual, please include the following name and revision number

Name: Astra Quick Start Manual

Revision: QMAN\ASTRA\Rev.0

We hope that you find this manual informative. If additional information or technical assistance is needed please contact:



Renu Electronics Private Limited

Survey No. 2/6,

Baner Road, Pune – 411 045

Maharashtra, INDIA

Tel: +91 20 729 2840

Fax: +91 20 729 2839

E-mail: support@renuelectronics.com

Website: www.renuelectronics.com

Contents

About This Manual

Chapter 1: Overview

- 1.1 Description:
- 1.2 Features:

Chapter 2: Installation

- 2.1 System Requirement:
- 2.2 ASTRA Shipping Contents:
- 2.3 Installation Procedure for ASTRA:

Chapter 3: Getting Started

- 3.1 Building a Project using ASTRA:
- 3.2 Checklist for building a Project:
- 3.3 Steps for building a Project:
- 3.4 Opening an existing Project
- 3.5 Creating a New Project:
- 3.6 Configuring a Project:
- 3.7 Saving a Project:
- 3.8 Closing a Project:
- 3.9 Quitting a Project:
- 3.10 Running a Project
- 3.11 Using a Run Mode:
- 3.12 Maintan User Accounts
- 3.13 Data Logging:
- 3.14 Working with windows

Chapter 4: Toolbox

- 4.1 Customizing Toolbox:
 - 4.1.1 Drawing Tools
 - 4.1.2 Shape Properties
 - 4.1.3 Alarm / Trend / Polyshapes
 - 4.1.4 Color Tools
 - 4.1.5 Alignment Tools
 - 4.1.6 Text Tools
 - 4.1.7 Editing Tools

Chapter 5: Animation

- 5.1 How to attach Animation?
- 5.2 Types of Animation

Chapter 6: Componants

- 6.1 Making a componant
- 6.2 Editing a componant
- 6.3 Componant Library
- 6.4 Loading Componant
- 6.5 Configuring Componant
- 6.6 Breaking a Componant

Chapter 7: Tag Database

- 7.1 System Tags
-

| | |
|------|--------------------------------------|
| 7.2 | Memory Tags |
| 7.3 | Device Tags |
| 7.4 | Special Tags |
| 7.5 | Adding, Editing or Deleting the Tags |
| 7.6 | Tag Properties |
| 7.7 | Setting the Advanced Attributes |
| 7.8 | Clamp Engineering Values |
| 7.9 | Clamp Raw Values |
| 7.10 | Using Tag Groups |
| 7.11 | Adding Tag Group |
| 7.12 | Editing Tag Group |

Chapter 8: Node Control

| | |
|-----|---|
| 8.1 | Node control during the Initial Scan |
| 8.2 | Node Control while the project is running |
| 8.3 | Setting-up a Node |
| 8.4 | Adding a Node |
| 8.5 | Editing a Node |
| 8.6 | Deleting a Node |

Chapter 9: Alarms

| | |
|------|------------------------------|
| 9.1 | Defining Alarm on a Tag |
| 9.2 | Configuring the Alarm Window |
| 9.3 | General Properties |
| 9.4 | Position Properties |
| 9.5 | Format Properties |
| 9.6 | Tag Group Properties |
| 9.7 | Color Properties |
| 9.8 | Setting Alarm Printer |
| 9.9 | Port Selection |
| 9.10 | Format Tab |
| 9.11 | Logging of Alarms |

Chapter 10: Data Logger and Reports

| | |
|------|---|
| 10.1 | Setting the Data Logger |
| 10.2 | Configuring Reports |
| 10.3 | Examples of Report Generation |
| 10.4 | Report Generation through Dynamic Data Exchange (DDE) |

Chapter 11: Trends

| | |
|--------|-----------------------------|
| 11.1 | Real-time Trend |
| 11.2 | History Trend |
| 11.3 | How to draw Trend |
| 11.4 | Properties Related to Trend |
| 11.5 | Trend Properties |
| 11.5.1 | General Properties |
| 11.5.2 | Position Properties |
| 11.5.3 | Value Scale Properties |
| 11.5.4 | Time Scale Properties |
| 11.5.5 | Color Properties |

Chapter 12: Scripts

| | |
|-----------------------------|---------------------------------------|
| 12.1 | Script Level |
| 12.1.1 | Project Level |
| 12.1.2 | Window Level |
| 12.1.3 | Component Level |
| 12.1.4 | Object Level |
| 12.1.5 | Tag Level |
| 12.2 | Script Command |
| 12.2.1 | IF-THEN-ELSE Statement: |
| 12.2.2 | WHILE DO Statement: |
| 12.2.3 | Arithmetic Operators: |
| 12.2.4 | Conditional Operators: |
| 12.2.5 | Assignment Operator: |
| 12.2.6 | Bitwise (Logical) Operators: |
| 12.2.7 | Boolean Functions: |
| 12.2.8 | String Concatenation Operator: |
| 12.2.9 | String Operators: |
| 12.2.10 | General Script Commands: |
| 12.2.11 | Script for Recipe: |
| 12.2.12 | Properties Related to Simple Objects: |
| 12.2.13 | Script for Reporting: |
| 12.2.14 | Some Examples of Report Generation: |
| Chapter 13: Recipe | |
| 13.1 | Creating Recipe File |
| 13.2 | Using the Recipe File |
| Chapter 14: Security | |
| 14.1 | Configuration Mode Security |
| 14.2 | Run Mode Security |



About This Manual

Purpose:

The purpose of **Quick Start Manual** is to provide complete information for developing and implementing the process application. It also provides information on how to control and monitor the Industrial process.

Audience:

The Quick Start Manual is written for:

- System Engineers responsible for designing and configuring the application.
- System Administrators responsible for monitoring and controlling the process and maintaining the security.



Chapter 1: Overview

1.1 Description:

ASTRA is Supervisory Control and Data Acquisition (SCADA) software. ASTRA provides effective management, monitoring and control of industrial process. It is a Man-Machine-Interface (MMI) communicating with Loop Controllers, Programmable Logic Controllers (PLCs), I/O devices and other field devices.

ASTRA has two constituents: **Configure Mode** and **Run Mode**. The Configure Mode allows a user to configure the process. To configure a process, a user need to :

- Start a project with basic settings
- Create a database
- Draw a mimic
- Animate a mimic
- Insert data monitoring tools

Run Mode involves the actual data acquisition, processing, monitoring and controlling.

1.2 Features:

ASTRA provides special features like Trends, Recipe and Scripts to control and monitor data. Trend and Alarm windows are useful for on-line data monitoring. Reports and Data Loggers can be used for analyzing process data.

Other than the easy to handle tools for basic process configuration ASTRA also gives some advanced features like Components, Node Control, Tag Groups and Recipe Management.

Trends:

ASTRA allows a user to configure both the real-time and the history trends. The horizontal axis in a trend denotes the time-span. User can attach multiple tags to the vertical axis using pens.

Recipe:

ASTRA provides the feature of a recipe management. Recipe is a set of process variables with definite values. Recipes need to be defined in the INI file similar to the project.ini file. A user can load a recipe in the running process or create and save a new recipe at run-time. Using this feature a user can change the values of an entire set of variables, as a batch process, at run-time.

Scripts:

ASTRA allows a user to use powerful English-like commands to customize the process configuration. Scripts are sequential set of commands and procedures that are executed at run time. A user can write scripts for taking control actions, defining complex interlocks and simulating the complete control logic.

Alarm:

For monitoring data, a user can define deviation alarms or high-low alarms on the tags. Deviation alarms interpret value changes in tags using the specified percentage. Whereas, the high-low alarms interpret value changes in tags using the specified range. Alarms can also be defined on the Boolean tags. A user can also specify the severity level for alarms. A user has an option to print and save the alarms in a specified format. ASTRA also allows a user to log the alarm messages in text files.

Drawing and Animating the Mimic:

ASTRA allows a user to draw the graphical representation of the process using various tools. Once you draw objects, you can animate them. Twelve different Animation types are available in ASTRA. A user can animate individual shapes or a group of shapes using these animations to display the dynamic field condition.



Data Logger:

ASTRA allows a user to keep a log of process variables using the built-in data logger. This logger saves data in log files. These files can be used for off-line applications such as MIS reporting. Data can also be displayed in a graphical format for Historical Trends.

Tag Groups:

ASTRA allows the user to group logical tags. Tags group give the facility to scan the set of tags in one scan to ensure all related tags are updated simultaneously. A user can also make the scanning of tags ON or OFF. This helps in avoiding the scanning of tags that need not be scanned at all. Tags groups can also be used for improving the readability of alarms.

Node Control:

ASTRA allows the user to create certain special tags while adding a node. These tags facilitate to:

- See the status of a node at any point of time
- Detect the failure of a node automatically
- Continue the processes unrelated to the node that has failed
- Stop or start a node if required

Components:

ASTRA allows user to draw a number of standard symbols, like bottles, sliders, motors, etc. User can use the Symbol Library to store symbols. Instead of drawing these symbols each time, insert ready symbols from the Symbol Library. It is also possible to make these symbols intelligent by saving them as components. Components can be loaded from the component library. User can edit some of their parameters like text on objects, tags and tag values to make them process specific.

Reports:

ASTRA allows generating reports at run-time by configuring them in Configure Mode. It supports generation of text as well as graphical reports. User can send the reports to a file or to a printer. As user can specify the report generating options using scripts, reports can be generated without using the operator interface.

Security:

ASTRA allows having security at configuration time as well as at run time. To ensure the security, process configuration can be protected with the use of password. While configuring a project, user can set the access levels from 0 to 255 for different run-time items and activities. Run time security can be observed by maintaining the user list. Users can be assigned the access levels and passwords. The login validates password and allows only the valid users to run the process.



Chapter 2: Installation

2.1 System Requirement:

Recommended Hardware Requirements:

| | | |
|--------------------|---|--|
| CPU | : | IBM / IBM compatible PC Pentium, 133 MHz or higher |
| RAM | : | 32 MB or more |
| Hard disk | : | 100 MB min. free |
| Monitor | : | SVGA Color |
| Mouse | : | Required |
| Communication Port | : | RS232C serial port (Project specific) |
| Operating System | : | Microsoft's Windows 95/98, Windows 2000/XP. |

2.2 ASTRA Shipping Contents:

Complete Package:

- ASTRA SCADA Software Installation CD with device driver
- ASTRA Hardware lock (One)
- User Manual for ASTRA (One)
- Quick Start Manual for ASTRA (One)
- Release notes for ASTRA (One for each driver device driver)

Demonstration Package:

- ASTRA SCADA Software (One Installation CD)
- Quick Start Manual for ASTRA (One)

2.3 Installation Procedure for ASTRA:

- Insert CD. It is auto run, otherwise select **Setup.exe**.
- The interactive dialogue boxes of ASTRA setup will appear. Follow the sequence. After complete installation of ASTRA, a "Setup Successful" message box will appear on the screen. The selected options are now installed in ASTRA directory and ASTRA group is created. The ASTRA group consists of following programs:

| | |
|-----------------|---|
| ASTRA Configure | ASTRA configuration mode for building and configuring a project |
| ASTRA Run | Run-time executable, which runs the pre-configured project |
| ASTRA Help | On-line help on ASTRA |
| ASTRA Run setup | Utility to setup |



Chapter 3: Getting Started

3.1 Building the Project using ASTRA:

There are two options to begin with the project:

- Start creating the new window and the mimic screens. Then, define I/O Tags
- Define the I/O tags and then create the mimic screens

Note: This manual follows the first option.

3.2 Checklist for building the Project:

Before creating the project, user should have following details:

- The input and output list of all process signals, which need to communicate with ASTRA along with the address mapping.
- The relation between the device address and the addressing scheme in ASTRA device driver as per the release notes or as per the help provided in the device driver program.
- The details of the process graphics / mimics to be generated and the control drawings to understand the interlocks involved in the process.
- The details and the formats for the report generation.

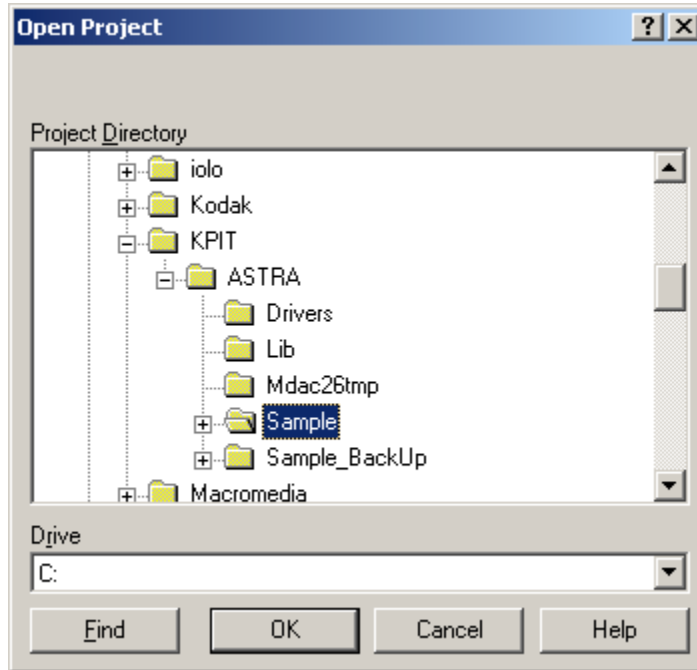
3.3 Steps for building the Project:

- Click **start** and select **Programs | ASTRA** menu option
- Select **ASTRA Configuration** program

The **Open Project** dialogue box with the list of all files, will appear on the screen.



3.4 Opening an existing Project



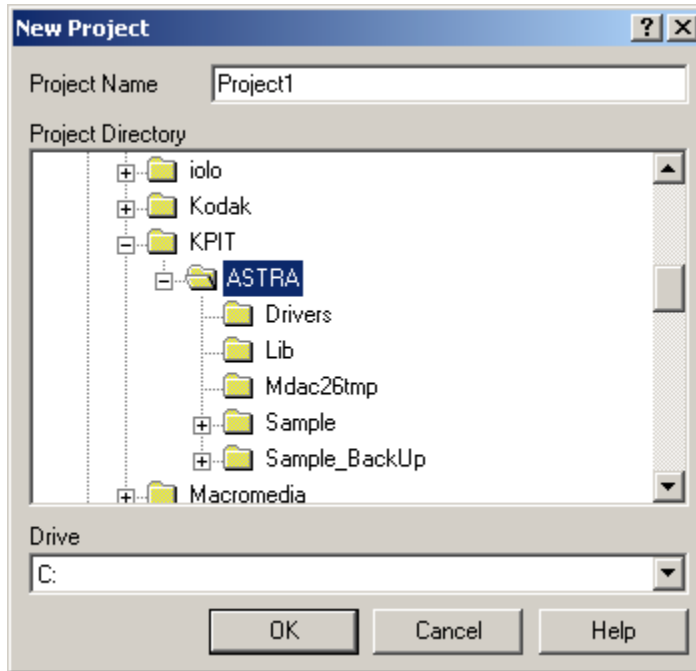
| | |
|--------------------------|--|
| Project Directory | Displays the list of files available on the drive. |
| Drive | Drop down list of the available drives. |
| Find | Search for a project. The first project is highlighted from the selected directory. Continued clicking will take you to the desired project. If no project is found in the directory, a message box will appear prompting, "Could not find any project in the directory specified". Click OK to return to the Open Project dialogue box |



3.5 Creating a New Project:

For creating a new project, Click Cancel.

- Click the File | **New Project** menu option. The **New Project** dialogue box is now displayed
- Select the project directory and enter the new file name for the project
- Click **OK**



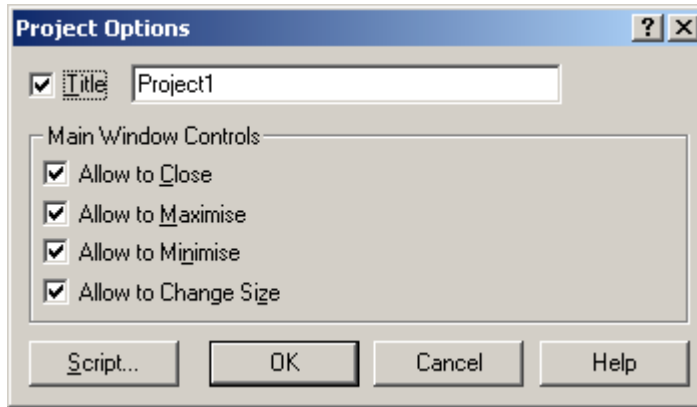
| | |
|--------------------------|---|
| Project Name | Enter the name of the project. (Maximum 8 characters). A new directory with this name will be created |
| Project Directory | Select the path where new project directory will be created |
| Drive | Select the drive on which you want to store your project files |



3.6 Configuring a Project:

ASTRA project options can be set at any time after creating the project. Project title, Main window controls and Start up options can be set in the project options. For writing the project level scripts, use the Script Editor option. To set the project configuration, click the **File | Configure** option. The **Project Options** dialogue box appears.

Warning: A deleting window is not reversible.



| | |
|-----------------------------|---|
| Title | Check this box to assign a title to the project. If the Title box is checked then at run-time the title bar is displayed for main window. By default this box is checked and the default title is “Project1” |
| Main Window Controls | The following controls are available for the main window at run-time |
| Allow to Close | Check this box for allowing the operator to close the project at run-time. If you clear this box, run-time user will not be able to close the main window by clicking the Close control button or by pressing ALT+F4 . By default is box is checked. If the run-time menu option of File Exit is made inaccessible for the operator using Menu Access Control, the operator cannot close the project, even if this box is checked |
| Allow to Maximize | Check this check-box for allowing the operator to maximize the project main window at run-time. If you clear the box, run-time user will not be able to maximize the window by clicking the Maximize control button or by using the control menu. By default this box is checked. |
| Allow to Minimize | Check this box for allowing the operator to minimize the main window at run-time. Click the Minimize control button OR use the control menu to minimize the main menu. By default this box is checked. |
| Allow to change size | Check this box for allowing the operator to change the window size. If this box is checked, the operator can change the size of window by dragging any of the four corners OR by using the control menu at run-time. By default this box is checked. |
| Script | Invokes the Script Editor for writing project level scripts |



3.7 Saving a Project:

Click **File | Save Project** menu option or Press **F7** to save the project. ASTRA saves all windows of the project along with the configuration information.

ASTRA also saves the project while closing or quitting the project. If you close or quit the project without saving the project using the Save option, ASTRA prompts that the contents have changed. Press **“Yes”** to save the changes and close or quit the project. Press **“No”** to close or quit the project without saving the changes. Press **“Cancel”** to cancel the operation.

3.8 Closing a Project:

Click the **File | Close Project** option or Press **F8** to close the project. ASTRA prompts to save the changes in tag database, windows and configure information.

After confirmation, ASTRA automatically compiles the project. Error messages are flashed, if any, during this process. Click **“OK”** to ignore the compilation error messages and close the project. Click **“Cancel”** to cancel the closing operation.

3.9 Quitting a Project:

Click the **File | Quit Project** menu option to quit the current project without compiling and return to the Configuration Mode Project menu. ASTRA prompts to save the changes before quitting the project.

3.10 Running a Project

A user can run the project either from the Configuration Mode or directly from the ASTRA Run file in **Program files | ASTRA** menu. To run a project from the configuration mode, Click the **File | Run** menu option or Press **F9** key.

ASTRA compiles the project. If the compilation is done successfully or if the errors are ignored, ASTRA switches over from the Configuration Mode to the Run mode.



3.11 Using a Run Mode:

Following are the options available for an user in the run mode:

Login:

Run Mode is provided with Login and Logout facility for security purposes. (For details please refer chapter 14, "Security") Whenever you open the project, you need to login, in order to access the controls defined in Configure Mode. Click the **Users | Login** menu option to open the Login dialogue box.

The screenshot shows a standard Windows-style dialog box titled "Login". It has a blue title bar with a close button (X) on the right. The main area is light gray and contains two text input fields. The first field is labeled "User Name" and contains the text "Mandar". The second field is labeled "Password" and contains a series of "x" characters to mask the password. Below the input fields are three buttons: "OK", "Cancel", and "Help".

| | |
|------------------|---|
| User Name | Enter your name as listed in the Users' List. |
| Password | Enter the password as configured by the user. |

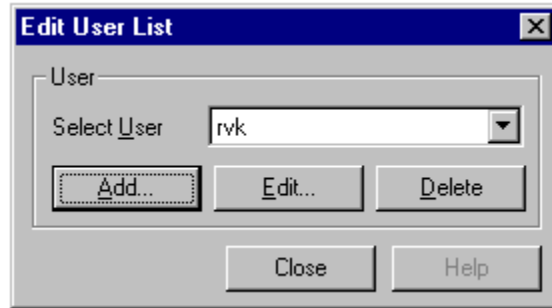
Note : If the user login is incorrect, ASTRA displays a message, "Invalid User name or Password. Please try again".

