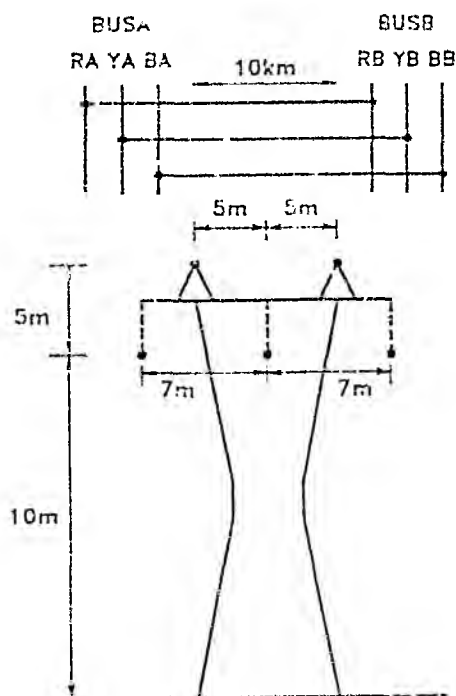


The general structure of an overhead transmission line is as follows:



Element Details - Transmission Line

General Transmission Line Details

Line Length (km): 10

Ground Resistivity (ohm.m): 1000

Calculation Frequency (Hz): 50

Transmission Line Model: Nominal-PI Model

Equivalent-PI Circuit Type: N

Binary Flag:

Continuous Ground Wires? No

Parallel Communication Circuit? No

Line Untransposed? Yes

JANITL Conductor

Transmission Line Conductor 1

Transmission Line Conductor 2

Transmission Line Conductor 3

Transmission Line Conductor 4

Transmission Line Conductor 5

Transmission Line Conductor 6

Transmission Line Conductor 7

Transmission Line Conductor 8

Move mouse over specific entries for extra information about these entries.

OK

CANCEL

HELP

Figure 10.1 - The Transmission Line Element Details Screen

The user is required to complete every entry in the Numerical Details box. These entries are the following:

- **Element Name** - the name of the transmission line associated with this pylon.
- **Line Length** - The length of the transmission line.
- **Ground Resistivity** - Ground Resistivity in Ωm .
- **Calculation Frequency** - The frequency of modelling calculation in Hz (must be between 0.0001 Hz and 500 kHz).
- **Continuous Ground Wires** - This is a combo box which is selected by default. If the ground wires are not continuous, select the Segmented ground combo box (see next entry).

- **Parallel Communication Circuit** - Select this combo box to indicate the presence of a communication circuit parallel to the transmission line (the default is no communication circuit).
- **Line Untransposed** - Select this combo box if the line is not continuously transposed (selecting this combo box automatically implies selection of the MODAL PARAMETERS or MARTI MODEL).

Once all the entries in this screen, select the *Transmission Conductor 1* button to enter the 1st Conductor Screen (see figure 10.2). You may complete as many of the 8 conductors available as you wish. **If more than 5 conductors are selected, it is assumed that 6 conductors are for the three phases, and the icon for the transmission line changes to a 6 conductor transmission line.** This depends on the number of conductors in the transmission line you are modelling. Two of the conductors can be used as earth conductors. For these conductors, select the Conductor Phase Number as 0. To cancel the description of the element, select the *CANCEL* button at any time. To obtain help on the Integrator, select the *HELP* button at any time.

Figure 10.2 - The Transmission Conductor Element Details Screen

Complete the entries in the Conductor screens as follows:

- **Conductor Left Node Name** - This is the node name of the left node of the conductor.
- **Conductor Right Node Name** - This is the node name of the right node of the conductor.
- **Conductor phase number** - This is the phase number of each of the three main conductors. Sub-conductors in a bundle must have the same phase number. Zero is reserved for shielded wires at ground potential during the analysis (for lightning analysis rather treat the shield wires like other phase conductors)
- **Skin Ratio** - the ratio of the thickness of a tubular conductor to its outer diameter - use 0.5 for a solid conductor (all three conductors are assumed to be the same).
- **Conductor resistance** - DC resistance of the conductor in Ω/km (Each conductor has a separated entry).

- **Conductor Diameter** - The outer diameter of the conductor in cm.
- **Horizontal Separation** - Horizontal separation of the centre of the conductor from some arbitrary reference point in metres (it is good practice to put conductor 1 at $-x$ (m), conductor 2 at 0 (m), and conductor 3 at x (m)).
- **Tower Height** - The height of the conductor at the tower in metres.
- **Midspan Height** - The height of the conductor at the midspan in metres.