Logout:

Click **Users | Log Out** to removes your login from the project. It closes all the windows accessed by you. If you are logging out because of some other user's login, ASTRA closes those windows that are accessible only to you. If any of the windows is accessible to the user who is logging in, that window remains open.



3.12 Maintain User Accounts

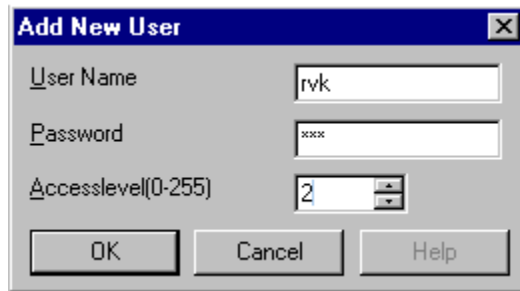


A user can maintain the Users' List in order to implement the security measures. Every user has to have an entry in the Users' List. While making this entry a user can assign the password and the access level for other users. Whenever the user login, ASTRA validates the name, password and access level of the user.

To maintain the Users' List, click the **Users | Edit User List** menu option. The Edit User List dialogue box appears. A user can access this facility only to maintain users having lower access level than yours.

Adding an User:

To add a new user, click the **Users | Edit User List** menu option. The Edit User List dialogue box appears on the screen. From the Edit User List dialogue box click on the Add button. The **Add New User** dialogue box appears on the screen.

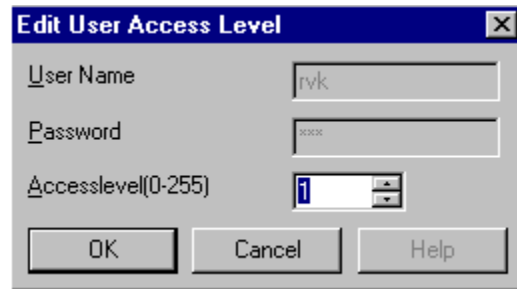


| | |
|---------------------|--|
| User Name | Enter the string identifier for the operator. |
| Password | Configure the password for the user. |
| Access Level | Assign the access level to the user. You can assign the access level that is lower than your own access level. |



Editing User:

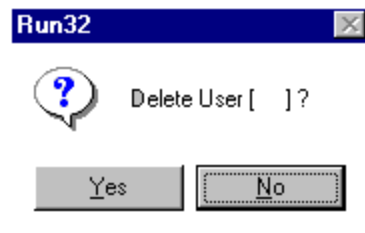
To edit the access level of existing user, click the **Users | Edit User List** menu option. Edit User List dialogue box appears on the screen. From Edit User List dialogue box select the user and click on the **Edit** button. **Edit User Access Level** dialogue box appears. You can change the access level of the existing user with access level lower than yours but you cannot change either the name or the password of the user.



| | |
|---------------------|---|
| Access Level | Assign the new access level to the user. You can assign the access level that is lower than your own access level |
|---------------------|---|

Deleting User:

To delete an existing user, click the **Users | Edit User List** menu option. Edit User List dialogue box appears on the screen. From the Edit User List dialogue box select the user and click on the **Delete** button.



ASTRA prompts you to confirm the delete operation. Press **“Yes”** to delete the user. Press **“No”** to cancel the delete operation.

Changing Password:

You can change your own password with **Change Password** menu option.

- Open project and login with your old password.
- Click the **Users | Change Password** menu option. **Changing Password** dialogue box appears.
- Enter old password and press **OK**.
- Now, enter new password and Press **OK**.
- Re-enter new password to confirm the changed password.

Pressing **OK** returns you to the main menu. User need to save the project to save the changes permanently.



3.13 Data Logging:

Data Logging options are set in the Configuration Mode. User can start or stop data logging from “Run Mode”.

When the tag is defined in Configuration Mode, Log Data option has to be enabled for it. Data will not be logged for the tag if its Log Data option is not checked.

Event Logging:

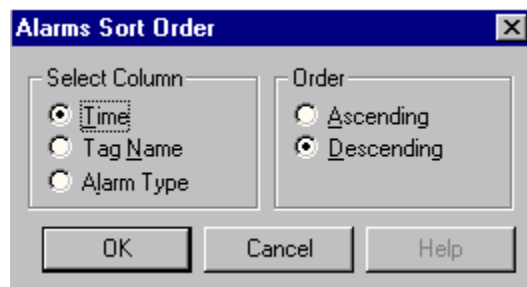
ASTRA logs all the important events occurring at the run time. Click **Windows | Show Events** menu option to open the Event Logger window.

Whenever the Event Logger window appends the messages to a list, a new event occurs. To delete the messages from the Event Logger, click the **File | Clear** menu option.

ASTRA logs the message with date and time stamping. ASTRA also logs these events in a text file. Events are logged in the **evt.log** file. This file is generated and stored in the project directory. You can start / stop event logging using the **Logger** menu options from the Event Logger Window. When 3000 events are logged in **evt.log** file, ASTRA creates a backup of this file. The backup file is named as **evtbk.log**. If ASTRA is unable to write new messages to the file, ASTRA creates the **evt1.log** file and up to **evt20.log** file.

Alarm Sorting and Formatting:

Project can be configured allowing you to sort and / or format the alarm messages at run time. This can be done using the SORT and FORMAT script commands in the Configure Mode. For sorting the alarm messages, the Alarm Sort Order dialog box appears at run time.



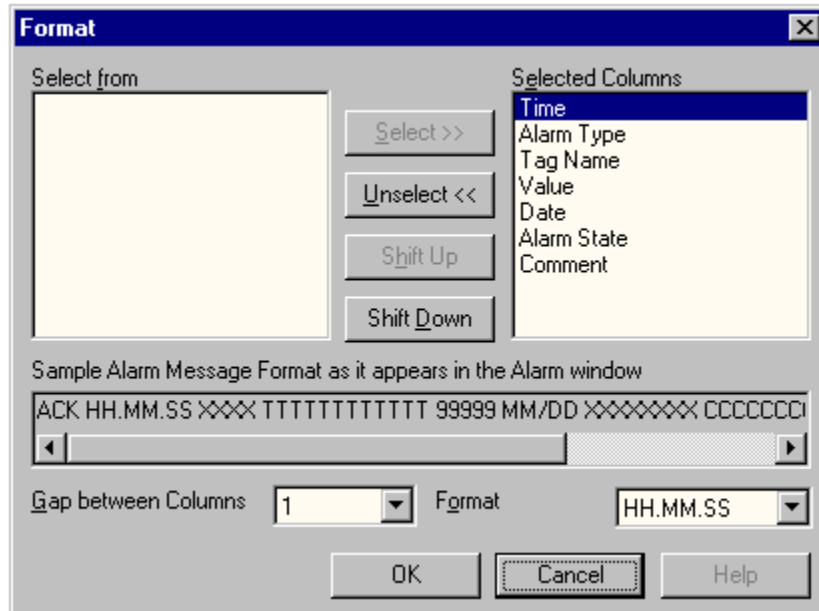
| | |
|----------------------|---|
| Select Column | Sort alarms using one of the following fields. By default, the alarm messages are sorted Time , Tag Name , Alarm Type |
| Order | Select either the Ascending or the Descending order to sort alarm messages by the selected field |



For formatting the alarm messages, the Alarm Message Format dialog box appears at run time on giving the FORMAT command in script.

Any alarm can have following fields in its message:

- Date : Date on which the alarm is generated.
- Time : Time when the alarm becomes active.
- Tag Name : Name of the tag on which the alarm is defined.
- Value : Value of the tag at the current time.
- Alarm Type : Alarm type defined for tag: Highlow, Deviation or. None.
Alarm State Acknowledged and active, Unacknowledged and active, Unacknowledged and inactive
- Comment : Alarm text specified while defining alarm





By setting the format you can select fields, space between them and their sequence. You can set different format for displaying and printing alarms.

| | |
|-----------------------|--|
| Select from | This list gives the fields that may be selected for the message. Highlight the desired field(s), and choose the Add button. The field is added to the Selected Columns list. |
| Select Columns | This list gives the fields that are selected and will appear in the alarm message. To deselect the unwanted field(s) highlight them and choose the Delete button. |
| Select | Adds the highlighted field from the Select from list to the Selected Column. |
| Unselect | Deletes the highlighted field from the Selected Column and lists it in the Select from column. |
| Up | This button is enabled only if the field from the Selected Column is highlighted and if the field is not the first one in the column. Click on this button to move the field up by one position. The sequence of the fields in this column decides the sequence of the fields in the alarm message. |
| Down | This button is enabled only if the field from the Selected Column is highlighted and if the field is not the last one in the column. Click on this button to move the field down by one position. The sequence of the fields in this column decides the sequence of the fields in the alarm message. |
| Gap | Enter or select the space between two columns. You can have the space within the range 0 to 9. |
| Format | Enter the format for the highlighted field from the Selected Column. For the Date and Time fields you can select the format from the combo box. For other fields you need to enter number of characters for the format. |

Alarm Logging:

ASTRA logs alarm messages to a text file. After running the process, if any alarm is generated, it is logged in the *alm.log* file. By default alarm logging is "On". You can stop alarm logging using the **Logger | Stop Alarm Logging** menu option. To start the alarm logging, use the **Logger | Start Alarm Logging** menu option.

Using Data Entry Dialogue Box:

If the Data Entry animation is defined on any object, click on the object or press the shortcut key to open a dialogue box for data entry.

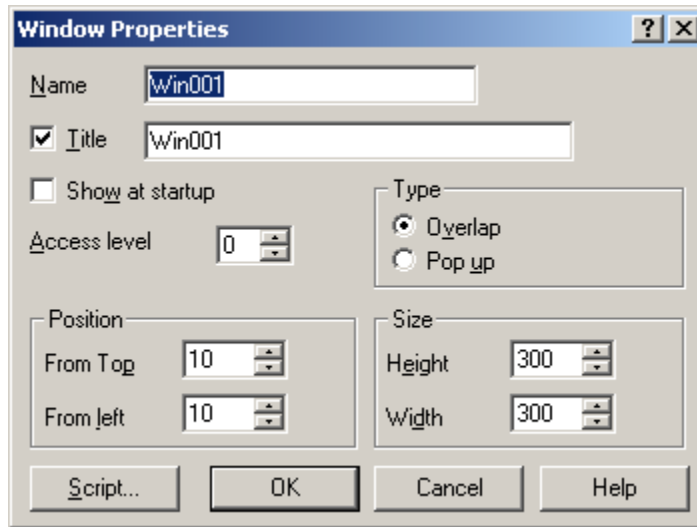
Enter the value in the Fill-in Field. The value is written to the attached tag if you press the **OK** button. The previous value is retained if you click the **Cancel** button and return the calling window.



3.14 Working with windows

Creating a New window:

Once the new project name is entered, a **Window Properties** dialogue box appears. User can set the window properties from this dialogue box. After defining the required properties, click **OK**.



| | |
|------------------------|--|
| Window Name | Enter name for the window. Default name is “Winnnn” where ‘nnn’ is the number given by ASTRA. For example, Win001, Win002 etc. |
| Window Title | Enter the title for the window to be displayed on the title bar. The default title is same as the default window name. You can check the ‘Title’ box, to display the title bar in the Configuration Mode as well as in the Run Mode. |
| Show at Startup | Display the window immediately on opening the project in the Run Mode. |
| Access Level | Choose or enter the access level between 0 and 255 for the window. You cannot access the window in Run Mode, if you enter the access level less than the specified level. |
| Type | Allows you to change the window appearance to Overlap type and Pop-up type. By default the Overlap option is selected. The Overlap type allows you to activate any of the open windows at run-time. The Pop-up type window always overlaps any other open window at run-time. You can move the pop-up window any where on the screen. To move the Pop-up window, click on the title bar and drag the window to the desired position. |



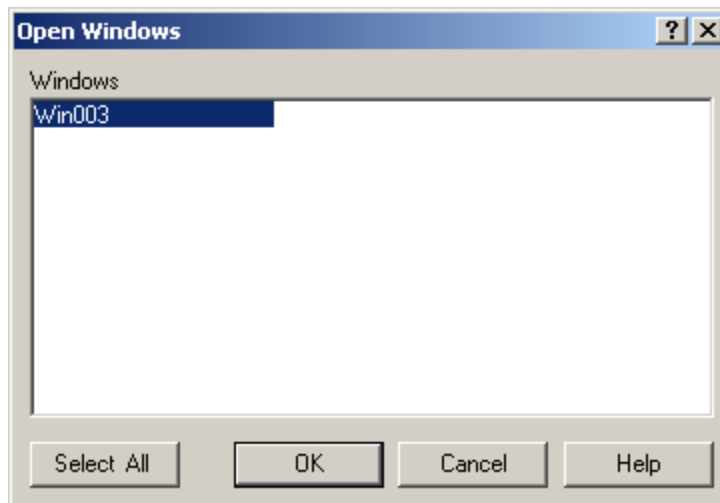
| | |
|-------------------|--|
| Dimensions | Allows you to specify position and size of the window. The unit of measurement is Pixel. Enter the following fields for specifying the Dimensions: |
| From Top | Distance between top edges of the window and the desktop. |
| From Left | Distance between left edges of the window and the desktop |
| Height | Height of the window in pixels. You can enter up to 5 digits. However, if the height is greater than the monitor resolution, ASTRA gives the message accordingly |

Note: You cannot enter the negative values for the Height and Width of the window. If you enter the negative value, ASTRA prompts you to enter a valid value depending upon the window resolutions. Once the window is created in the project window you can always change its position by dragging from the title bar. You can also change window dimensions by dragging the mouse from the edges. Any change in the window dimension will not affect the size of the graphic.

| | |
|---------------|---|
| Width | Width of the window in pixels. You can enter up to 5 digits. However, if the width is greater than the monitor resolution, ASTRA gives the message accordingly. |
| Script | Displays the Script Editor to write window level scripts. |

Opening Window:

To open the existing windows in a project click **File | Open Windows** menu option OR Press **F3**. The **Open Windows** dialogue box is displayed. This menu option is enabled only if there is any closed window in the project.



| | |
|--------------------|--|
| Window List | Lists the existing windows that are not yet opened. Select the windows to be opened from the list. |
| Select All | Selects all windows from the list. |



Saving Window:

To save the changes made in the windows click the **File | Save Windows** menu option or Press **F2**. The **Save Windows** dialogue box appears. The other steps remain the same as above for 'Opening a windows' section.

Closing Window:

To close the windows, click the **File | Close Windows** menu option. The **Close Windows** dialogue box appears. Select the windows to be closed and click **OK**.

Deleting Windows:

To delete unwanted windows, choose the **File | Delete Windows** menu option. The **Delete Windows** dialogue box appears. Select the windows to be deleted and click **OK**.

Editing the Window Properties:

The properties of the existing windows also can be modified for an existing project. Click the **Edit | Properties** option OR Double-click in the window OR Press **F4**. The **Windows Properties** dialogue box appears. For more information on Windows Properties, refer page 19, section "Creating new window".



Chapter 4: Toolbox

ASTRA provides a Toolbox that has the tools to edit or draw the objects on the screen. Tools in the toolbox can be selected by a mouse click on the tool object. The tool tip is displayed when the cursor is positioned on the tool. The Toolbox disappears if the Tools | Toolbox menu option is deactivated. At least one window of the project should be open to enable the toolbox and to the tools.

Note: If a project does not have any window opened, the tool box is visible but is disabled. Only the tool tips are enabled. Open at least one window to enable the tool box and use the tools.

4.1 Customizing Toolbox:











To optimize the drawing space you can easily customize the tool box. The tool box can be moved to any convenient position by holding and dragging the title bar of the tool box.

You can also hide the tools that are not required for your current task. Use the first row of tool box for this purpose as explained below.

This row of the toolbox has toggle buttons to show/hide the rest of the rows in the tool box. By default all buttons are depressed displaying all rows of the tool box. By toggling any of the buttons OFF, its corresponding row will be removed from the tool box. The following table describes each button of the first row and shows the corresponding row.

4.1.1 Drawing Tools









To draw any object/shape using the toolbox, select the respective tool with a mouse click. Place the Pen shaped cursor in on the position and hold down the left left mouse button and drag the cursor to the required size. Different drawing tools available are:

| | |
|---|--|
|  | Cursor tool and is useful for selection of a new tool from the tool box. |
|  | Draws a line. |
|  | Draws a rectangle/square. |
|  | Draws a rounded rectangle/square. |
|  | Draws an ellipse or a circle. |
|  | Draws a button. |
|  | Inserts a bitmap picture. |
|  | Inserts a text object. |







4.1.2 Shape Properties

To assign the different shape properties to any object, select the object and click the required shape property tool. The properties that can be assigned to different shapes are:

| | |
|---|---|
|  | Attributes the single line thickness to the selected object(s). |
|  | Attributes the double line thickness to the selected object(s). |
|  | Attributes the triple line thickness to the selected object(s). |
|  | Attributes the thickest line thickness to the selected object(s). |
|  | Attributes the fill property to the selected object(s). |
|  | Removes the fill attribute of the selected object(s). |
|  | Decreases the Z level of the selected object(s) by 1. |
|  | Increases the Z level of the selected object(s) by 1. |





4.1.3 Alarm / Trend / Polyshapes

To draw any object / shape using the toolbox, select the respective tool with a mouse click. Place the Pen shaped cursor in on the position and hold down the left mouse button and drag the cursor to the required size. Different drawing tools available are:

| | |
|---|-------------------------|
|  | Draws the alarm object. |
|  | Draws the trend object. |
|  | Draws a polyline. |
|  | Draws a polygon. |









4.1.4 Color Tools

ASTRA has a color palette with 256 color options. To assign the color to any object / Shape, select the object and click on the corresponding color tool. ASTRA will now display the color palette. Select the required color and click **OK**.

| | |
|---|---|
|  | Displays the color palette for selecting the line color for the object. |
|  | Displays the color palette for selecting the fill color for the object. |
|  | Displays the color palette for selecting the color for the window. |
|  | Displays the color palette for selecting the color for the text. |






4.1.5 Alignment Tools

Select the group of objects whose alignment is to be changed and click on the alignment tool.

| | |
|---|--|
|  | Aligns the selected objects with the left edge of the leftmost object in the selection. |
|  | Aligns the selected objects with the right edge of the rightmost object in the selection. |
|  | Aligns the vertical centre of all the objects with the vertical axis of the selection. |
|  | Aligns the selected objects with the top edge of the topmost object in the selection. |
|  | Aligns the selected objects with the bottom edge of the lowest object in the selection. |
|  | Aligns the horizontal centre of objects with the horizontal axis of the selection. |
|  | Equalizes the width of the selected objects. Choosing this tool with the SHIFT key pressed, equalizes the width of all the selected objects with the smallest object amongst them. Otherwise it equalizes the width with the largest object. |
|  | Equalizes the height of the selected objects. Choosing this tool with the SHIFT key pressed, equalizes the height of all the selected objects with the smallest object amongst them. Otherwise it equalizes the height with the largest object. |

4.1.6 Text Tools









Select along with the text object or select the text whose property is to be changed and click on the text tool. Text tools are related to the attributes of text objects.

| | |
|---|--|
|  | Attributes the bold typeface to the selected text object(s). |
|  | Makes the selected text object(s) underlined and sets it as default style. |
|  | Attributes the italic typeface to the selected text object(s). |
|  | Attributes the Strikethrough typeface to the selected text object(s) |
|  | Gives the Font dialogue dialog box to assign the font for the selected text object(s). |



4.1.7 Editing Tools

Select the object / objects to be edited and click on the editing tool.

| | |
|---|--|
|  | Places the selected objects at equal horizontal distance from each other. |
|  | Places the selected objects at equal vertical distance from each other. |
|  | Converts the selected objects into a single group object. |
|  | Breaks the selected group in to individual objects. |
|  | Duplicates the selected object(s). |
|  | Cuts the selected object(s) and pastes on the ASTRA clipboard. |
|  | Copies the selected object(s) and pastes on the ASTRA clipboard. |
|  | Pastes the object(s) from the ASTRA clipboard in the active window or the library. |

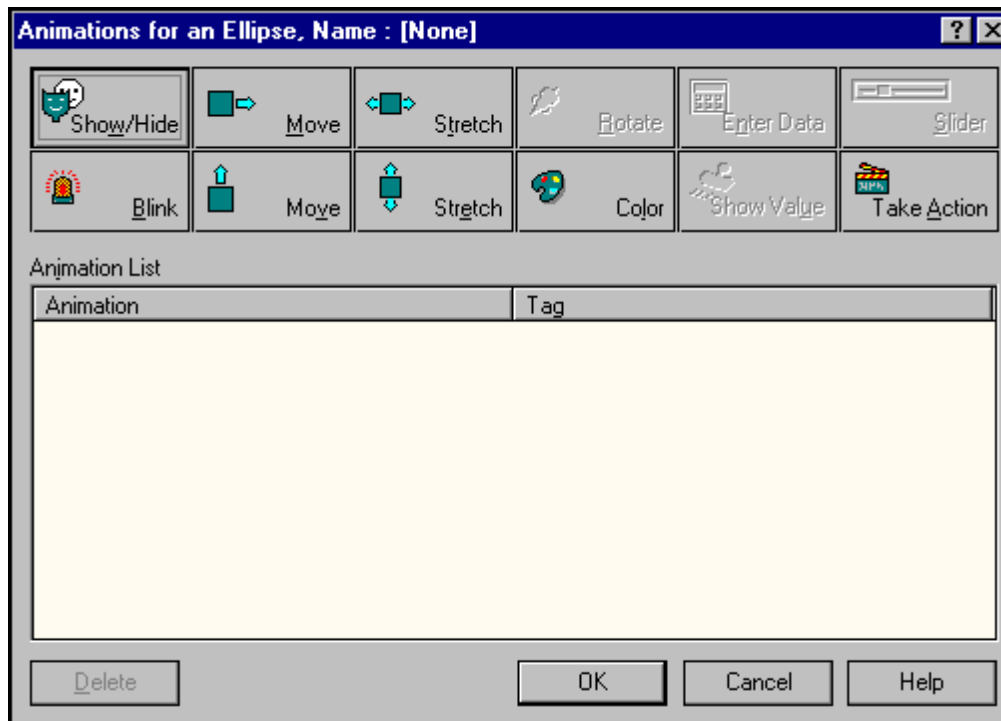


Chapter 5: Animation

Animation is a tool which can be attached to any object / group of objects. The types of available animation depend on the animation defined on any of the group member.

5.1 How to attach Animation?

- 1 Select the object or a group of objects to which animation is to be attached.
- 2 Click **Edit | Animation** menu option or Click the right mouse button or Press **F5 KEY**. **Animation** dialogue box will be displayed.




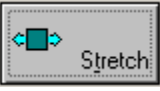



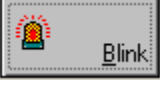





- 1 Select the required animation. The dialogue box for the selected animation will appear.
- 2 Enter the details and click **OK**.

Animation has to be attached to a tag. When you select a particular animation on some object you need to give the tag name. You can enter the tag that is compatible with the selected animation type.



5.2 Types of Animation

There are 12 different Animation available in ASTRA. These are listed below.

| | |
|--|--|
|  Show/Hide | Show / Hide To display or to hide the make the object from the window depending on the current value of the Process parameter. It is applicable to all objects |
|  Stretch | Stretch Horizontal Allows you to change width of object in proportion with the tag as the data changes. It is applicable to all objects except line and text. |
|  Rotate | Rotate Allows you to display the rotation of the line at run time as the tag data changes. It is applicable only for line object. |
|  Enter Data | Enter Data Allows you to Enter or assign a write value (Typically Set point) to the parameter attached. It is applicable to all objects except line and ellipse object. |
|  Slider | Slider A ready made tool, allows you to display and to make the data entry easy without the help of numerical Key pad. It is applicable for all objects except line and ellipse object. |
|  Blink | Blink To blink the object (Invisible or change the color) depending on the value of the Process parameter. It is applicable to all objects. |
|  Move | Move Horizontal To display the Linear Horizontal movement of the object proportional to the value of the field parameter. It is applicable to all objects. |
|  Stretch | Stretch Vertical To change the height of object in proportion with the tag data change. It is applicable to all object except line and text. |
|  Color | Color Allows you to change the color of an object or to fill color in to the object as the process parameter value enters or exceeds the pre-set limit. It is applicable to all objects except bitmap. |
|  Show Value | Show Value Allows you to display the current value of a tag or a field parameter in the text format It is applicable only for button and text object. |
|  Take Action | Take Action Allows you to define the logic and to configure the script to be fired on certain action. It is applicable for all objects except line object. |



Chapter 6: Components

Component is a user configurable group of objects. It is associated with the object properties to facilitate the multiple usage of the objects created for a mimic. It can be stored in the Component libraries.

These components can be reloaded in the same or in different projects. Component can also be edited to suit its properties as required in new project.

Component is an advanced feature provided by ASTRA, apart from the different tools like duplicate, cut-paste, copy-paste.

Following example explains the importance of making the component.

Example:

Consider an example of a slider created in ASTRA.

To create a slider, you can use the basic objects shapes (like line, rectangle and button etc.) and also the animations. On some of these shapes, if you are duplicating or cut pasting same symbol at different places, using available methods of duplicating, cut-pasting, copy-pasting or loading from library, all the properties including tag names and animations get duplicated. Hence, the properties of these objects cannot be directly extracted or edited. Properties from these duplicated or pasted objects have to be manually changed. You can change certain parameters like tag name, values, etc. in every relevant property. Instead of a group, if a component is formed with these objects, it allows you to configure the properties without breaking the group of objects. Once these properties are configured, ASTRA changes all the relevant properties of that component automatically.

You can save the component in a component library. It can be re-used whenever required by loading it. When you make a component, certain animation parameters are extracted. In our example, the tag attached to slider animation, its low and high values and its access level are extracted.

You can also delete, duplicate cut-paste or copy-paste the component like any other object.

6.1 Making a component

- 1 You can make the component by selecting object(s) and / or groups. Select the objects using which a component is to be formed.
- 2 Open the **Edit | Make Component** menu option.

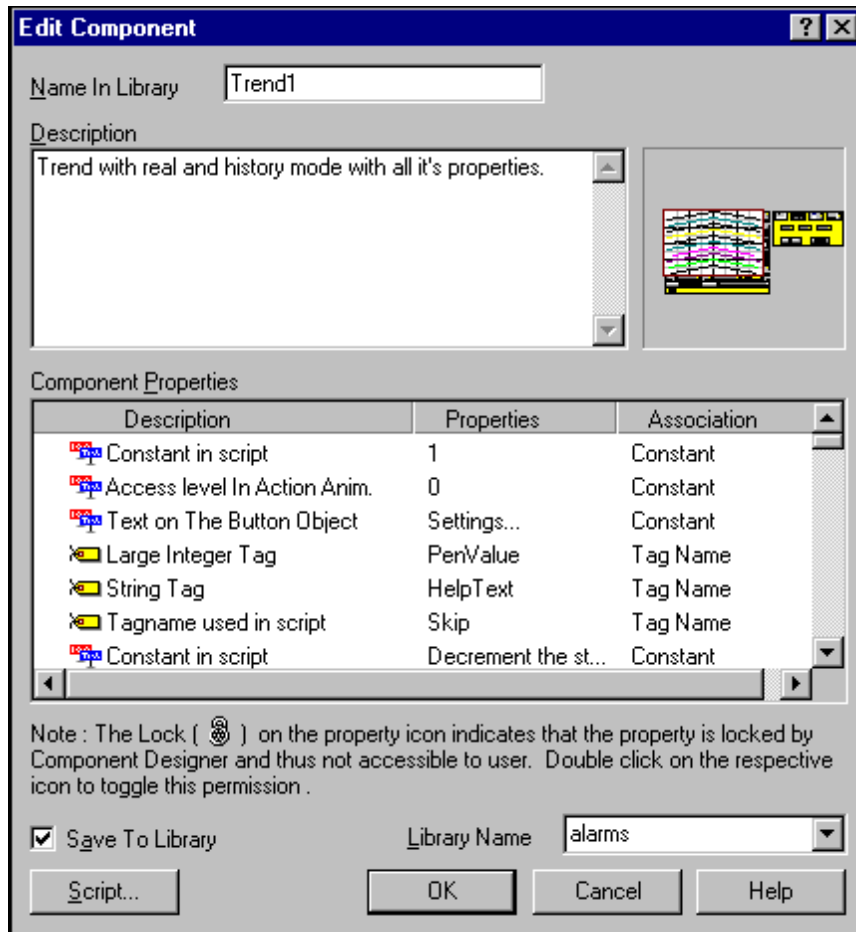
The prompt appears that the “**Component is made successfully**” and whether you want to edit the design. Press “**Yes**” to open **Edit Component** dialog box.



6.2 Editing a component

- 1 You can edit the details of component you have made or loaded. After making, Select the component to be edited.
- 2 Open the **Edit | Component** menu option or Press “ the Yes “ button on the prompt for editing component design.

The dialog box displays the icon image of the component objects you have selected.





| | |
|---|---|
| Component Name in Library | Enter the name for component. ASTRA saves the component with this name in component library, if you choose to save. You cannot use any other special character, except the underscore, in name. You can use alphanumeric characters in name. However, you have to start the name with alphabets. |
| Description Component Properties | Enter the description for component. A list of extracted properties, if any, is displayed. If there are no properties that can be extracted, dialog box displays “No Properties” column. |
| Description | Describes, in short, the extracted properties. You can change the description, if required. |
| Properties | Displays the value of extracted property that you can change while editing the component. |
| Association | Displays the association of extracted property with component. For example, whether it is a tag, a constant or a window name, etc. |
| Locking the Property | You can lock or unlock the property by double-clicking on the property description or icon.. A lock is displayed on the property icon. Locking makes the extracted property unchangeable. By making the property unchangeable you can disallow the process engineer to configure or edit property value of the component wherever and whenever it is loaded. By default, the property is unlocked. That means all the extracted properties are changeable by default. |
| Save to library | Check this box to save the component in a library. A drop-down list gives the name of existing libraries. Select the name if you want to save the component in the existing library. Enter the name for a new library, if the library with specified name does not exist. ASTRA prompts before creating the library. |
| Script | Gives the Component Script Editor. |

6.3 Component Library

ASTRA provides a Library facility for reusability of the Symbols and the Components created. Symbol or Component libraries can be created using the **Library** menu options.

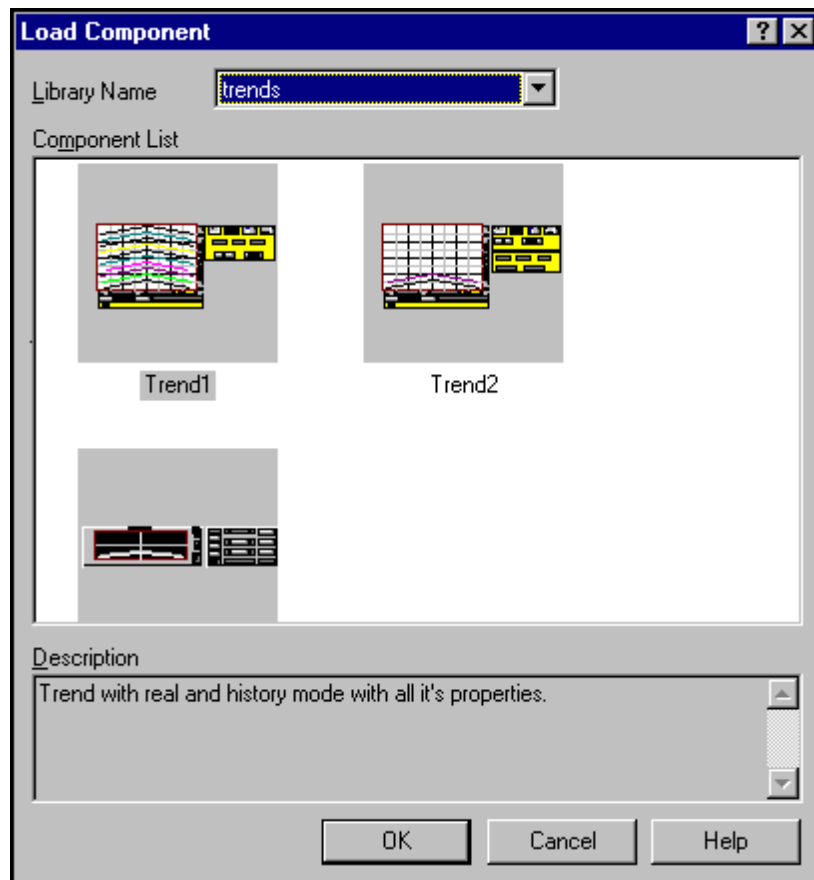
You can save the symbols in symbol library by simply cut-pasting and copy-pasting the drawn symbols. You can maintain different libraries for components..

To create a new Component library and to save the component in the component library, you can use the **Edit | Make Component** or **Edit | Edit Component** menu options. Component libraries are stored in the *lib* directory under the ASTRA root directory. You can get a component from Component Library, by clicking the **Library | Load Component** menu option.

ASTRA saves the component with the name entered in the component library. Any other special character, except the underscore, in name is not accepted. You can use alphanumeric characters in name. However, you have to start the name with alphabets.

6.4 Loading the Component

You can load existing components from the Component library in any project window. Before using any component from the component library, the tags associated with the component along with its properties needs to be added manually in to the Tag database of the application project. If a tag is extracted as a property then the tag and its data change script, if any, should be manually not automatically added to the project when the component is loaded from library in some other project. To load a component, click the **Library | Load Component** menu option. The Load Component dialog box appears.



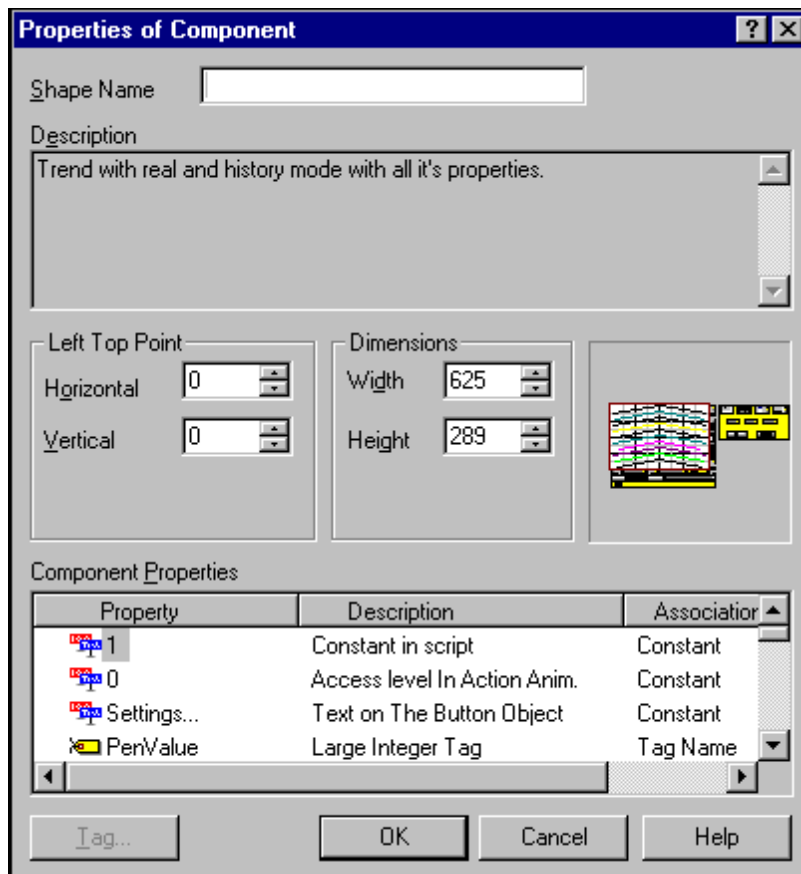
| | |
|-----------------------|--|
| Library Name | Select the name of the library from which where you want to load the component is to be loaded. |
| Component List | Displays the images and names of components available in from the library selected. |
| Description | Displays the description of a selected component. This description is assigned by the designer of component while saving component in the component library. |



6.5 Configuring Component

When a component is loaded from the component library, some of its properties as Position, Dimensions and extracted properties can be reconfigured to suit the current project.

You can change its default position and dimensions. You can also change the extracted properties of components. To configure the component, select it and click the **Edit | Properties** menu option or Press **F4** key. Or Even a Double-click on the component. The **Configure Component** dialog box appears.





| | |
|-----------------------------|---|
| Shape Name | Enter the name for the component shape. This name is specific to the current project. It is not related to It has nothing to do with the component name in the component library. |
| Left Top Horizontal | 'Select the X' co-ordinate of the left top point of the component in the window. |
| Left Top Vertical | 'Select the Y' coordinate of the left top point of the component in the window. |
| Dimensions Width | Width of the component. If you enter (0 or 1 is treated as the width, ASTRA makes it 2. You cannot enter a negative values are not accepted). |
| Dimensions Height | Height of the component. (0 or 1 is treated as 2 negative values are not accepted). |
| Component Properties | Extracted and changeable properties of the component are displayed. High light the property and change the parameter to configure component for the current application project. |
| Property | Change the value of extracted property to configure component. The change in parameter is applicable wherever it appears in the current project. |
| Description | Displays the description assigned to the extracted property at the time of making component. |
| Association | Displays the association of the parameter with component. |
| Tag | Displays the 'Tag Browser' dialog box to select the tag for changing the parameter. This button is enabled only for the property with having tag association. |

6.6 Breaking a Component

To break the component select it the component and click the **Edit | Break Component** menu option. If you have configured the component and changed any of its parameter, after breaking component the object within component retains the changed value.

Example:

If you have rectangle1 and rectangle2 and you define Slider animation on rectangle2. The tag attached is temp1. You make component of two objects and configure it. You change the tag to temp2. If you break the component now, rectangle 2 has the slider animation attached but the tag is temp2 and not the original temp1.



Chapter 7: Tag Database

Tags are the variables holding the data. To open the Tag database, click **Tools | Edit tag database** menu option. ASTRA provides you the following four types of tags as described below:

- System Tags
- Memory Tags
- Device Tags
- Special Tags

7.1 System Tags

Tags with '**System**' as a node are the System tags. These System tags are **Read only** tags and are predefined. The properties of this type of tags can not be edited. These tags are designed to read the values from the System or from a predefined location in ASTRA. There are 14 system tags as listed below:

| Sr.No | System Tag | Can be used to display |
|-------|---------------------|---|
| 1 | _User | The Login user name |
| 2 | _Access level | Access level of the Login user |
| 3 | _Date | Current date (DD) |
| 4 | _Day | Current day |
| 5 | _Month | Current month (MM) |
| 6 | _Year | Current year (YYYY) |
| 7 | _Hour | Current hour value |
| 8 | _Minute | Current minute value |
| 9 | _Second | Current seconds value |
| 10 | _Msecond | Current milliseconds value |
| 11 | _AlmActAckCount | Active, acknowledged alarms (count) |
| 12 | _AlmActUnAckCount | Active, unacknowledged alarms (count) |
| 13 | _AlmInActUnAckCount | Inactive, unacknowledged alarms (count) |
| 14 | _DiskFree | Hard disk free space (in Mb) |

7.2 Memory Tags

Tags with '**memory**' as a node are the Memory tags. These are Memory tags user definable. The user can edit the properties of these type of tags. Typically these tags can be used for internal calculations and for temporary storage of the data.

7.3 Device Tags

Tags with a user defined node name (for an external device, as defined in the node list) are the device tags. Device These tags represent the actual field variables for that node. ASTRA uses these tags for collecting the current values of the monitoring and controlling field variables.