If the Jmarti model is selected, the JMARTI command button will be enabled. Clicking on this command button will produce the Jmarti screen (see figure 10.3 below)

Element Details - Transmission Line MARTI model

MARTI Model Details:

Diagonalization Frequency (Hz)	5000	OK
Steady State Frequency (Hz)	50	CANCEL
Lowest Frequency (Hz)	0.01	HELP
Number Of Decades	6	
Points Per Decade	9	

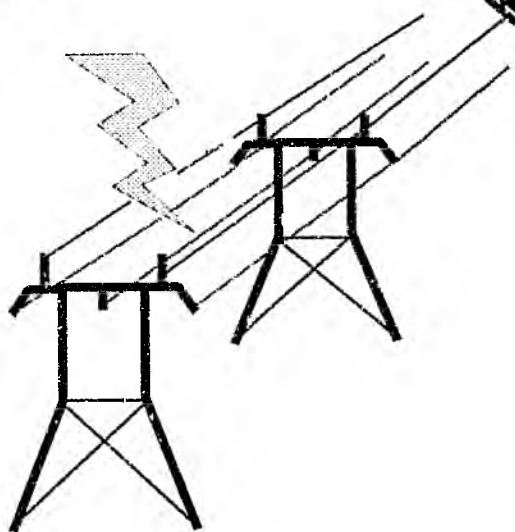
Number of points per decade for the frequency domain description of the transmission line. In ranges below recommended value (10) (eg. 9)

The user is required to enter the following details:

- **Diagonalization Frequency** - The frequency at which diagonalization transformation is to be performed if the line is untransposed, in Hz (recommended value 5000Hz).
- **Steady State Frequency** - The frequency of steady state solution in Hz.
- **Lowest Frequency** - The lowest frequency domain description of the transmission line in Hz (recommended value - 0.01 Hz).
- **Number Of Decades** - The number of decades to be swept in the frequency domain description of the transmission line (recommended value - 9).
- **Points Per Decade** - The number of points per decade for the frequency domain description of the transmission line (recommended value - 10).

Volume 2 - Appendix C

ATP FOR WINDOWS



TECHNICAL REFERENCE MANUAL

ACKNOWLEDGEMENTS

I would like to acknowledge that use was made of 'AN INTRODUCTION TO EMTF (ELECTROMAGNETIC TRANSIENTS PROGRAM)' by J. M. Van Coller (1993) for many of the descriptions of elements used in ATP / data members used in ATP for Windows.

DOCUMENT STATUS CONTROL

Configuration Control

Project / Owner:	Tamir Orbach
Title:	<i>ATP for Windows</i>
Doc. Ref. Number:	ATP002
Version Number:	1.0
Version Release Date:	17 June 1995
Author:	Tamir Orbach
Location and Filename:	C:\OFFICE\WPWIN\WPDOCS\ TRM.001

Chapter Status Control

Chapter	Revision	Date
1	1.0	17/06/95
2	1.0	17/06/95

Document Change History

DCR	Date	Chapter	Page	Change Description / Reason	Old Rev	New Rev
ATP for Windows	01/02/95	All	All			0.1
ATP for Windows	17/06/95	All	All	Program completed, requires update	0.1	1.0

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	ii
DOCUMENT STATUS CONTROL	iii
1 SCOPE	1-7
1.1 Introduction	1-7
1.2 Purpose	1-7
1.3 Audience	1-7
1.4 Applicable Documents	1-7
2 THE ATPWIN PROGRAM UNIT	2-1
2.1 Visual Basic Controls	2-1
2.2 List of Forms	2-2
2.3 The frmATP Form	2-11
2.4 The ATP.BAS Module	2-33
2.5 The frmAbout form	2-45
2.6 The frmARRDAT form	2-46
2.7 The frmBCTRAN form	2-48
2.8 The frmBctranMore form	2-49
2.9 The frmBctranTestData form	2-50
2.10 The frmBlackBox Form	2-52
2.11 The BLACKBOX.BAS Module	2-53
2.12 The frmLeftBottomImages Form	2-55
2.13 The frmBottomImages Form	2-55
2.14 The frmRightBottomImages Form	2-56
2.15 The frmClosingSwitch Form	2-56
2.16 The frmControlledIntegrator Form	2-57
2.17 The frmConstantDelaySwitch Form	2-59
2.18 The frmControllerTopologyElement Form	2-61
2.19 The frmCurrentFluxCharacteristics Form	2-63
2.20 The frmCurrentFluxCharacteristics Form	2-63
2.21 The frmDoubleExponentialFunction Form	2-64
2.22 The DBLEXFNC.BAS Module	2-66
2.23 The frmDCLevel Form	2-68
2.24 The frmDiodeSCR Form	2-69
2.25 The DIODESCR.BAS Module	2-71
2.26 The frmDistributedRLC Form	2-72
2.27 The DISTRRLC.BAS Module	2-75
2.28 The frmDualRampFunction	2-76
2.29 The DUALRAMP.BAS Module	2-78