7.4 Special Tags

Special tags are similar to the device tags excepting that they have a predefined address and they provide the information about the Node. These tags are automatically created whenever a new Node is defined or can also be defined by the user. These are the tags enabling **node control** facility. The 3 special tags provided by ASTRA are:

| | |
|----------------------------------|--------------------------------|
| <Node-name>_Status | Displays the status of a node. |
| <Node-name>_Name | Identifies the failed node. |
| <Node-name>_Command | Controls the node. |

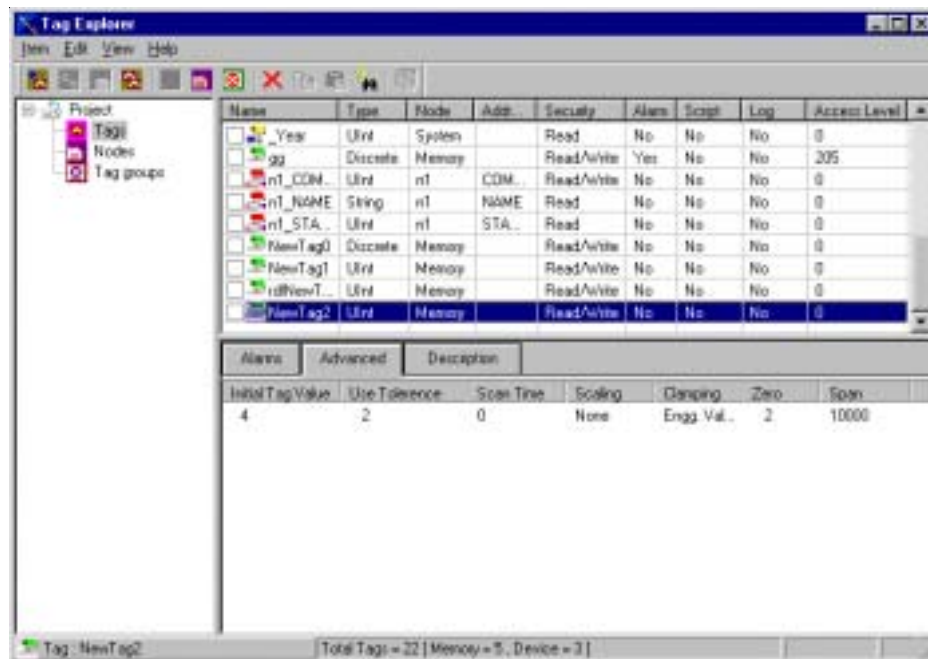
7.5 Adding, Editing or Deleting the Tags

Tags can be added, edited or deleted from the Tag browser in the configuration mode. Tag browser can be accessed in the following ways:

- Click **Tools | Edit Tag Database** menu option.
- Press **F6** Key.

Double click on the **Tags** icon during the configuration of the project.

ASTRA will display the **Tag Explorer** with a tag list along with the following options:



Tag List:

Tag list is a list of the existing tags with their important attributes. Select the desired tag from the alphabetical list of tags if you want to delete or edit the tag.

Add New:

Click right button of mouse to add a new tag. Either select from **Item | New | Tag**.

Edit :

Select and double click the leftmost mouse button.

Delete:

Either click rightmost mouse button and delete the tag or select **Item | Delete**.



Show Columns:

Displays the **Tag List Settings** dialog box to select the tag attributes to be displayed in the tag list of the Tag Browser. The attributes include Node, Type, Security, Address, Scan time, Alarm, Log, Clamp, Tolerance, Scale.

If you have accessed this dialog box from the Animation dialog box to select the tag for a particular animation, the list gives tags applicable to the specified animation.

For example, if “data entry” animation is defined then only the tags with read-write property are listed. The tags with read-only property and system tags are not listed. Or if you have chosen “Slider” then only the tags having data type integer are displayed.

Select:

Selects the highlighted tag for the animation, pen, component property or the script from where you have called the list. This button is disabled if you have accessed the browser using the **Tools | Edit Tag Database** menu option or by pressing **F6**.

You can also select the tag by double-clicking on it. You can select only one tag at a time. To highlight the tag for selecting, editing or deleting, you can navigate in to the Tag List using the arrow keys or the Search button.

Arrow buttons:

Arrow buttons are enabled only if total tags are more than those displayed in one frame.

| | |
|----|---|
| << | Goes to the first frame of tag list. |
| < | Goes to the previous frame of tag list, if any. |
| > | Goes to the next frame of tag list, if any. |
| >> | Goes to the last frame of tag list. |

Searching for a Tag:

If the tag list in the Tag Browser is running across multiple pages and you want to search a particular tag, ASTRA allows you to search that tag. Access the Tag Browser from any of the action points depending on your current activity and click on the Search button. The Search Tag dialog box appears.

| | |
|----------------------|---|
| Search String | Enter the tag name that you want to highlight in the Tag Browser. |
|----------------------|---|



7.6 Tag Properties

The various properties to be defined are as under:

| | | |
|-----------------|-------------------------|--|
| Tag Name | | Enter the unique tag name for the project |
| Node | | Select the node for the tag you are defining. You can define a tag, either as memory tag or a device tag. |
| Address | | This field is enabled only if you have selected device node. Enter the address of the tag, as in the Driver release notes and Driver manual. |
| Type | | ASTRA supports six data types for tags: |
| | Discrete | Boolean value, False = 0, True = 1 |
| | Integer | -32767 to 32767 |
| | Unsigned Integer | 0 to 65535 |
| | Large | -2,147,483,647 to 2,147,483,647 |
| | Real | 1.5 E-45 to 3.4 E38 |
| | String | Text (memory tags only). |
| Security | | Allows or disallows the assignment of value to the tag through Read only and Read write options. |
| Log Data | | Check this box to log data in the data logger files. It is not enabled for string type of tags. |
| Script | | Opens the script editor to write "Data Change" scripts. |
| Alarm | | Displays the Alarms dialog box to define alarms on the current tag. |
| Advanced | | Gives the Advanced Attributes dialog box. For more information on Advanced Attribute refer section "Setting the Advanced Attributes". |
| Add | | Adds the tag you have created to the tag database on validating the name, address and the value fields. |
| Delete | | Deletes the selected tag from the tag database after confirming. |
| New | | Allows you to create a new tag. |
| Close | | Closes the Tag Properties dialog box. If there are any changes, ASTRA confirms whether to save them or not. |



7.7 Setting the Advanced Attributes

From the Tag Properties dialog box, click on the **Advanced** button. Advanced attributes are:

| | |
|--------------------------|--|
| Initial Tag Value | ASTRA assigns this value to the tag when the project starts at run-time. |
| Use Tolerance | Check this box to set the margin for value change acceptance. Enter the value that will act as the least count for the engineering values. Change in value is displayed or logged only if the change is more than or equal to the tolerance value. |
| Scan Time (ms) | This field is enabled only for the device tags. Enter the rate, in milliseconds, at which the tag is to be scanned at run-time. Tag is scanned at this rate if the tag does not belong to any tag group or if it belongs to the scan tag group. |
| Scaling | Choose Linear scaling to convert raw values of device variables to the engineering values. If you choose this option you need to specify two ranges, one of raw values and the other of the engineering values. The raw values in the specified range will be linearly converted to the specified range of engineering values. |

7.8 Clamp Engineering Values

This box is enabled for numeric memory tags. For the device tags this box is enabled only if the linear scaling is selected. Check this box to clamp engineering values and to set minimum and maximum values for clamping. In case of device tags this range is linearly scaled with the range of raw values.

| | |
|----------------|---|
| Minimum | Enter the value to which all engineering values below this value will be clamped. |
| Maximum | Enter the value to which all engineering values above this value will be clamped. |

7.9 Clamp Raw Values

This box is enabled only for the device tags having numeric data types. Check this box to clamp the raw data and to set the minimum and maximum values for clamping.

| | |
|----------------|---|
| Minimum | Enter the value to which all raw values below this value will be clamped. |
|----------------|---|



7.10 Using Tag Groups

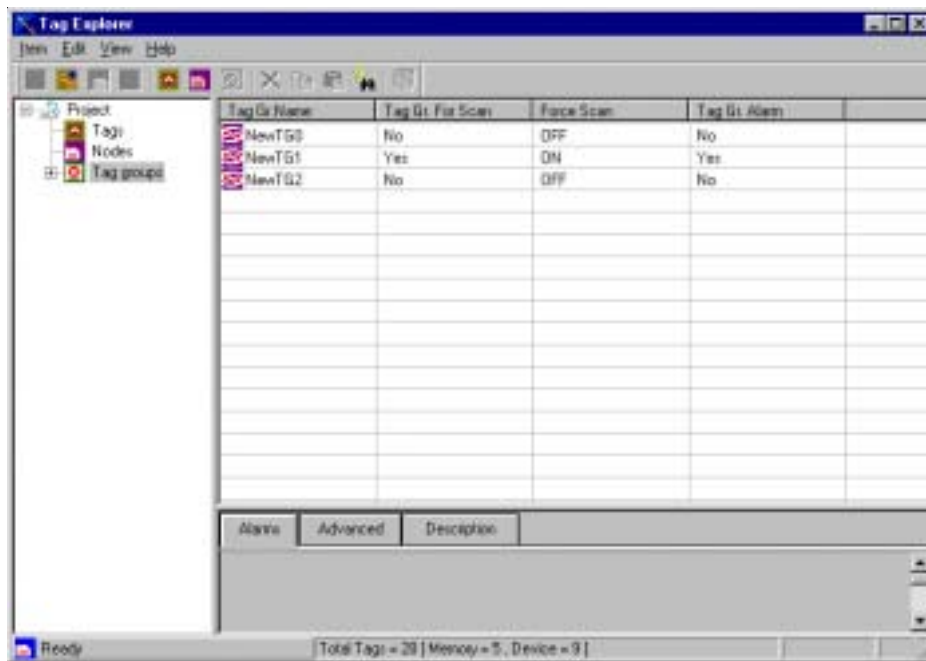
ASTRA provides you with a facility that allows you to group the logical tags, irrespective of their properties. You can associate scan and alarm properties to tag groups. Use a drop-down list of existing tag groups in the application project. If there are no tag groups in the application project, this option list is disabled. Each tag can be a part of multiple tag groups. Tag groups can be edited or deleted.

Understanding Tag Groups:

Tags can be grouped for two purposes - For Scanning and for Alarms.

- Scan Tag Group: Tags that are required to be scanned while the project is running.
- Alarm Tag Group: Associates all the tags in the group to single alarm window.

To define a new tag group click **Items | New | Tag Groups** menu option. The **Tag Group** dialogue dialog box appears.





7.11 Adding Tag Group

Click right button of mouse to add a new tag. Either select from **Item | New | Tag**.

| | |
|--------------------------|---|
| Tag Groups | Enter new name for the tag group. You can use (Numeric or other special characters are not accepted). |
| Properties | Click this button to assign a property to the new Tag group. There are two main properties of the Tag group are: |
| Scan Property | All the tags in the tag database of ASTRA are scanned after predefined interval of time as per the requirement of the application project. The application project may consist of some of the tags, which are used for downloading the parameter values from the recipe. These tags do not or only some times require the scanning. These tags can be grouped to form a Scan tag group. Scanning of this tag group can be made on or off with the help of the Script while the application project is running. You can also set this property while forming the group by selecting the Force Scan On or Force Scan Off options. |
| Alarm Property | Logically related tags can be grouped to form an Alarm tag group. The Alarm tag group can be associated with an Alarm window. Thus logically related alarms will be displayed in one alarm window. If a tag is in multiple tag groups and all the tag groups are associated with one alarm window, alarm on that tag will be displayed only once. |
| Tags in Tag Group | The lists of tags in the selected for this tag group. |
| Add | Click this button to get a list of all the tags existing in the project. |
| Delete | Deletes the highlighted tag from the Tags in Tag Group list. The tag is deleted only from tag group and not from the tag database. |

7.12 Editing Tag Group

To edit, double click the leftmost mouse button for selected tag group.

In this mode the tag group name can be changed or the tags can be added or deleted from the group.

Deleting a Tag Group:

To delete a Tag group, open **Item | Delete**. Click the rightmost mouse button and select Delete option. The dialog box is similar to the Add Tag Group dialogue dialog box except the name.

You can delete unwanted tag groups. The selected tag group is deleted with after confirmation. However, in case of any associations, ASTRA does not delete the tag group until the association is deactivated.

Example

If the tag group is associated with any alarm window, it is not deleted till you delete the tag group name is taken out from the alarm window property.

Chapter 8: Node Control

Node is a link between the field and ASTRA. Each node defined in ASTRA represents a field device. Click **Tools | setup nodes** menu command to define the Node. After defining a node ASTRA automatically creates a set of Tags known as **special Tags**. The special tags can also be added manually at a later stage. Following are the list of special tags:

- **<Node-name>_Status**
- **<Node-name>_Name**
- **<Node-name>_Command**

Node control feature of ASTRA detects the failure of a node. If any node is found as failed, at initial scan, a choice is given to ignore the failed node and to continue with other nodes. Or while the process is running, Or while the process is running, the status of node can be displayed with the help of special tags. On line, you can also start or stop certain nodes without discontinuing the process.

With the help of these special tags **Node control** performs the following functions.

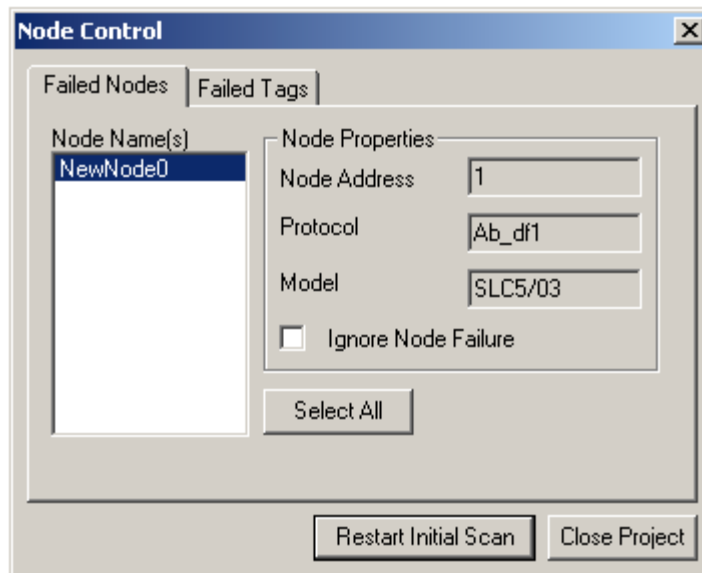
- Detects the failure of a node.
- Identifies the node failed.
- Allows the user to start or stop the communication with a node.

For any project, node control activity is done in two stages.

- 1 During the initial scan.
- 2 While the project is running.

8.1 Node control during the Initial Scan

ASTRA checks for the healthy communication with node (device) connected. If any device fails to communicate with ASTRA, **failure of some node is detected, then ASTRA prompts you to ignore the node. Node control** dialog box is displayed. This dialog box displays the name of the node failed



This dialog box provides the user with the following option:

- To ignore the failed node. OR
- To continue with the rest of the project. OR

It allows you to ignore the failed node and restart initial scan for other nodes or exit the application project.



8.2 Node Control while the project is running

Data transfer between the device and ASTRA tag database starts after the initial scan. To detect the failure of a node while the project is running, ASTRA makes use of the special tags. These tags can be configured to display the status of the nodes. The information stored by each of these tags is as follows:

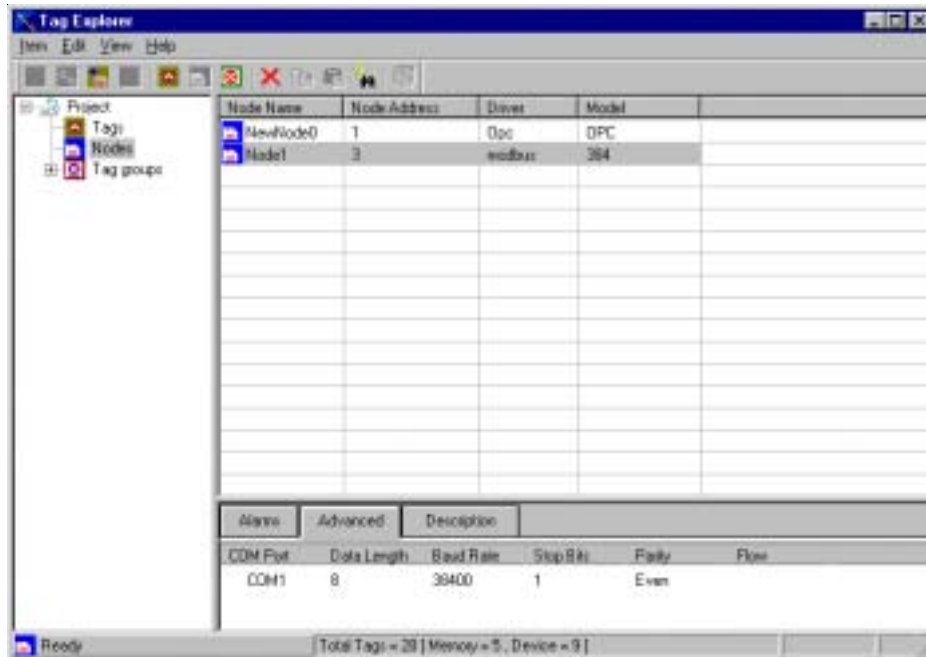
| | |
|----------------------------------|--|
| <Node-name>_Status | Allows user to see the status of a node at run-time. The address of tag has to be Status. This is a read only tag. The data type is unsigned integer. Its values are 0 for "running node", 1 for "stopped node" and 2 for "failed node". User can monitor this tag to detect node failure, at run-time. To monitor the tag you can use tools like animation or alarm. |
| <Node-name>_Name | This tag tells a user the name of the node at run-time. The address of tag has to be Name. This is a read only tag. The data type is string. The node-name becomes its value. |
| <Node-name>_Command | Allows user to start or stop a node at run-time whenever required. The address of tag has to be Command. This is a read-write tag. The data type is unsigned integer. Its values are 0 for "start node" and 1 for "stop node". User can change the value of this tag to start or stop node, at run-time. To change the value of the tag you can use "Enter Data" animation or write script using WRITETAG command. |

Note: If a node fails to communicate, ASTRA makes three attempts to fetch the data from that node. If the fetching fails, all the three times in succession, node is considered as failed. If during any of the three attempts if node is found working then fetching continues and the values are assigned to the tags. If the node is failed then the tags pertaining to in that node retains the last fetched values. However the user can force 'safe' values by using the data change script on the **<Node-name>_Status** tag.

8.3 Setting-up a Node

Nodes are the important links between actual process and ASTRA. Each node represents stands for the field device to be linked connected to with ASTRA.

To set-up the node for a project, click the **Tools | Set-up Nodes** menu option. The **Set-up Nodes** dialog box is displayed.



| | |
|--------------------|---|
| Select Node | The list is provided with node address, driver and model. |
| Add | Add a new node and its properties. |
| Edit | Edit the properties of an existing node. |
| Delete | Delete the selected node. |



8.4 Adding a Node

To add a node, click **Item | New | Node** menu option. Also can be added by clicking the rightmost mouse button.

| | |
|---------------------|---|
| Node Name | Enter the name for the node to be added. |
| Node Address | In a multi-drop system, driver uses this identifier for serial line communication with the device. Enter the unique identifier for the node. (For more information, please refer Release manual for ASTRA device driver). |
| Driver | Provides the list of the installed device drivers. Select the driver to be used for the current node. |
| Model | The drop-down list displays the device models supported by the selected driver / protocol. Select the appropriate model for the current node. |

8.5 Editing a Node

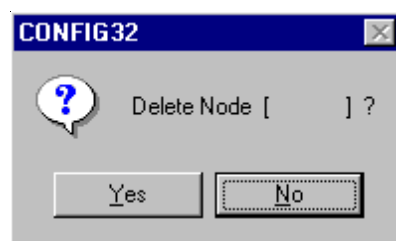
To edit a node, click **Tools | Setup Node** menu option. Double click the leftmost mouse button and the selected grid becomes editable.

ASTRA allows you to change the Node ID, Protocol and the Model associated with the node. You cannot edit the Node Name.

| | |
|---------------------|--|
| Node Address | Enter the unique identifier for the node. |
| Driver | Provides the list of the installed device drivers. Select the driver to be used for the current node. |
| Model | Displays the device models supported by the selected driver / protocol. Select the appropriate model for the current node. |
| OK | Accepts changes made to the node setup. |
| Cancel | Cancels changes and returns you to the Setup Nodes dialogue box. |

8.6 Deleting a Node

To delete a node, click **Tools | Setup Node** menu option. The Setup Nodes dialog box appears on the screen. From the Setup Nodes dialog box select the node from the Select node option and click on the **Delete** button.



ASTRA prompts you to confirm the delete operation. Press **“Yes”** to delete the node. Press **“No”** to cancel the delete operation.



Chapter 9: Alarms

Alarm is one of the important features supported by ASTRA. When you define the alarm on a tag, alarm is generated whenever the value exceeds the specified range or deviates from the target value by the specified percentage. For every value change beyond the specified range, alarm becomes active. For the acceptable value alarm is inactive. The alarm types supported are High - Low and Deviation type.

9.1 Defining Alarm on a Tag

Alarm can be set for each individual tag defined in ASTRA. To define an alarm for a tag, follow the following procedure:

- 1 Select the tag from the tag browser
- 2 Click **Alarms** button. The **Define Alarm** dialog box will be displayed.