2.30	The frmEditCircuitPanel Form	2-80
2.31	The EDITCCTP.BAS Module	2-85
2.32	The frmExponSgArr Form	2-86
2.33	The frmFortranElement form	2-88
2.34	The FORTRANE.BAS Module	2-89
2.35	The frmImpedanceElements Form	2-90
2.36	The IMPELIB.BAS Module	2-91
2.37	The frmInputParameters Form	2-92
2.38	The frmJMart Form	2-92
2.39	The frmLevelTriggerredSwitch Form	2-93
2.40	The frmOpeningSwitch Form	2-96
2.41	The frmRampFunction Form	2-97
2.42	The RAMPFUNC.BAS Module	2-99
2.43	The frmReadOutput Form	2-100
2.44	The READOUTP.BAS Module	2-100
2.45	The frmRe'ayOperatedSwitch Form	2-101
2.46	The frmRepetitivePulse Form	2-102
2.47	The frmRepetitiveRamp Form	2-104
2.48	The frmSelectElement Form	2-106
2.49	The frmRightImages Form	2-106
2.50	The SELECELE.BAS Module	2-107
2.51	The frmSetupSimulation Form	2-107
2.52	The SETUPSIM.BAS Module	2-111
2.53	The frmSimpleDifferentiator Form	2-113
2.54	The frmSimpleSources Form	2-114
2.55	The SIMPLLIB.BAS Module	2-115
2.56	The frmSimpleRLC Form	2-115
2.57	The SIMPLRLC.BAS Module	2-117
2.58	The frmSinFunction Form	2-119
2.59	The SINFUNC.BAS Module	2-121
2.60	The frmSpecDevInputParameters Form	2-123
2.61	The frmSinusoidalWaveformGenerator Form	2-123
2.62	The frmSlaveClosingSwitch Form	2-125
2.63	The frmSlaveOpeningSwitch Form	2-127
2.64	The frmSpecialDevice Form	2-129
2.65	The frmSurgeArrestor Form	2-130
2.66	The SRGARR.BAS Module	2-133
2.67	The frmStepFunction Form	2-137
2.68	The STEPFUNC.BAS Module	2-138
2.69	The frmStandardSwitch Form	2-140
2.70	The SWELEMEN.BAS Module	2-142
2.71	The frmSwitchesDiodes Form	2-147
2.72	The SWDIOLIB.BAS Module	2-148
2.73	The frmSwitchElement Form	2-148

2.74	The frmSystematicClosingSwitch Form	2-151
2.75	The frmTACSElementElement Form	2-152
2.76	The TACSELEM.BAS Module	2-154
2.77	The frmTACSElements Form	2-160
2.78	The TACSELIB.BAS Module	2-161
2.79	The frmTACSSources Form	2-161
2.80	The TACSSLIB.BAS Module	2-162
2.81	The frmTACSSource Form	2-163
2.82	The TACSSRCE.BAS Module	2-164
2.83	The frmTACSCcontrolledComponent Form	2-165
2.84	The TCNTRLEL.BAS Module	2-166
2.85	The frmTACSCcontrolledResistor Form	2-167
2.86	The TCNTRRES.BAS Module	2-168
2.87	The frmTACSCcontrolledSource Form	2-169
2.88	The TCNTRSRC.BAS Module	2-170
2.89	The frmTACSCcontrolledSwitch Form	2-171
2.90	The TCNTRSWI.BAS Module	2-172
2.91	The frmTempImages Form	2-173
2.92	The frmLeftTopImages Form	2-173
2.93	The frmTopImages Form	2-174
2.94	The frmTransformer Form	2-174
2.95	The TRNSFRMR.BAS Module	2-177
2.96	The frmTransformer3Phase Form	2-179
2.97	The TRNSFRM3.BAS Module	2-182
2.98	The frmTransmissionConductor Form	2-185
2.99	The frmTransformers Form	2-187
2.100	The TRNSFLIB.BAS Module	2-188
2.101	The frmTransmissionLine Form	2-189
2.102	The TRNSLINE.BAS Module	2-191
2.103	The frmUserDefinedSource Form	2-194
2.104	The USERDFSR.BAS Module	2-196

1 SCOPE

1.1 Introduction

ATP for Windows is a pre-processor to the Alternative Transients Program, which performs the role of a graphical user interface for ATP

ATP for Windows was developed for Microsoft Windows using object-oriented / based programming and design in the Visual Basic (3.0) programming system.

This document presents a complete description of the high and low level program design of ATP for Windows.

1.2 Purpose

The purpose of this document is to provide the reader enough information, about the functionality, needed by the reader to change ATP for Windows, and/or to add functionality to ATP for Windows with relative ease.

1.3 Audience

This document is relevant to anyone involved in the design or development of any part of ATP for Windows.

1.4 Applicable Documents

ATP for Windows User Requirements Specifications, document reference ATP000.

ATP for Windows User Reference Manual, document reference ATP001.

1.4.1 Standards

- a. SEAL (Wits University Software Engineering Applications Laboratory) QMS Document Creation Template, QS 002, Revision 0.01, 5 April 1994.
- b. SEAL QMS Document Layout, Presentation and Typesetting Guide, QS 003, Revision 0.03, 28 June 1994.

2 THE ATPWIN PROGRAM UNIT

This module contains the Visual Basic code and design which is the physical / visible part of ATP for Windows. It contains the entire front end code of ATP for Windows. Since it was written in Visual Basic, it does not contain cpp and h files, but rather .frm, and .bas files. These are explained briefly in the next section.

2.1 Visual Basic Controls

Visual Basic is an event driven, and object based programming language which is ideal for the programming of front ends or interfaces. Visual Basic is based entirely on classes called controls, with each class or control having several regular events which can be overridden, properties (data members) which can be changed, and a number of methods which can also be overridden. Although Visual Basic is not entirely object-oriented, it is object based, which means that by writing code in the events and methods which exist, one is effectively overriding existing methods which are inherited by every instance of the specific control. For a list of all of the controls and their properties, events and methods, see the Visual Basic Language Reference Manual. The most important control in Visual Basic is the form. This control is the general screen, and for every screen that exists in ATP for Windows, there is a corresponding form on which the entire design was performed. The structure of this section involves initially listing all of the forms used. This will be followed by a list of all the controls in each form separately, and then a list of all the methods, events, and properties used in each of the controls. Each form has a corresponding module called a BAS file. Most of the code is placed in the BAS file to speed up the running of the program. The structure of each form is object-oriented, and is similar to the structure of the frmATP form seen in figure 15.

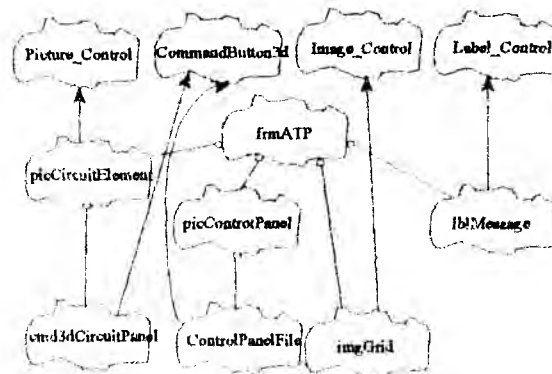


Figure 2 - The frmATP Class Hierarchy

As can be seen in the above figure, each part of the frmATP form inherits its methods, events and properties from the relevant Visual Basic control.

2.2 List of Forms

In all cases in the descriptions below, if the 'Description' section does not describe the element in enough detail, this element is described separately. If a method or event has data members which are seen to be important to the method in a critical way, these will be described in a section following the 'List of Methods' section.

Form Name	File Name	Description
frmATP	ATP.FRM	This is the main screen of the program which includes the Circuit Element Panel, the Control Panel / Toolbar, the main menu, the Work Region, the Message Bar, the Time Bar, the scroll bars for the Work Region, and the mini-map.
frmAbout	ABOUT.FRM	The about form describes the program name and author when the user clicks on the Help - About option.