The screenshot shows the 'Define Alarm' dialog box with the following configuration:

- Alarms** tab selected.
- Set Alarm** button.
- Alarm Text:** Alarm1
- Alarm Settings:**
 - Low Low: 100
 - Low: 150
 - High: 250
 - High High: 500
- Alarm Type:**
 - None
 - High Low
 - Deviation
 - Boolean
- Actions:**
 - Display
 - Print
 - Severity: 5



| | |
|---------------------|--|
| Alarm Text | Alarm description. (Max. upto 50 characters). |
| Alarm Types | You can define two types of alarm - Highlow and Deviation. |
| Highlow | <p>If you select a Highlow type, dialogue box displays the following options:-</p> <ul style="list-style-type: none">• Lowlow• Low• High• Highhigh <p>Check the desired box to define alarm for the limit. For every box checked, you get the text box to enter the tag value. Enter the tag value for every limit you have chosen. You can enter maximum of 12 characters.</p> |
| Deviation | If you select Deviation alarm, you will have following options: |
| Target Value | Enter the target value from which the deviation is to be reported. |
| Minor | Check this box if you want to report the minor deviation from target value and enter the percentage of deviation, from the target value. |
| Major | Check this box if you want to report the major deviation from target value and enter the percentage of deviation, from the target value. |
| Action | Two types of options are available for all the alarm types, Display and Print. You can display the alarm in window as well as print the alarm. If you choose the Display option you need to have an Alarm window in the project. If you choose the Print option you need to set the Alarm Printer. For displaying as well as printing alarms you need to set the Alarm Message Format. |
| Severity | Severity level of the alarm ranges between 0 and 9. The default value of the severity level is 5. ASTRA displays only those alarms having the severity level greater than or equal to that of the alarm window. |

Note: If you have configured report printing as well as alarm printing you need to set two different ports for both of these printings. If you are using only one port, you have to disable any one of them.



9.2 Configuring the Alarm Window

An Alarm window can be drawn either from the tool box using the draw alarm tool or from the **Draw | Alarm** menu option.

Once an alarm window is drawn, the prompt to edit the alarm properties is displayed. Alarm properties can also be edited using the **Edit | Properties** menu option. The different types of properties that can be set from the 'Alarm Properties' dialogue box are explained below:

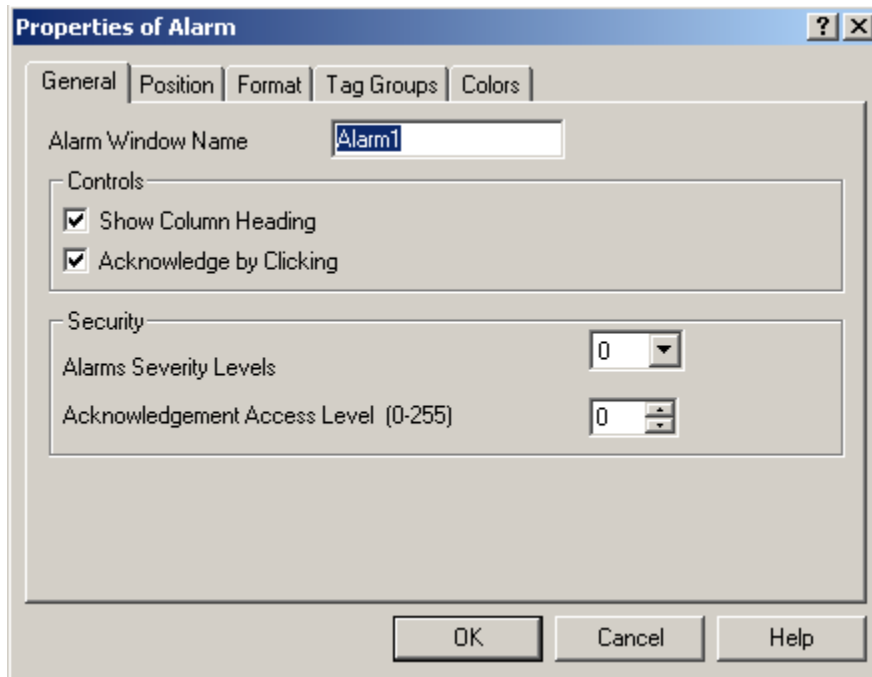
Properties Related to Alarms:

The following properties of Alarms related to alarms can be used through scripts.

| Proper | Data Type | Access Object Type |
|-----------------|--------------|--------------------|
| AckAll | Service | Alarm |
| AlarmCount | Integer Read | Alarm |
| Format | Service | Alarm |
| ScrollDown | Service | Alarm |
| ScrollRight | Service | Alarm |
| ScrollLeft | Service | Alarm |
| ScrollUp | Service | Alarm |
| SeverityInteger | Read / Write | Alarm |
| Sort | Service | Alarm |



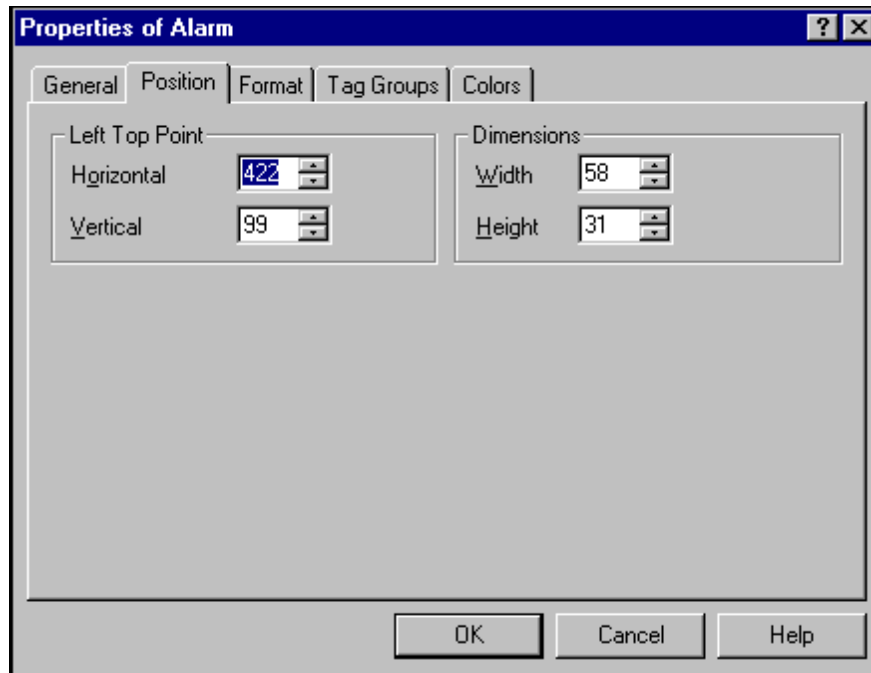
9.3 General Properties



| | |
|---|---|
| Alarm Window Name | Enter User definable name for the alarm window. |
| Show Column Headings | Check this box to display the column headings for the alarm message. |
| Acknowledge by Clicking | Check this box to enable acknowledgement of the alarm. The alarm can be acknowledged on the screen with a double click on the alarm message. The color of the alarm message gets changed once it is acknowledged. |
| Alarm Severity Level | Enter or select the severity level of the alarm window. The available range is between 0 and 9. Only the alarms with greater or equal severity level will be displayed in this window. |
| Access Level for Acknowledgement | Enter the access level between 0 and 255 for the operator. Operator with the greater or equal access level will be able to acknowledge the alarms from this window. |



9.4 Position Properties

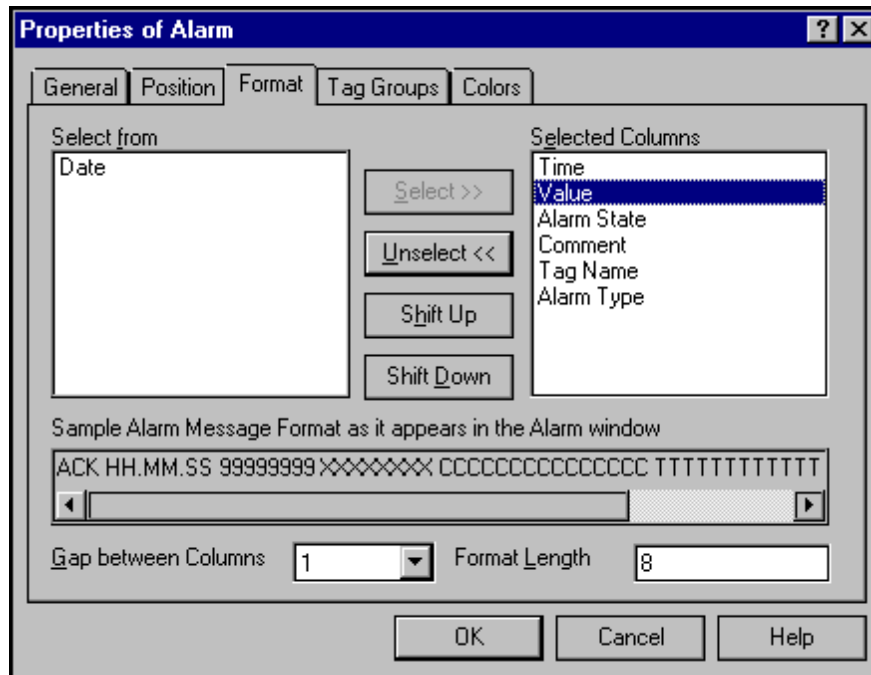


| | |
|------------------------------|---|
| Left Top - Horizontal | Select the X co-ordinate of the left top point of the alarm window. |
| Left Top - Vertical | Select the Y co-ordinate of the left top point of the alarm window. |
| Dimensions Width | Select or enter the width of the alarm window (min 42). |
| Dimensions Height | Select or enter the height of the alarm window (min 30). |

Note: ASTRA accepts negative values and 3-digit number for the position settings. All co-ordinate measurements are in pixels.



9.5 Format Properties



Any alarm can have the following fields in its message:

Date , Time , Tag Name , Value , Alarm Type , Alarm State , Comment. These can be either selected or deselected and their position can be changed using the following options.

| | |
|----------------------------|--|
| Select from | Lists the fields that may be selected for the message. Highlight the required field(s) of the alarm, and click Add button. The field is added to the Selected Columns list. |
| Selected Columns | Displays the fields that are selected which would appear in the alarm message. To deselect the unwanted field(s) highlight them and click the Delete button. |
| Select | Adds the highlighted field from the ' Select from ' list to the ' Selected ' Column. |
| Unselect | Deletes the highlighted field from the ' Selected ' Column and brings it back to the ' Select from ' column. |
| Shift Up | Click on this button to move the field up by one position. The sequence of the fields in this column decides the display sequence of the fields in the alarm message. (This button is enabled only if the field from the ' Selected ' Column is highlighted and if the field is not the first one in the column). |
| Shift Down | Click on this button to move the field down by one position. The sequence of the fields in this column decides the sequence of the fields in the alarm message. (This button is enabled only if the field from the Selected Column is highlighted and if the field is not the last one in the column). |
| Gap Between Columns | Enter or select the space between two columns. (Max. upto 9 spaces). |
| Format Length | Enter the format for the highlighted field from the ' Selected ' Column. For the Date and Time fields you can select the format from the combo box. For other fields you need to enter number of characters for the format. |

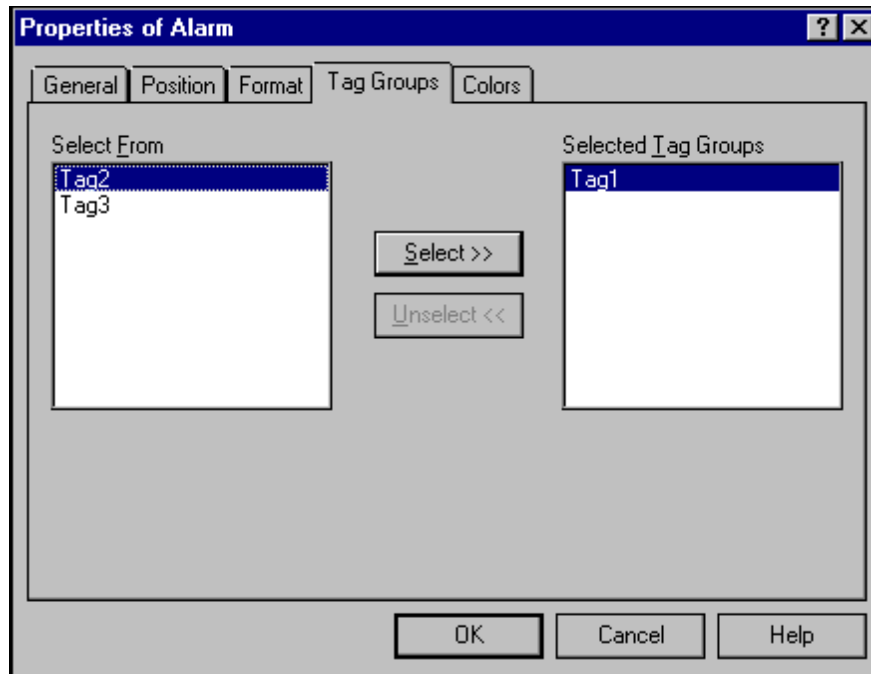


9.6 Tag Group Properties

ASTRA provides the advanced feature of grouping the logical tags. Alarm property can be associated with a tag group and these logical tag groups can be associated with an alarm window. Such an Alarm window displays only the alarms for those tags which are included in the logical tag group. The alarms on system tags are by default included in the alarm window. This gives more convenience in acknowledging alarms on logical tags.

The tag group of the logical tags can be defined from the **Tools | Tag Group** menu option and the alarm property can be attached to the relevant groups.

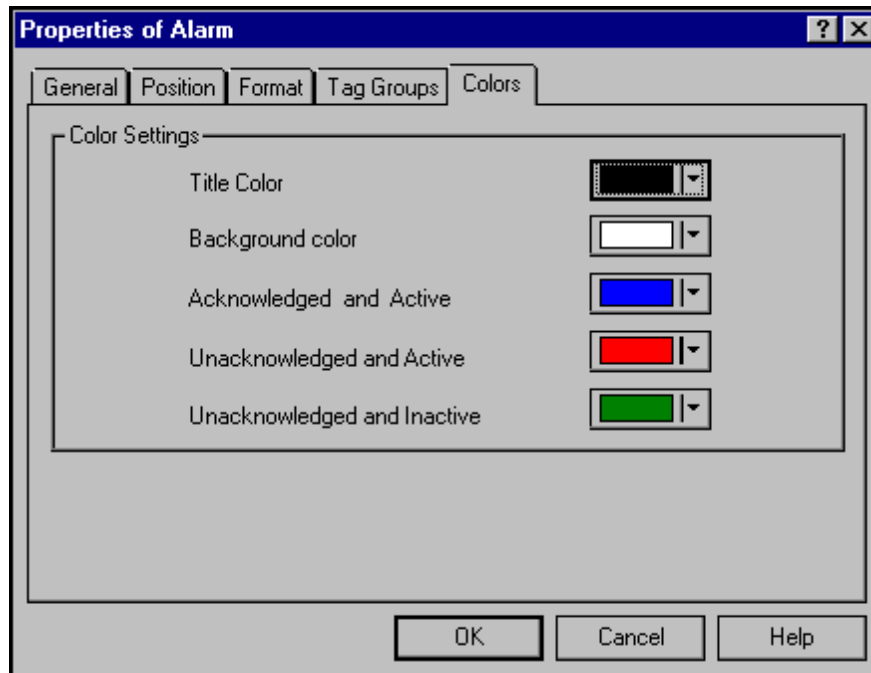
For more details on Tag Groups and their properties, please refer to chapter 7 “Tag Database”, section “Using Tag Groups”.



| | |
|----------------------------|--|
| Select From | Lists the tag groups that can be attached to the current alarm window. Highlight the desired tag group, one at a time, and click the Add button. The tag group is added to the Selected Tag Groups list. |
| Selected Tag Groups | Lists the tag groups that are selected. To deselect the unwanted tag group highlight the field and click delete button. |
| Select | Associates the Tag Group with the alarm window. |
| Unselect | Dissociates the highlighted tag group from alarm window. |



9.7 Color Properties



Color Selection:

You can set the colors of the following parameters of the alarm window:

- Title Color
- Background Color
- Acknowledged and active alarms
- Unacknowledged and active alarms and
- Unacknowledged and inactive alarms

Click on the color rectangle to change color of any of these parameters. The corresponding color palette appears. Click on the desired color.

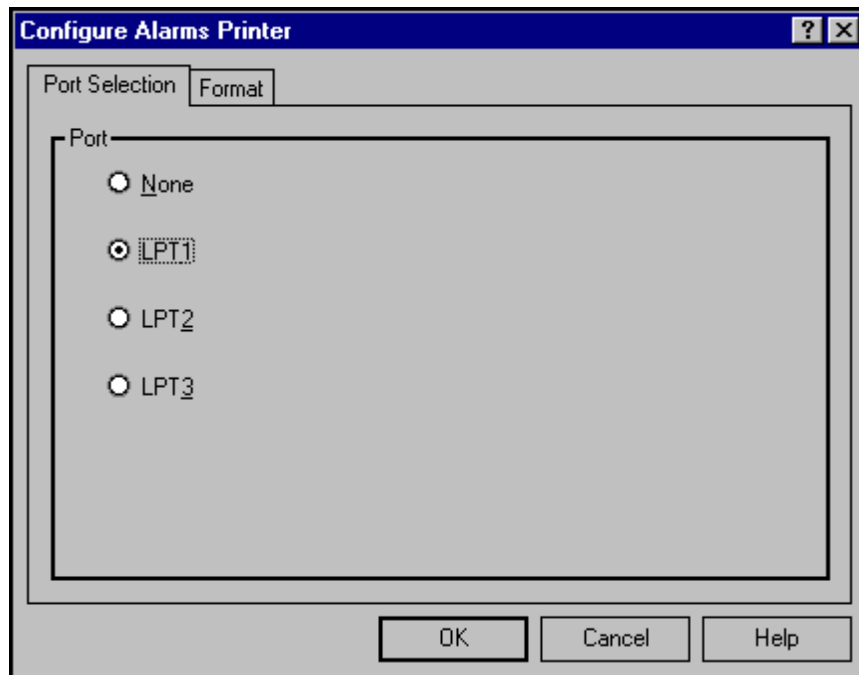
Note: Alarm display can be configured using *SORT* and *FORMAT* script commands. At run time, the Alarm Sort Order dialog box appears if you use a *SORT* command and the 'Alarm Format' dialog box appears if you use the *FORMAT* command. Other properties like Changing the Priority, Acknowledging or Scrolling can also be defined using scripts.



9.8 Setting Alarm Printer

Alarm messages can be printed by clicking the Print option in the **Define Alarm** dialog box. ASTRA prints the alarm whenever its status changes (from active to inactive or vice versa) or it is acknowledged.

If you want to print the alarm messages, you need to set the default port for alarm printer. To set the printer clicks the **Tools | Setup Alarms Printer** menu option. The **Configure Alarms Printer** dialog box appears.



9.9 Port Selection

Choose the Local Printer Port that is connected to your printer. The available ports are LPT1, LPT2 and LPT3.

Warning

If you click the **None** option or the wrong port, ASTRA will not print the alarms at run time. The event logger will display the error message accordingly.

Note: *If you have configured report printing as well as alarm printing you need to set two different ports for both of these printings. If you are using only one port, you have to disable either one of them at a time.*

9.10 Format Tab

Set the format for alarm messages to be printed. The Format Tab of Properties of Alarm dialog box allows you to set up the format of alarm messages to be displayed and logged.

Note: *If there are multiple windows displaying the same alarm ASTRA prints it on the printer only once for each status change.*



9.11 Logging of Alarms

ASTRA has the facility of logging the alarms automatically in a file, in the text format. If any alarm is generated, Astra logs the alarm messages in to a text file. In Run Mode, when the process is running, and if any alarm is generated, it is logged in to a text file called the *alm.log* file. This file is typically generated and stored in the project directory.

You can start or stop alarm logging in the Run mode. An option is provided in the **Logger** menu option. Alarms are logged in the same format as that is set for printing alarm messages.

If the format for printing alarm messages is not specified, the alarms are logged in the default format. When 3000 alarm messages are logged in the *alm.log* file., Astra creates a backup of this file. The backup file is named as *almbk.log*. The *alm.log* file gets cleared and new messages will be logged in to it. If the *alm.log* file is opened by some other application and if ASTRA is unable to write messages to the file, after alarm logging is started, ASTRA creates the *alm1.log* file. If *alm1.log* file is also opened, Astra will create *alm2.log* file, and so on up to *Alm20.log* file.



Chapter 10: Data Logger and Reports

You can keep a log the process parameter values using its built-in data logger. This logger saves the data in the form of .dlo files, in the log directory. This data is displayed in the graphical format using Historical Trends, in a proprietary format.

The name of the log file is in format *ddmmyyhh.dlo,o* representing the year, month, date and hour at which the file is saved. These files can be used for off-line applications such as MIS reporting. The logged data can also be displayed in a graphical format using Historical Trends.

Data Logger:

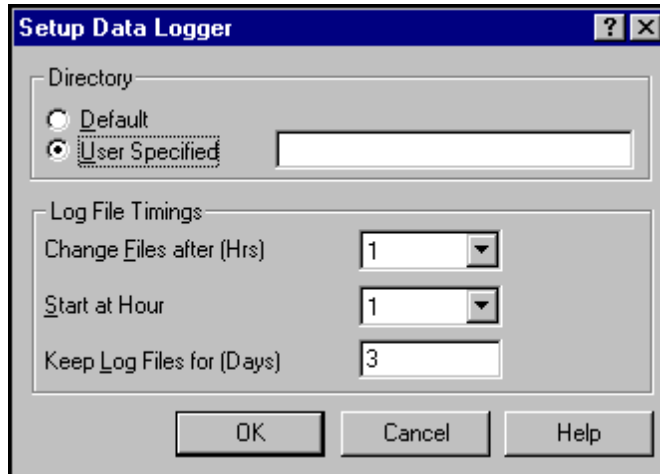
If you choose the data logging option or enable history trend the built-in data logger of Astra, it logs data in proprietary format. ASTRA generates a *log* directory in the root directory of the project. This directory contains the files in which data is logged. The name of the file is in format *ddmmyyhh.dlo* indicating year, month, date and hour when the file is created. ASTRA provides a converter utility for conversion of log file to DBF format. In the log file, the following information is logged. Each record of this file contains the following information

| | |
|-----------------------|---|
| Tag Name | Name of the tag that is logged. |
| Tag Type | Data type of the tag. (String tags cannot be logged). |
| Tag Value | Value of the tag. It is stored as numeric value with maximum possible resolution. Once the logging is started, values of all tags for which the data logging is enabled are logged. There after the tag is logged when its value is changed beyond the tolerance value specified in the Tag Database. |
| Log Time stamp | Date and Time at which the record is logged. This data is taken from the system clock. If the system clock is not set properly then wrong timings for logged data will be reflected. |



10.1 Setting the Data Logger

To read the .dlo files in the text format, use the ASTRA file conversion utility from program files of ASTRA. To set the path of the log file, locations, timeings, changing log files, click **Tools | Setup Data Logger** menu option. The **Setup Data Logger** dialoguedialog box is displayed with the following options.



| | |
|----------------------------------|--|
| Directory | Enter path for saving the log files. It has two options, as described below: |
| Default | The default path is <i>log</i> sub directory under the project directory. |
| User Specified | Option to save the log files in to the directory of your choice. Enter the directory name with complete path. |
| Log File Timings | Enter the time to change and delete the log files. It has three controls, as described below: |
| Change Files After (Hrs) | Enter or select the duration after which each log file should change. For example, if you specify 3 hours, a new log file will be created after every 3 hours counting from the 'StartHour'. |
| Start at Hour | Enter or select the time in hours to which the 'change files' duration should be added. |
| Keep Log Files for (Days) | Number of days for which each of the log file should be stored. Log (.dlo) files will be deleted automatically after completion of the specified days from the date of creation. |



10.2 Configuring Reports

Reports can be configured in ASTRA with the help of following options:

- Script commands
- Any other windows based package-supporting DDE like Excel using DDE services.

The exclusive Report Generation facility provided by ASTRA can be used either to produce text reports or the graphic reports. The reports can be printed automatically at a predefined time interval using the powerful Script commands.

Report Generation using Script Commands:

Generating Text Reports:

You can generate the text reports. They can be generated in ASTRA using PRINTTEXT script command. To generate such a report it is necessary to specify the report id, row- and column positions and the text matter to be printed in the report.

Each text report can contain upto a maximum of 255 characters. You can specify tag names for printing text. Using this command you can generate a report of maximum 255 characters. This command overwrites the previous text, if the text is present at the specified position. The default system font is used to print the report.

Generating Report Templates:

You can generate the template for the report using PRINTTEMPLATE script command. Using it you can create headers-footers. Along with the script command it is necessary to specify for the report-id. You can specify report-id, page numbers and the text to be printed on every page. You can generate templates only for the text reports.

PRINTTEMPLATE does not create Report ID. You must create Report ID by first using the PRINTTEXT command.

Generating Graphic Reports:

You can generate the graphic reports by using PRINTWINDOW command. It generates the graphic report by specifying the report-id and window name. The window print area can be defined by specifying the left top or a part of it by specifying X and -Y co-ordinates and the dimensions (Width and height) of the window to be printed in Pixel sizes. (For sending entire window contents to a printer or to a file, give the values of bitmap in height and width as 0). You can place the graphic of any size in the report. You can have multiple bitmaps in one report. You can also have multiple bitmaps in one page. The window to be printed needs to be at the top when the report is fired.

Saving Reports in a File:

You can save both the text and graphic reports in the files by using SENDTOFILE command. The report needs to be specified with the report Id and the file name with the path. You can send the report to file by specifying the report-id and the file name. The file extension need not be specified. If you specify the file extension it is neglected by ASTRA. For text reports ASTRA provides .rep file extension and for the graphic reports, ASTRA uses the page number as the extension with the specified file name. The graphic report file is a bitmap file and can be opened in paintbrush. For example, if you specify the file name as window1 the first page of the graphic report is saved in the file *window1.1*. The second page is saved in the file *window1.2*, and so on.



You can specify and append the existing text report or overwrite the existing one. However, you cannot append the graphic report to the existing file. A multiple paged text report can be saved in a single file. However, when you send a graphic report to the file, for each page a separate file is created. You can choose to send only the new or changed pages of the report to file by keeping the save entire option false. To send the entire report to file, make this option true.

Printing Reports:

You can print both the text and graphic reports. They can be printed in the files by using SENDTOPRINTER command. The printer needs to support the graphic printing to print a graphic report. Graphic report is printed if the printer supports graphic printing. You can print the report specifying the report-id and by choosing an option to print. You can choose to print only the new or changed pages of the report or the entire report by keeping the printentire option false. To print the entire report make this option true.

The printed report is always sorted, primarily by the page number and then by the row number.

Deleting Reports:

You can delete the unwanted reports. They can be deleted using DESTROYREPORT command. When the report is sent to file or to the printer it is automatically deleted. In case of graphic reports it is advisable to delete the report using the DELETEREPORT command after storing it in to a file or after printing.

10.3 Examples of Report Generation

It is necessary to use the script commands to print or to save the Reports.

A Single Page Text Report to be Saved in a File:

The Syntax is,

```
PrintText <ReportId>, <Page Number>, <Row>, <Column>, <Text to be Printed>;  
SendToFile <ReportId>, <File Name>, <Append>, <SaveEntire>;
```

A Single Page Text Report to be Printed:

The Syntax is

```
PrintText <ReportId>, <Page Number>, <Row>, <Column>, <Text to be Printed>;  
SendToPrinter <ReportId>, <File Name>, <SaveEntire>;
```

A Multiple Paged Text Report to be Saved in a File:

The Syntax is

```
PrintText <ReportId>, <Page Number-1>, <Row>, <Column>, <Text to be Printed>;  
PrintText <ReportId>, <Page Number-2>, <Row>, <Column>, <Text to be Printed>;  
PrintText <ReportId>, <Page Number-n>, <Row>, <Column>, <Text to be Printed>;  
SendToFile <ReportId>, <File Name>, <Append>, <SaveEntire>;
```



Similarly, user can print the multiple paged text report using the SendToPrinter command

A Text Report with Header is to be Printed:

The Syntax is,

```
PrintText <ReportId>, <Page Number>, <Row>, <Column>, <Text to be Printed>;  
PrintTemplate <ReportId>, <Row>, <Column>, <Header Text>;  
SendToPrinter <ReportId>, <File Name>, <SaveEntire>;
```

Syntax to provide header-1 for first 2 pages of the report and header-2 for the last three pages of the report the is:

```
PrintText <ReportId>, <Page Number-1>, <Row>, <Column>, <Text to be Printed>;  
PrintText <ReportId>, <Page Number-2>, <Row>, <Column>, <Text to be Printed>;  
PrintTemplate <ReportId>, <Row>, <Column>, <Header1 Text>;  
PrintText <ReportId>, <Page Number-3>, <Row>, <Column>, <Text to be Printed>;  
PrintText <ReportId>, <Page Number-4>, <Row>, <Column>, <Text to be Printed>;  
PrintText <ReportId>, <Page Number-5>, <Row>, <Column>, <Text to be Printed>;  
PrintTemplate <ReportId>, <Row>, <Column>, <Header2 Text>;  
SendToPrinter <ReportId>, <File Name>, <SaveEntire>;
```

Similarly, report can be saved in a file using SendToFile command.

A Report with Single Bitmap to be Saved in a File:

The Syntax is

```
PrintWindow <Window Name>, <ReportId>, <Source Left Top X>, <Source Left Top Y> <Bitmap Width>, <Bitmap  
Height> <Destination Left Top>, <Destination Left Top Y>;  
SendToFile <ReportId>, <File Name>, <SaveEntire>;
```

Multiple bitmaps can be printed on a single page by entering appropriate Left Top X and Y positions on destination, depending on the sizes of bitmaps. To print the Entire window, enter the bitmap height and width as 0.

10.4 Report Generation through Dynamic Data Exchange (DDE)

Creating DDE links in the target program application operating under Windows can also configure reports. An example of creating DDE links in MS EXCEL is illustrated below:

- 1 Run the project in ASTRA.
- 2 Open the target program application (EXCEL) for creating the DDE links.
- 3 Select a cell in EXCEL spread sheet and enter the following formula to establish the DDE link with ASTRA.

=DATAMANAGER | TAGTABLE! TAGNAME

Where:

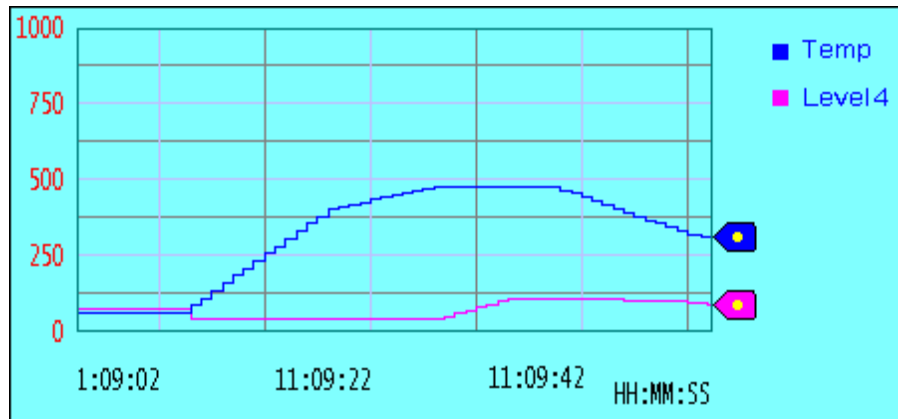
Datamanager: Is the application name (fixed).

Tagtable: File containing the data (fixed).

Tagname: Is the name of the tag whose value is to be displayed (variable).

Chapter 11: Trends

Trend is a horizontally scrolling graph displaying changes in tag values. Trend of a tag value is displayed against time. ASTRA allows you to configure two types of trends.



11.1 Real-time Trend

On line display of the current tag values in the Graphical form. User can use this graph to monitor the process at run-time.

To configure the real-time trend, insert a trend object in the window and set the Trend Properties. When you set the trend properties you can define its appearance, attach tags to the Y axis and enable History Trend.

11.2 History Trend

Online display of the tag values pertaining to a user defined date, time and period, in the Graphical format. This trend displays the graph of historical data. User can specify the Start date, start time and the time span of the Historical trend. To display the historical data, the data stored in data logger is used. ASTRA allows you to see the history data dating back for almost four years. User use this graph to analyse the process.

To configure the history trend you need to insert the trend object in the window and enable history trend. Rest of the settings can be done using scripts. History trend covers the data that is logged up to the point when you switch to history trend from the real-time trend in Run Mode.

11.3 How to draw Trend

Trend can be drawn with the help of the trend tool or from the **Draw | Trend** menu option. The prompt to edit the trend properties is displayed after the Trend object is drawn. Press 'Yes' to display the 'Properties of trend ' dialogue box. Trend properties can also be edited using the **Edit | Properties** menu option.



11.4 Properties Related to Trend

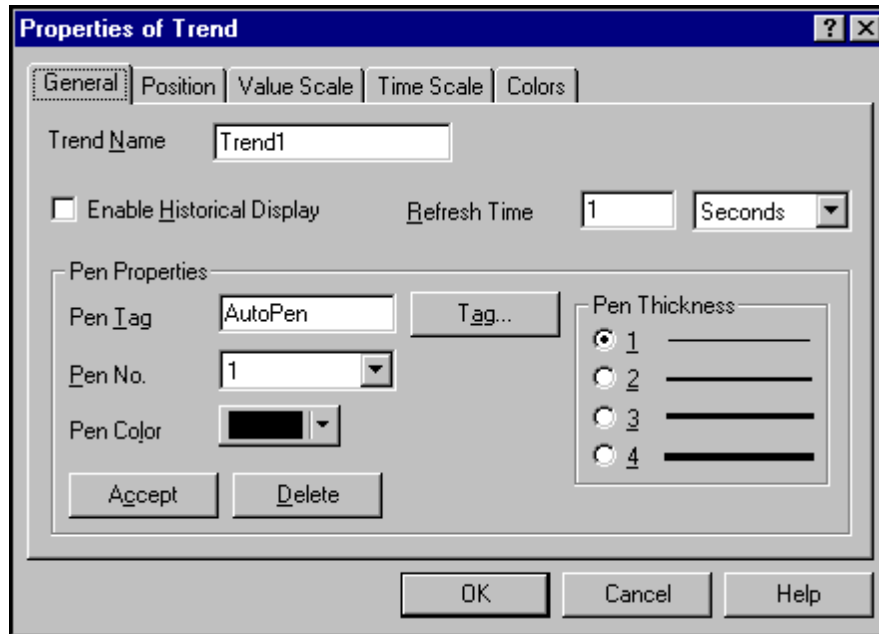
The following trend properties can be accessed through scripts.

| Property | Data Type | Access Object | Type |
|-----------------|------------------|----------------------|-------------|
| CurrentPen | Integer | Read/Write | Trend |
| Cursor Large | Integer | Read/Write | Trend |
| DrawMode | Integer | Read/Write | Trend |
| Mode | Integer | Read/Write | Trend |
| Path | String | Read/Write | Trend |
| PenTag | String | Read/Write | Trend |
| PenValue | Real | Read | Trend |
| Redraw | Service | | Trend |
| Skip | Large Integer | Read/Write | Trend |
| StartDate | String | Read/Write | Trend |
| StartTime | String | Read/Write | Trend |



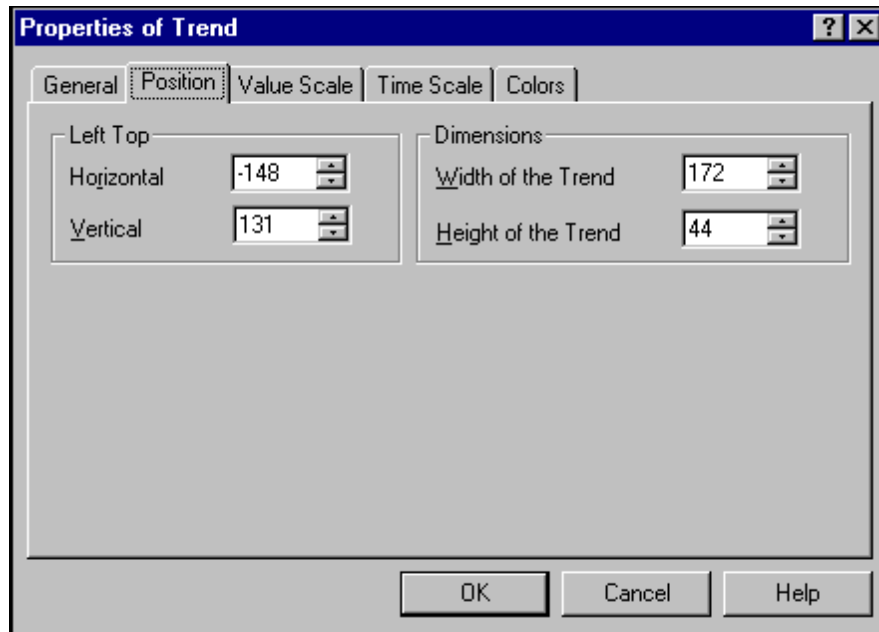
11.5 Trend Properties

11.5.1 General Properties



| | |
|----------------------------------|--|
| Trend Name | User definable name for the trend. |
| Enable Historical display | Click this box to use the current trend window for displaying the historical trend. Enabling this option automatically enables the data logging for the Tags defined in the Trend. |
| Refresh Time | Enter the time after which the trend is to be scrolled (min 50 milliseconds) |
| Pen Tag | Name of the tag that is to be attached to the pen. |
| Pen No | Enter or select the pen number for the tag you want to attach to the trend. |
| Pen Thickness | Enter or select the thickness of the trend. ASTRA provides four thickness levels. |
| Pen Color | To set the color of the trend for the current pen, click on the color rectangle. The Color palette is displayed. Select the desired color. |
| Accept | Accepts the pen attributes. |
| Delete | Deletes the current pen after confirmation. |

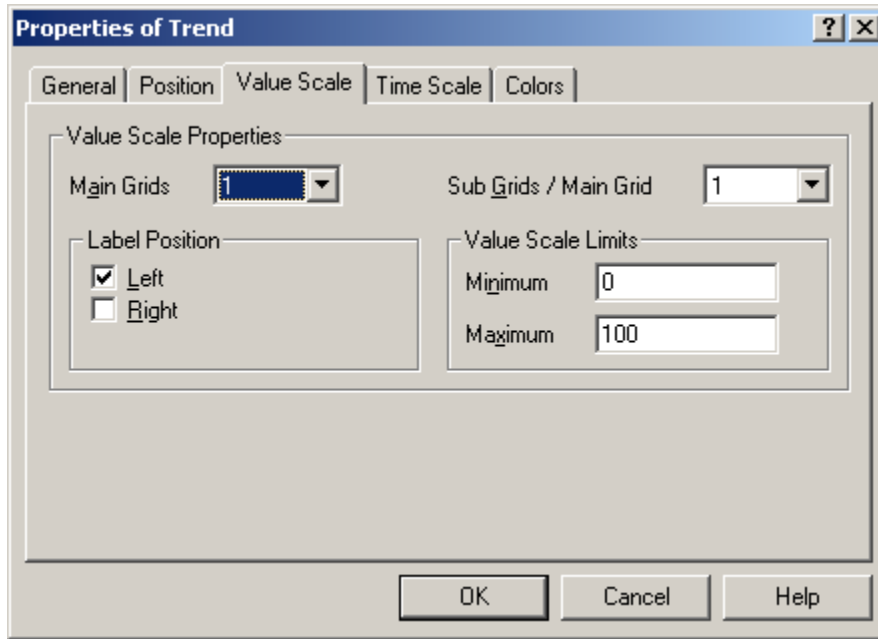
11.5.2 Position Properties



| | |
|---|--|
| Left Top Horizontal and Vertical | X and Y co-ordinates of the left top point of the trend |
| Width of the Trend | Width of the trend window (min 172). |
| Height of the Trend | Height of the trend window (min 44). |

Note: ASTRA accepts negative values and 3 digit numbers for all these parameters and all measurements are in Pixels.