Form Name	File Name	Description
frmARRDAT	ARRDAT.FRM	This is the Element Details ARRDAT form, which corresponds to the ARRDAT add on program to ATP.
frmBctran	BCTTRAN.FRM	This is the Element Details for which is used to enter the BCTTRAN transformer support program details. This form includes command buttons to call a screen for each winding of the transformer, as well as a command button for more entries, which calls the frmBctranMore screen.
frmBctranMore	BCTTRANMR.FRM	This is the Element Details screen for details needed by the BCTTRAN support program which do not fit in the frmBCTTRAN screen. This screen contains four command buttons, each of which opens a screen for entering data related to the BCTTRAN support ATP support program.
frmBctranTestData	BCTRNTST.FRM	This is the Element Details screen for the test data (such as short circuit test) required by the BCTTRAN ATP support program.
frmBlackBox	BLACKBOX.FRM	This is the Element Details form for the Black Box element.
frmLeftBottomImages	BLEFTIMG.FRM	This form is used to store elements which are in the left bottom area of the entire circuit region (this form is not visible).
frmBottomImages	BOTIMGS.FRM	This form is used to store elements which are in the bottom area of the entire circuit region (this form is not visible).

Form Name	File Name	Description
frmRightBottomImages	BRGHTIMG.FRM	This form is used to store elements which are in the right area of the entire circuit region (this form is not visible).
frmClosingSwitch	CLOSESWI.FRM	This is the Element Details form for the Closing Switch element.
frmConductorSynchMach	CNDPHAS1.FRM	This is the Element Details form for the 3 phases of the synchronous machine.
frmControlledIntegrator	CNTRINTG.FRM	This is the Element Details form for the Controlled Integrator element.
frmConstDelaySlaveSwitch	CONSTSWI.FRM	This is the Element Details form for the Constant Delay Slave Switch element.
frmControlElement	CONTRELE.FRM	This is the Element Details form for the TACS Controller Topology Element.
frmCurrentFluxCharacteristics	CURRFLUX.FRM	This is the Element Details screen for the current-flux entries related to the current-flux characteristic of the transformer element.
frmCurrentVoltageCharacteristics	CURRVOLT.FRM	This is the Element Details screen for the current-voltage entries related to the exponential and piecewise-linear characteristics of a surge arrestor.
frmDoubleExponentialFunction	DBLEXFNC.FRM	This is the Element Details form for the Double Exponential Function element.
frmDCLevel	DCLEVEL.FRM	This is the Element Details form for the DC Level or Single Pulse element.
frmDiode/SCR	DIODESCR.FRM	This is the Element Details form for the Diode / SCR element.

Form Name	File Name	Description
frmDistributedRLC	DISTRRLC.FRM	This is the Element Details form for the Distributed RLC element.
frmDualRampFunction	DUALRFNC.FRM	This is the Element Details form for the Dual Ramp Function element.
frmEditCircuitPanel	EDITCCTP.FRM	This is the Edit Circuit Panel form which along with the corresponding module contains the functionality needed to edit the Circuit Panel.
frmExponSurgeArrestor	EXPSGAR.FRM	This is the Element Details form for the Exponential Modelled Surge Arrestor element.
frmFortranElement	FORTRANE.FRM	This is the Element Details form for the TACS Fortran Relationship element.
frmImpedanceElements	IMPELIB.FRM	This is the form which contains the Impedance Elements Library, and allows the user to select an element from this library and then place it in the circuit.
frmJMarti	JMARTI.FRM	This form is the Element Details form used to enter the details required for the JMARTI support program for ATP (used for the modal parameter JMARTI model of transmission lines).
frmInputParameters	INPUTPRM.FRM	This is the form which contains the input parameter entries of the Controller Topology element.
frmLeftImages	LEFTIMGS.FRM	This form is used to store elements which are in the left area of the entire circuit region (this form is not visible).
frmLevelTriggeredSwitch	LVTRGSWI.FRM	This is the Element Details form for the TACS Level Triggered Switch element.