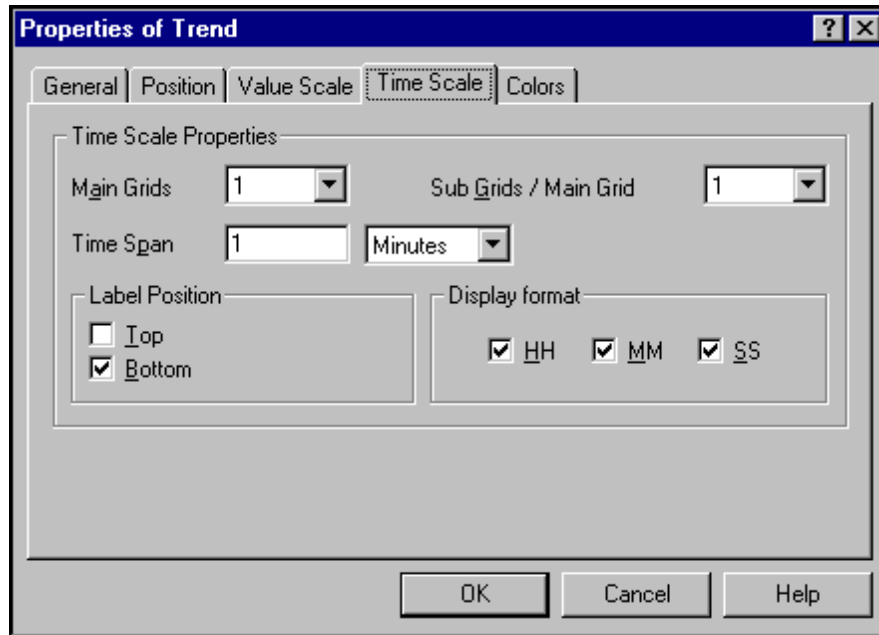
11.5.3 Value Scale Properties



| | |
|------------------------------|--|
| Main Grids | Number of main grids for the value (Y) axis. |
| Sub-Grids / Main Grid | Number of sub-grids in each main grid for the value (Y) axis |
| Label Position | Position of the value label. |
| Value Scale Limits | Minimum and the Maximum range of the value. (x axis). |



11.5.4 Time Scale Properties

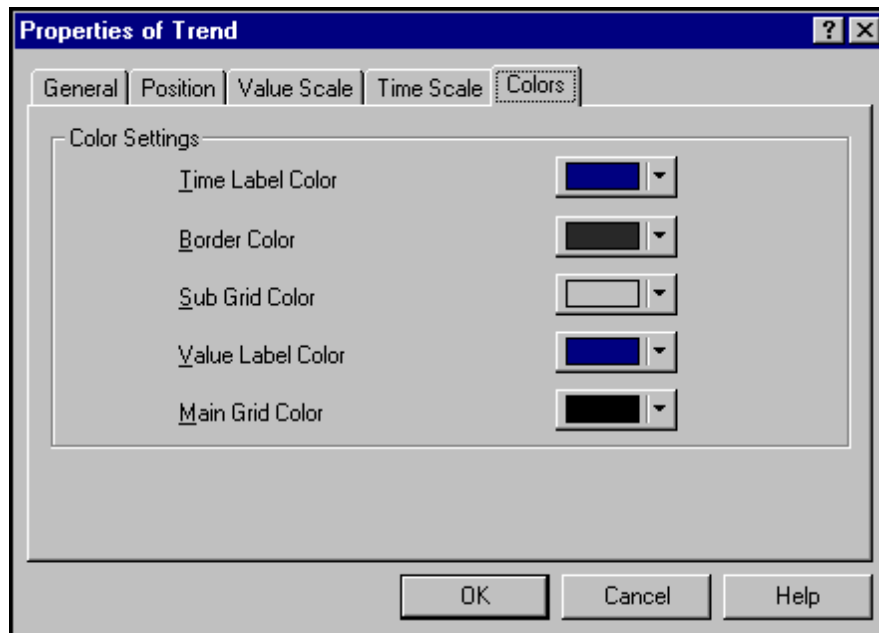


| | |
|------------------------------|--|
| Main Grids | Number of main grids for the Time (X) axis. |
| Sub-grids / Main Grid | Number of sub-grids in each main grid. |
| Time Span | Total time span of the X-axis. Time span can be in Hours, Minutes or in Seconds. |
| Label Position | Position of the time labels for X-axis. |
| Display Format | Units of Time to be displayed in the time label. |

Note: ASTRA internally calculates the trend scan rate. Scan rate of the Trend is the time interval after which the tag values are refreshed. It is calculated as Time Span / Width of the trend object. It can be decreased by either increasing the width or by decreasing the time span.



11.5.5 Color Properties



Color:

Provides color palette for selection of the colors for the Time Label, Value Label, Borderline, Main Grid and Sub Grid. The background of the trend is the same as the Window color as trend is a transparent / unfilled object.



Chapter 12: Scripts

Script is a sequential set of commands and procedures that is executed at run-time. Scripts are written for event-driven, taking control actions, designing complex interlocks and simulating the complete control logic.

12.1 Script Level

Script can be written at different levels within a project. These levels are explained below:

- Application (Project)
- Window
- Component
- Object
- Tag

For all these levels, there are multiple events at which the script will be executed. The following sections describe the events for each level.

12.1.1 Project Level

Script can be written at the project level with options for execution of the script. The options are:

| | |
|------------------------|---|
| Before start up | This script is executed before the project is opened at run-time. |
| While running | This script is repeatedly executed at the defined Scan Rate. |
| After shut down | This script is executed when the project is closed. |

To open the project level script editor, open **File | Configure** menu option. The **Project Options** dialogue box appears. Click on the **Script** button. The Script Editor appears to write project level scripts.

12.1.2 Window Level

Script can be written at the Window level with options for execution of the script. The options are:

| | |
|-----------------------|--|
| Before showing | This script is executed once just before the window is opened. |
| While showing | This script is repeatedly executed at the defined Scan Rate once the window is opened. |
| After hiding | This script is executed once just before the window is closed or hidden. |

To open the window level script editor, open the window for which you want to write script. Open **Edit | Properties** menu option or double-click in the window or press **F4** Key. The **Windows Properties** dialogue box appears. Click on the **Script** button. The Script Editor appears to write the window level script.



12.1.3 Component Level

Script can be written for a component with options for execution of the scripts. The options are:

| | |
|--------------------------|--|
| At initialisation | This script is executed once when the window-containing component is displayed. |
| Periodically | This script is executed, periodically, at the specified Scan Rate, once the window-containing component is opened. |
| After termination | This script is executed once just before the window containing component is closed or hidden or the project is closed. |

To open the Component level script editor, select the component for which you want to write script. Open **Edit | Edit component** menu option. The **Edit Component** dialog box appears. Click on the **Script** button. The Script Editor appears to write the window level script.

12.1.4 Object Level

Scripts can be written for the objects (typically for buttons) through Take action animation. The options for execution of the script are:

| | |
|--------------------------|---|
| At key down | This script is executed when the user clicks left mouse button on the object having Take Action animation or presses the respective hot key, if any |
| While key pressed | The specified Scan Rate executes this script, periodically, while the left mouse button or hot key is pressed down. |
| After key up | This script is executed after the mouse button or hot key is released. This script is also executed even if the mouse button or hot key is pressed and ESCAPE key is pressed. |

Note: Key refers to the mouse click or the shortcut key defined for the object.

To open the Object level script editor, select the object for which you want to write script. Attach Take action Animation to the object. Click on the Script button. The Script Editor appears to write the object level script.

12.1.5 Tag Level

Scripts can be written for the tags for execution of the script after data change. The tag level script is called as Data Change script. This script is executed once, every time the tag value changes.

To open the Tag level script editor, edit the tag from the Tag browser. Click the Script button. The Script Editor appears to write the tag level script.

Note: A user cannot change the value of a tag in its Data Change script. This might lead to unpredictable behaviour.



12.2 Script Command

12.2.1 IF-THEN-ELSE Statement:

The IF-THEN-ELSE statement (loop) is used for the conditional execution of the script within the loop.

Syntax

```
If <Condition> Then
<Statement list>
Else
<Statement list>
Endif;
```

12.2.2 WHILE DO Statement:

WHILE DO loop is used to execute a statement or a group of statements, repetitively based on a condition.

Syntax:

```
While <Boolean function> Do
<Statement list>
Endwhile;
```

Operator Precedence

Operators are listed in the order of precedence.

() - , NOT , * , / , + , - ,
< , > , <= , >= , <> , == , = (This is the conditional operator.)
And , Or , = (This is the assignment operator.)

12.2.3 Arithmetic Operators:

Addition [+]

This operator is used for Addition Operation.
TAG_C = TAG_A + TAG_B;

Subtraction [-]

This operator is used for Subtraction Operation
TAG_C = TAG_A - TAG_B;

Multiplication [*]

This operator is used for Multiplication Operation
TAG_C = TAG_A * TAG_B;

Division [/]

This operator is used for Division Operation
TAG_C = TAG_A / TAG_B;



12.2.4 Conditional Operators:

Parenthesis [()]

This operator is used to decide the priority of evaluation for arithmetic operations. Operations written in parenthesis are evaluated with highest priority.

```
TAG_D = ( TAG_A + TAG_B ) / TAG_C
```

Greater Than [>] and Less Than [<]

Greater Than or Equal to [>=] and Less Than or Equal to [<=]

These operators are used for comparison between two variables. Typically used in IF-THEN-ELSE statements to execute various instructions based on the comparison.

Syntax

```
IF TAG_A > TAG_B THEN  
<Statements>;  
ENDIF;
```

Equal Equal to [==]

This operator is used to check if a value of one variable is Equal to the other. Typically It is used in the IF-THEN-ELSE statements.

Syntax:

```
IF TAG_A == TAG_B THEN  
<Statements>;  
ENDIF;
```

Not Equal to [<>]

This operator is used to check if a value of one variable is not Equal to the other. Typically It is used in the IF-THEN-ELSE statements.

Syntax:

```
IF TAG_A <> TAG_B THEN  
<Statements>;  
ENDIF;
```

Note: Both the arithmetic and conditional operators are used for numeric expressions only.



12.2.5 Assignment Operator:

Equal to [=]

This operator is used to assign a value to the Memory tag.

Syntax
TAG_A = <Numeric Expression>;

WRITETAG

This operator is used to write the value to device tag in the data manager as well as in the device. It is applicable only to the device tags.

Syntax:
WRITETAG <Device Tag Name>, <Numeric Expression>;

FORCETAG

This operator is used to assign values to the Device (I/O) tags. It assigns value to the tag in Data Manager irrespective of the node status. In data manager, it changes the value of device tag. **It does not change the actual value in device.** Hence it can be used to keep the process running if a particular node fails.

Syntax:
FORCETAG <Device Tag Name >, <Numeric expression>

12.2.6 Bitwise (Logical) Operators:

The following are the Bitwise operators that can be used in Scripts:

- 1) **AND** 2) **OR** 3) **NOT**

Syntax:
IF TAG_A = 10 AND TAG_B = 20 THEN
<Statements>;
ENDIF;

12.2.7 Boolean Functions:

| | |
|---------------|--|
| OFF | Used in the Boolean expressions where OFF means FALSE. |
| TOGGLE | Toggles the discrete read-write memory tag (ON / FALSE).Syntax TOGGLE <Tag Name>; |
| TRUE | Works as other Boolean operators like any non-zero value and ON. |
| ON | Can be used in the Boolean expressions where ON means TRUE. |
| FALSE | Works as other Boolean operators like, OFF or 0. |



12.2.8 String Concatenation Operator:

'+' This operator is used to concatenate strings.

For example `string_tag = "Stock" + "Report";`
It returns the value of `string_tag` as "Stock Report".

Math / Trigonometric operators

ABS

This operator is used to calculate the absolute value of an expression.

Syntax:

`ABS <Numeric Expression>;`

POW

This operator is used to return the value of first numeric expression raised by the second numeric expression.

Syntax:

`POW (<Numeric Expression1>, <Numeric Expression2>;`

If `<Numeric Expression2>` is 0, then returns 1. If `<Numeric Expression1>` is not zero and returns 0 if `<Numeric Expression1>` is also zero.

If the value returned is of type float and it is assigned to type integer/long/unit then it is truncated before assigning.

Exceptions like floating point underflows or overflows will be truncated according to the lowest or highest limit. The event logger will display the message of lowest or highest limit.

SQRT

This operator gives the square root of the numeric expression. Returns 0 for value less than or equal to 0. For negative numbers, the Event Logger displays the message "Square root of negative number. Using 0."

Syntax:

`SQRT (<Numeric Expression>;`

SIN

This operator is used to perform trigonometric operation (Sine) on the given value. The numeric expression is treated as degrees.

Syntax:

`SIN <Numeric Expression>;`

ASIN

This operator is used to perform trigonometric operation (Sine inverse) on the given value. The output value is in degrees.

Syntax:

`ASIN <Numeric Expression>;`



COS

This operator is used to perform trigonometric operation (Cosine) on the given value. The numeric expression is treated to be in degrees.

Syntax:

COS <Numeric Expression>;

ACOS

This operator is performs trigonometric operation (Cos inverse) on the given value. The output value is in degrees.

Syntax:

ACOS <Numeric Expression>;

TAN

This operator is used to perform trigonometric operation (Tangent) on the given value. The numeric expression value is treated to be in degrees.

Syntax:

TAN <Numeric Expression>;

ATAN

This operator is used to perform trigonometric operation (Tan inverse) on the given value. The output value is in degrees.

Syntax:

ATAN <Numeric Expression>;

12.2.9 String Operators:

LCASE

This operator will return a string by converting it into lower case.

Syntax:

LCASE (<String Expression>;

LEFT

This operation is used to return the specified number of characters from the left of a string. If the <String Expression> is NULL, returns NULL. If the <Numeric Expression> is more than the length of the string, returns the complete string.

Syntax:

LEFT (<String Expression>,<Numeric Expression>;

RIGHT

This operator returns the specified number of characters of string from its right. If the <Numeric Expression> is negative or zero then returns NULL. If <String Expression> is NULL then returns NULL. If <Numeric Expression> is more than the length of string then returns a complete string.

Syntax:

RIGHT (<String Expression>, <Numeric Expression>;

ASC

Returns the ASCII numeric value for the first character in the given string.

Syntax:

ASC (<String Expression>;



FINDSTR

This operator is used to return the index where the second string is found in the first one. Index value is from 1 onwards. If any one of the string is NULL, returns 0. If the match is not found returns -1.

Bool Case is an optional parameter. If it is absent then FINDSTR is case sensitive. To make it case insensitive specifies Bool_Case parameter as 0. Non-zero Bool_Case parameter makes it case sensitive.

Syntax:

FINDSTR (<String Expression1>,<String Expression2>, [BOOL_CASE]);

LTRIM

This operator deletes all the blank characters on the left of the string.

Syntax:

LTRIM(<String Expression>)

RTRIM

This operator is used to delete the blank characters from the right of the string.

Syntax:

RTRIM (<String Expression>;

TRIM

This operator deletes the blank spaces from both the sides of a string.

Syntax:

TRIM(<String Expression>;

UCASE

This operator is used to change the string to the upper case.

Syntax:

UCASE(<String Expression>;

LEN

This operator returns the length of the string expression. Returns 0 if string is NULL.

Syntax:

LEN (<String Expression>;

LOGMESSAGE

This operator is used to log a message in the Event Logger. String Expression specifies the message to be logged.

Syntax:

LOGMESSAGE <String Expression>;



12.2.10 General Script Commands:

APPLICATION

This operator is optional keyword for EXIT command. Closes the project in run mode. Refer description of EXIT for more details.

BEEP

This operator creates a beep on the machine speakers. Numeric expression is converted to a positive integer before playing and beeps for those many numbers. If the expression evaluates to 0 or less value, no sound is played. If numeric expression is not given it beeps only once.

Syntax:

BEEP [<Numeric Expression>];

CHR

This operator returns the ASCII character for the given number. Works only for numbers in the range 0 - 255. For others returns NULL.

Syntax:

CHR (<Numeric Expression>);

DECR

This operator is used to decrement the tag value by 1. Tag should be read-write memory tag. The tag has to be a numeric tag and should not be discrete or string tag.

Syntax:

DECR <Tag Name>;

EQUAL

This operator compares two string expressions. Comparison is case sensitive.

Syntax:

EQUAL (<String Expression1>,<String Expression2>);

EXIT

This operator ends the execution of script if used without [APPLICATION] parameter. This is a very useful command to avoid writing complex ELSE clauses to give a list of IF statements one after another.

Example:

```
IF <Condition> THEN  
<Statement>  
EXIT;
```

If you use the optional keyword, APPLICATION, the project in Run Mode is closed.

Syntax:

EXIT [APPLICATION]



HIDE

This operator is used to make the window invisible but it is not closed. Name of hidden window appears in 'gray' in the window list.

Syntax:

HIDE [WINDOW] "<Window Name>"; OR HIDE "<Window Name>";

IIF

This operator returns the first string / numeric expression if the result of Boolean expression is True. Returns the second string/numeric expression if the result of the Boolean expression is false.

Syntax:

IIF (<BOOL Expression>, <String/NumericExpression1>, <String/Numeric Expression2>);

INCR

This operator is used to increment the tag value by 1. Tag should be a read-write, memory tag. It has to be a numeric tag. It cannot be string or discrete tag.

Syntax:

INCR <Tag Name>

LOAD

This operator is used to load the window in memory but does not display it. You can use SHOW command to display it. It will appear in the active window list. The Parameter [WINDOW] is optional.

Syntax:

LOAD [WINDOW] "<Window Name>"; OR LOAD "<Window Name>";

ME

This operator substitutes the window name for which the script is defined at compilation. Access the object from window without specifying the window name.

At compile time, ME cannot identify the Application Scripts and Data Change Scripts with any specific window. Therefore, ME should not be used in Application Scripts as well as in Data Change Scripts.

Syntax:

ME.<Shape Name>. <Property>

MID

This operator chops off the string and gets a part of it. Returns the part of the string starting from the character specified by <From_Numeric Expression>with length specified by <Length_Numeric Expression>. <Length_Numeric Expression>is optional. If it is not specified then returns the string from specified position up to the end of string.

If <From_Numeric Expression> is greater than the length of the string then returns NULL.

If <Length_Numeric Expression> is Zero or negative then returns NULL.

If <From_Numeric Expression>is Zero or negative then returns NULL.

If <Length_Numeric Expression> is greater than the string length then returns the characters up to the end.

PROJECTDIR

This operator returns a complete name of project directory including the drive. Path over the network can also be obtained.

Syntax

PROJECTDIR ();



RUN

This operator opens the specified project. The project to be run has to be clearly specified in a note marks. Event logger displays the status of executing the specified project. List of Status code and description is given below.

Syntax

RUN <String Expression>;

| | |
|----|--|
| 0 | System was out of memory, executable file was corrupt, or reallocations were invalid. |
| 2 | File was not found. |
| 3 | Path was not found |
| 5 | Attempt was made to dynamically link to a task, or there was a sharing or network-protection error. |
| 6 | Library required separate data segments for each task. |
| 8 | There was insufficient memory to start the project. |
| 10 | Windows version was incorrect. |
| 11 | Executable file was invalid. Either it was not a Windows application or there was an error in the .EXE image. |
| 12 | Application was designed for a different operating system. |
| 13 | Application was designed for MS-DOS 4.0. |
| 14 | Type of executable file was unknown. |
| 15 | Attempt was made to load a real-mode application (developed for an earlier version of Windows). |
| 16 | Attempt was made to load a second instance of an executable file containing multiple data segments that were not marked read-only. |
| 19 | Attempt was made to load a compressed executable file. The file must be decompressed before it can be loaded. |
| 20 | Dynamic-link library (DLL) file was invalid. One of the DLLs required to run this application was corrupt. |
| 21 | Application requires Microsoft Windows 32-bit extensions. |

SCAN

This operator makes the scanning of a tag group on or off. By default it is ON. Specify the Tag Group and the second parameter. Specify either ON or TRUE or 1 to put ON the scanning. Specify either OFF or FALSE or 0 to put OFF the scanning.

Syntax

SCAN <Tag group>, [TRUE \ FALSE];;

SHOW

This command displays a hidden window. Parameter WINDOW is optional.

Syntax:

SHOW [WINDOW] "<Window Name>"; OR SHOW "<Window Name>";

STR

This operator is used to convert the given numeric expression to string.

Syntax:

STR (<Numeric Expression>;



SUBSTR

This operator chops off the string and gets a part of it. Returns the part of the string starting from the character specified by <From_Numeric Expression>with length specified by <Length_Numeric Expression>. <Length_Numeric Expression>is optional. If it is not specified then returns the string from specified position up to the end of string.

If <From_Numeric Expression> is greater than the length of the string then returns NULL.

If <Length_Numeric Expression> is Zero or negative then returns NULL.

If <From_Numeric Expression>is Zero or negative then returns NULL.

If <Length_Numeric Expression> is greater than the string length then returns the characters up to the end.

Syntax:

SUBSTR (<String Expression>, <From_Numeric Expression>, [<Length_Numeric Expression>])

UNLOAD

This command is used to unload the window from the memory and hide the same. That is before unloading the window, ASTRA executes the After Hiding script. Keyword WINDOW is optional.

Syntax:

UNLOAD [WINDOW] "<Window Name>"; OR UNLOAD "<Window Name>";

12.2.11 Script for Recipe:

LOADRECIPE

This operator is used to load the specified recipe from the recipe INI file. Process runs using the loaded values of process parameters.

Syntax:

LOADSTATUS = LOADRECIPE (RECIPE_FILE_PATH, RECIPE_NAME);

SAVERECIPE

This operator is used to save the recipe in the specified recipe INI file. The INI file has to have the section with specified file name. This command can be used when the values of process parameters are changed at run-time and a new recipe is created that needs to be saved.

Syntax:

SAVSTATUS =SAVERECIPE (RECIPE_FILE_PATH, RECIPE_NAME);

Note: If the values of recipe parameters are changed at run-time, they are clamped, if required, using the tag database definition of the corresponding tags in a project. If the value is clamped for any variable, the message is shown in the event logger.



12.2.12 Properties Related to Simple Objects:

The following properties can be used in the scripts.

| Property | Data Type | Access | Object Type |
|-----------|---------------|------------|---|
| FillColor | Large Integer | Read/Write | Applicable to all objects except Bitmap, Alarms, Trends, Component and Group. |
| Filled | Discrete | Read/Write | Applicable to all objects except Bitmap, Alarms, Trends, Component and Group. |
| LineColor | Large Integer | Read/Write | Applicable to all objects except Bitmap, Alarms, Trends, Component and Group. |
| Name | String | Read | Applicable to all objects. |
| Show | Discrete | Read/Write | Applicable to all objects. |
| Text | String | Read/Write | Applicable to all objects. |

FILLCOLOR

This operator is used to fill the object with a specified color. This property is applicable to all objects except alarm and trend. Refer to “Color Index table” on page 99.

Syntax:

<Window Name>.<Shape Name>.FILLCOLOR = <Numeric Expression>;

FILLED

This operator decides the fill attribute of a shape. This is applicable to all objects except alarms and trends.

Syntax:

<Window Name>.<Shape Name>.FILLED = True/False;

LineColor

This operator is used to assign the specified color to the lines/borders of a shape. This property is applicable to all shapes except Alarms/Trends. Refer to “Color Index Table” on page 99.

Syntax:

<Window Name>.<Shape Name>.LINECOLOR = <Numeric Expression>;



Color Index Table:

| Column1 | Column2 | Column3 | Column4 |
|------------|------------|------------|-----------|
| 0 | 8,388,608 | 8,388,672 | 4,210,752 |
| 8,421,504 | 16,711,680 | 8,388,672 | 8,405,056 |
| 32,768 | 8,421,376 | 4,194,304 | 8,454,016 |
| 65,280 | 16,776,960 | 16,711,808 | 65,408 |
| 8,388,736 | 128 | 4,194,432 | 8,388,863 |
| 16,711,935 | 255 | 33,023 | 64 |
| 12,632,256 | 32,896 | 4,227,200 | 8,421,440 |
| 16,777,215 | 65,535 | 8,454,143 | 8,453,888 |

NAME

This operator invokes the name of the specified shape. This is a Read-only property. This is applicable to all shapes including Alarm/Trend.

Syntax:

<String Tag> = <Window Name>. <Shape Name>. NAME

SHOW

This operator is used to display or hide the specified shape. This property is applicable to all objects including alarms and trends.

Syntax:

<Window Name>.<Shape Name>.SHOW= True/False;

TEXT

This operator assigns the text on specified button or text object. This property is not applicable to any other object except button and text object.

Syntax:

<Window Name>.<Shape Name>.TEXT = <String Expression>;

CURRENTPEN

This operator is used to changes the current pen in use and assigns properties like PENTAG, CURSOR and PENVALUE (Between 0 and 10).

Syntax:

<Window Name>.<Trend Name>.CURRENTPEN = <Numericexpression>;

CURSOR

This operator sets the offset from current STARTTIME for cursor. Assigns the specified time in milliseconds as the offset.

Syntax:

<Window Name>.<Trend Name>.CURSOR= <Numeric Expression>;

DRAWMODE

This operator is used to change the graph drawing mode for a trend as specified by the numbers. The draw mode is set for the real time as well as the history mode.

Syntax:

<Window Name>.<Trend Name>.DRAWMODE = 0/1/2;



| | |
|----------|--|
| 0 | Draws in Scatter mode. Plots only the data points along the line of the graph. |
| 1 | Draws in Discrete mode. Joins data points assuming the value is constant till the next data point, thus giving the square waves. |
| 2 | Draws in Smooth mode. Joins data points directly, thus giving a smooth line. This is a default mode. |

MODE

This operator changes the mode of trend between real-time (by specifying 0) and history (by specifying 1). The default mode is a real-time trend.

Syntax:

<Window Name>.<Trend Name>.MODE = 0/1;

PATH

This operator is used to change the path used by history trend to search for data. Enter the valid path.

Syntax:

<Window Name>.<Trend Name>.PATH = <String Expression>;

PENTAG

This operator is used to change the tag name associated with the current pen.

Syntax:

<Window Name>.<Trend Name>.PENTAG = <String Expression>;

Note: If you use both, *CURRENTPEN* and *PENTAG* properties, and if the values are not matching, *ASTRA* uses the value of *PENTAG* to draw the trend.

PENVALUE

This operator returns the value of the current pen at the current cursor position.

Syntax:

VAL1 = <Window Name>.<Trend Name>.PENVALUE;

REDRAW

This operator refreshes the historical trend using current trend properties.

Syntax:

<Window Name>.<Trend Name>.REDRAW;

SKIP

This operator allows forward and backward scrolling of historical trend by the time specified in milliseconds.

Syntax:

<Window Name>.<Trend Name>.SKIP= <Numeric Expression>;



STARTDATE

This operator sets the date for searching in the database to display historical trend. The date has to be specified in MM/DD/YYYY format (or as set in Windows).

Syntax:

<Window Name>.<Trend Name>.STARTDATE= <String Expression>;

STARTTIME

This operator sets the time for searching in the database to display historical trend. The time has to be specified in HH:MM:SS:ss format.

Syntax:

<Window Name>.<Trend Name>.STARTTIME = <String Expression>;

TIMESPAN

This operator sets the time span in seconds for the historical trend. Data for the specified period in history is displayed. (Max. range of time span is 0 to 124416000 seconds equivalent to 4 years).

Syntax:

<Window Name>.<Trend Name>.TIMESPAN = <Numeric Expression>;

ACKALL

This operator is used to acknowledge all current alarms.

Syntax:

<Window Name>.<Alarm Name>.ACKALL;

ALARMCOUNT

This operator is used to count all the unacknowledged (active and inactive) alarms present in the alarm window.

Syntax:

<Window Name>.<Alarm Name>.ALARMCOUNT;

FORMAT

This operator displays and allows to change the 'Alarm Format' dialogue box at run time.

Syntax:

<Window Name>.<Alarm Name>.FORMAT;

SCROLLDOWN

This operator is used to scrolls the alarm display one line down.

Syntax:

<Window Name>.<Alarm Name>.SCROLLDOWN;

SCROLLLEFT

This operator is used to scroll the alarm display one column left.

Syntax:

<Window Name>.<Alarm Name>.SCROLLLEFT;



SCROLLUP

This operator is used to scroll the alarm display one line up.

Syntax:

<Window Name>.<Alarm Name>.SCROLLUP;

SCROLLRIGHT

This operator is used to scroll the alarm display one column right.

Syntax:

<Window Name>.<Alarm Name>.SCROLLRIGHT;

SEVERITY

This operator sets the severity level from 0 to 9 for the alarm window. Alarms with greater or equal severity level will be displayed in the alarm window. The severity level for alarm is assigned at the time of defining alarm on a tag.

Syntax:

<Window Name>.<Alarm Name>.SEVERITY = <Numeric Expression>;

SORT

This operator is used to choose a field for sorting the alarms

Syntax:

<Window Name>.<Alarm Name>.SORT;

12.2.13 Script for Reporting:

PRINTTEMPLATE

This operator creates header/footer for all pages of the report. Needs report id, row and column position and the text (Max. up to 255 characters) for creating header/footers for the report.

Syntax:

PrintTemplate <ReportId>, <Row>, <Column>, <Text to be Printed>;

PRINTTEXT

This operator sends the specified text(max. upto 255 characters) to report along with report id, page, row and column position. Tag names for text to be printed can be specified by the user. A non-string tag can be printed by first converting it into a string using STR command.

Syntax:

PrintText <ReportId>, <Page>, <Row>, <Column>, <Text to be Printed>;

PRINTWINDOW

This operator is used to send a window to graphic report. Window contents can be printed on a specified page. Window-name, report-id, page number, left top X and Y positions of graphic on the screen, width, height, left top X and Y positions of graphic on the paper need to be specified.

For printing or sending entire window contents to the file, the values of both bitmap width and height have to be 0.

Syntax:

PrintWindow <Window Name>, <ReportId>, <Page>,<Source Left Top X>,<Source Left Top Y> <Bitmap Width>, <Bitmap Height> <Destination Left Top X>,<Destination LeftTop Y>;



SENDTOPRINTER

This operator is used to send the report to the printer. Both, graphic and text reports can be sent to the printer with a specified report id and with option to print. The options are :

- 1 Print only the new or changed pages (false).
- 2 Print the entire report (true).

Syntax:

SENDTOPRINTER <ReportId>, <SendEntire>;

SENDTOFILE

This operator is used to send a report to the file. Both, graphic and text reports can be sent to the file. Specify the report id, file name, option to append (true) or overwrite(false) the file. The options to send the entire report (true) to the file or only new/changed pages only (false) are to be specified.

Syntax:

SendToFile <ReportId>, <File Name>, <Append>, <SaveEntire>;

DESTROYREPORT

This operator deletes the unwanted report specified by the ReportId.

Syntax:

DESTROYREPORT <ReportId>;

12.2.14 Some Examples of Report Generation:

The set of script commands to generate a report and print it or save it are as under.

A Single Page Text Report to be Saved in a File

Syntax:

PrintText <ReportId>, <Page Number>, <Row>, <Column>, <Text to be Printed>;
SendToFile <ReportId>, <File Name>, <Append>, <SaveEntire>;

A Single Page Text Report to be Printed

Syntax:

PrintText <ReportId>, <Page Number>, <Row>, <Column>, <Text to be Printed>;
SendToPrinter <ReportId>, <File Name>, <SaveEntire>;

A Multiple Paged Text Report to be Saved in a File

Syntax:

PrintText <ReportId>, <Page Number-1>, <Row>, <Column>, <Text to be Printed>;
PrintText <ReportId>, <Page Number-2>, <Row>, <Column>, <Text to be Printed>;
PrintText <ReportId>, <Page Number-n>, <Row>, <Column>, <Text to be Printed>;
SendToFile <ReportId>, <File Name>, <Append>, <SaveEntire>;

Similarly, you can print the multiple paged text report using the SendToPrinter command.



A Text Report with Header is to be Printed

Syntax:

PrintText <ReportId>, <Page Number>, <Row>, <Column>, <Text to be Printed>;
PrintTemplate <ReportId>, <Row>, <Column>, <Header Text>;
SendToPrinter <ReportId>, <File Name>, <SaveEntire>;

If user want to have header 1 for first 2 pages of the report and header 2 for the last three pages of the report.

Syntax:

PrintText <ReportId>, <Page Number-1>, <Row>, <Column>, <Text to be Printed>;
PrintText <ReportId>, <Page Number-2>, <Row>, <Column>, <Text to be Printed>;
PrintTemplate <ReportId>, <Row>, <Column>, <Header1 Text>;
PrintText <ReportId>, <Page Number-3>, <Row>, <Column>, <Text to be Printed>;
PrintText <ReportId>, <Page Number-4>, <Row>, <Column>, <Text to be Printed>;
PrintText <ReportId>, <Page Number-5>, <Row>, <Column>, <Text to be Printed>;
PrintTemplate <ReportId>, <Row>, <Column>, <Header2 Text>;
SendToPrinter <ReportId>, <File Name>, <SaveEntire>;

Similarly, user can save the report in a file using **SendtoFile** command.

Report with Single Bitmap to be Saved in a File

Syntax:

PrintWindow <Window Name>, <ReportId>,<Source Left Top X>,<Source Left Top Y> <Bitmap Width>, <Bitmap Height> <Destination Left Top>, <Destination Left Top Y>;
SendToFile <ReportId>, <File Name>, <SaveEntire>;

Multiple bitmaps can be printed on a single page by entering appropriate Left Top X and Y positions on destination, depending on the sizes of bitmaps.

To print the Entire window, enter the bitmap height and width as 0.



Chapter 13: Recipe

Recipe is a set of process variables. Recipe is typically required for the batch process. Each recipe consists of a group of set point values to be assigned to a set of tags in the process for running a particular batch. In a process there could be 'n' number of recipes. Each recipe representing a different batch of the single process.

In ASTRA, when a recipe is loaded, the corresponding tags in ASTRA are assigned with the values defined in the recipe. On the contrary when a recipe is saved in ASTRA, the corresponding tag values are assigned to those defined in the recipe.

13.1 Creating Recipe File

Recipe can be defined using any text editor (Typically Notepad) in a file called **Recipe.ini**. As a convention, it is advisable to store the recipe.ini file in the root directory of the project. The structure of this file is pre defined in ASTRA and is as follows:

```
[Recipe _name]
variable_name1=value
variable_name2=value
:
variable_name n= value
```

You can create multiple recipes with different names in a single recipe.ini file. A typical example of Recipe is shown below:

```
[Batch-1]
Setpoint_1=35
Setpoint_2=65
Setpoint_3=50
Setpoint_4=50
[Batch-2]
Setpoint_1=45
Setpoint_2=50
Setpoint_3=60
Setpoint_4=65
.
.
.
[Batch-n]
```

Where Batch-1 and Batch-2 are two different Recipes., Setpoint_1, Setpoint_2, Setpoint_3 and Setpoint_4 are the process variables(Tags which need to be defined in ASTRA). Each recipe has a different set of values assigned for these variables.

Note: While configuring a project with recipes, you need add all the variables used in recipe to the tag database. You can use both, the device tags and the memory tags in recipes.



13.2 Using the Recipe File

Once the tag database and the recipe file are ready with one or more recipes, these can be used in the project with the help of the ASTRA script commands, namely LOADRECIPE and SAVERECIPE.

LOADRECIPE:

Loads the user-defined recipe from recipe file to the tags. This command returns the Boolean value depending on the success or failure of load function. Loading may fail if the specified recipe or the file name does not exist. If loading fails event logger displays the message accordingly.

Syntax:

```
LOADSTATUS = LOADRECIPE (RECIPE_FILE_PATHRECIPE_NAME);
```

SAVERECIPE:

Saves the tag values to the user specified recipe in the recipe file. This command returns the Boolean value depending on the success or failure of the save function. For saving the recipe you need to specify the recipe INI file and the recipe name. The file and the section for the specified recipe in that file have to exist, otherwise the saving may fail. If saving fails event logger displays the message accordingly.

Syntax:

```
SAVESTATUS = SAVERECIPE (RECIPE_FILE_PATHRECIPE_NAME);
```

At run-time, when you change the values of recipe parameters, ASTRA will clamp the values depending on the corresponding tag definitions. If the values are clamped, the message is shown accordingly, in event logger.



Chapter 14: Security

To avoid the access of the project to the unauthorised persons, ASTRA provides multilevel **Security**. The security is in the form of Passwords and the Users access levels. The security is available for two modes:

14.1 Configuration Mode Security

In the configuration mode, project can be protected from unauthorised access by assigning a **Password** to the project. To set the password for a project you are configuring, open the project and choose the **File | Change Password** menu option. The **Changing Password** dialogue box appears.



Enter the old password and press OK.

Now, enter the new password in the subsequent dialogue box. You can enter maximum of 6 characters. You can enter any alphabets, numeric character or “=” sign. You can not have spaces or underscore in the password. The password is case sensitive. For example, ‘myproj’ and ‘Myproj’ are treated as different passwords. Press OK. The Changing Password dialogue box appears to confirm the changed password. The new password is activated when the project is saved and re-opened.

Enter the valid password to configure the project. Only the authorised users having a valid password will be able to open the project for configuring.

Note: *If you are entering password for the first time, press OK without entering any password. If you are changing an old password enter the old password and press OK. The Changing Password dialog box appears to enter the new password.*



14.2 Run Mode Security

In the run mode or while the project is running, the project can be protected from the unauthorised access. Run mode security is provided in two ways. These are:

Security can be defined for Window control:

You can prevent users at run-time from controlling main window of project. These controls include closing, minimizing, maximizing and re-sizing a window. Choose the **File | Configure** menu option. The **Project Options** dialogue box appears.

Set the Main Window Controls from this dialogue box. Checking the box for control means you are allowing the user, at run-time, to use that particular window control. In that case the user can close, minimise, maximise or re-size the project window depending on the box is checked or not.

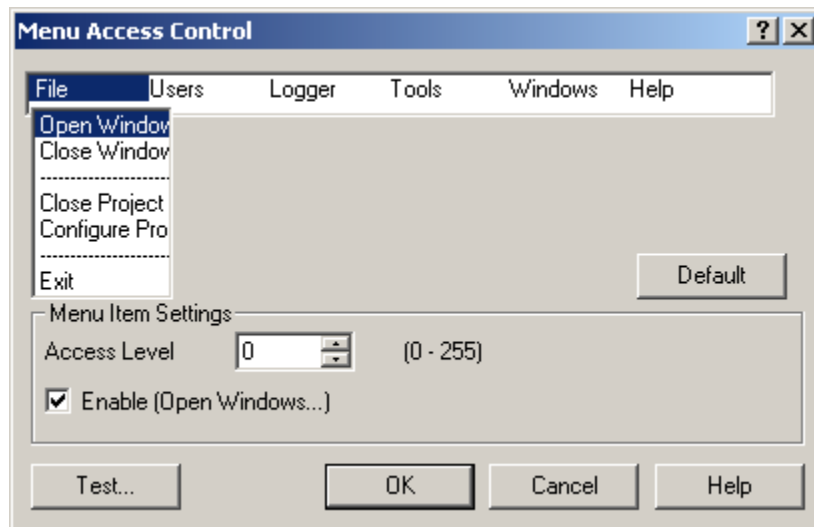
Every action defined can be allotted an access level:

There are 256 Access levels in ASTRA ranging from 0 to 255. An access level of 255 offers maximum security where as 0 (default access level) does not offer any security. You can define the Access levels for:

a Run time menu items

This security is provided to avoid tampering of Run time menu items from unauthorised persons. Following are the steps to assign access levels to the run time menu items:

- 1 Open the project in the configuration mode.
- 2 Click the **Tools | Setup** Menu option. The Menu Access Control dialogue box appears on the screen.
- 3 Select the main menu items and the sub menu items for which an access level is to be entered.
- 4 Enter the access levels.





The Access levels are described below:

| | |
|---------------------|---|
| Enable | Displays the selected Menu and sub menu items at run time. |
| Access Level | Assigns an access level for each enabled menu or selected sub menu item. At run-time only the enabled menu and sub menu items will be displayed. These will be accessible only if the user with equal or higher access levels is logged in. |
| Test | Displays the preview of run-time menu. |
| Default | Sets the default Configuration of run-time menu and default access level. This enables all menu options without any security (0 access level). |

b Each object in the project

Access levels can be defined to various objects in ASTRA. At run mode, these objects can be accessed if the user with an equal or higher access level is logged in. Steps to assign access levels to different objects:

1 Access level for a window

- Select the window and click **Edit | Properties** menu option.
- Enter the access level in the Windows Properties dialogue box.

2 Access level for Alarm Acknowledgement

- Select the Alarm object. Click **Edit | Properties** or press **F4** button to edit the properties of the Alarm object.
- Enter the acknowledgement access level.

3 Access level for entering data through Enter Data or Slider animation

- Select the object for which access level is to be assigned.
- Click **Edit | Animation** options or press **F5** button.
- Attach Enter data (or Slider) animation and enter the access level in the Animation dialogue box.

4 Access level for Take Action animation

- Select the object for which access level is to be assigned.
- Click **Edit | Animation** option.
- Attach Take Action animation.
- Enter the required access level in the animation dialogue box.

Factory

RENU ELECTRONICS PVT. LTD.
Survey Number 2 / 6
Baner Road,
Pune - 411 045, INDIA
Tel: + 91 20 2729 2840
Fax: + 91 20 2729 2839
E-mail: info@renuelectronics.com
Website: www.renuelectronics.com