Form Name	File Name	Description
frmMassCards	MASSCARD.FRM	This form is used to enter element details related to the masses used for the synchronous machine element.
frmOpeningSwitch	OPENSWI.FRM	This is the Element Details form for the Opening Switch element.
frmPrintImages	PRINTIMG.FRM	This form is used to print the circuit. The circuit is copied from the visible circuit region to this form, and the form is then printed.
frmRampFunction	RAMPFUNC.FRM	This is the Element Details form for the Ramp Function element.
frmReadDataFile	READDATA.FRM	This form is used as a text editor which allows the user to view the data files which have been written by ATP for Windows as well as ATP output files.
frmReadOutput	READOUTP.FRM	This is the screen for the menu on <i>Read Output</i> from the <i>RUN</i> sub-menu. It contains the functionality to allow the user to view one or more of the requested variables from the output of the simulation on a graph.
frmRelayOperatedSwitch	RELOPSWI.FRM	This is the Element Details form for the TACS Relay Operated Switch element.
frmRepetitivePulse	REPPULSE.FRM	This is the Element Details form for the TACS Repetitive Pulse element.
frmRepetitiveRamp	REPRAMP.FRM	This is the Element Details form for the TACS Repetitive Ramp element.
frmSelectElement	SELECELE.FRM	This is the form which contains the screen for the <i>Select Element</i> option from the <i>RUN</i> sub-menu.

Form Name	File Name	Description
frmSetupSimulation	SETUPSIM.FRM	This is the form which contains the screen for the <i>Setup Simulation</i> option from the <i>VIEW</i> sub-menu.
frmSimpleDifferentiator	SIMPLDIF.FRM	This is the Element Details form for the TACS Simple Differentiator element.
frmSimpleSources	SIMPLLIB.FRM	This is the form which contains the Simple Sources Library, and allows the user to select an element from this library and then place it in the circuit.
frmSimpleRLC	SIMPLRLC.FRM	This is the Element Details form for the Simple RLC element.
frmSinFunction	SINFUNC.FRM	This is the Element Details form for the Sinusoidal Function element.
frmSpecDevInputParameters	SPECDEV.FRM	This form is used to enter the input parameters for the TACS special devices (U1 - U5).
frmSinusoidalWaveformGenerator	SINWVGEN.FRM	This is the Element Details form for the TACS Sinusoidal Waveform Generator element (TACS Sinusoid).
frmSlaveClosingSwitch	SLCLSSWI.FRM	This is the Element Details form for the Slave Closing Switch element.
frmSlaveOpeningSwitch	SLOPNSWI.FRM	This is the Element Details form for the Slave Opening Switch element.
frmSpecialDevice	SPECDEV.FRM	This is the Element Details form for the TACS Special Device element. It contains functionality to select which special device is required.
frmSurgeArrestor	SRGARR.FRM	This is the Element Details form for the Piecewise-Linear modelled Surge Arrestor element.

Form Name	File Name	Description
frmStepFunction	STEPFUNC.FRM	This is the Element Details form for the Step Function element.
frmStandardSwitch	STNDSW1.FRM	This is the Element Details form for the Standard Switch element.
frmSwitchesDiodes	SWDIOLIB.FRM	This is the form which contains the Switches and Diodes & SCRs Library, and allows the user to select an element from this library and then place it in the circuit.
frmSwitchElement	SWELEMEN.FRM	This is the Element Details form for the Switch element.
frmSynchronousMachine	SYNCHMCH.FRM	Not available yet.
frmSynchMachineMore	SYNCHMRE.FRM	Not available yet.
frmSynchMachTACS	SYNCHTAC.FRM	Not available yet.
frmSystematicClosingSwitch	SYSCLOSW.FRM	This is the Element Details form for the Systematic Closing Switch element.
frmTACSElementElement	TACSELEM.FRM	This is the Element Details form for the TACS element.
frmTACSElements	TACSELIB.FRM	This is the form which contains the TACS Elements Library, and allows the user to select an element from this library and then place it in the circuit.
frmTACSSources	TACSSLIB.FRM	This is the form which contains the TACS Sources Library, and allows the user to select an element from this library and then place it in the circuit.
frmTACSSource	TACSSRCE.FRM	This is the Element Details form for the TACS Source element.
frmTACSCcontrolledComponent	TCNTRLEL.FRM	This is the Element Details form for the TACS Controlled Component element.

Form Name	File Name	Description
frmTACSControlledResistor	TCNTRRES.FRM	This is the Element Details form for the TACS Controlled Resistor element.
frmTACSControlledSource	TCNTRSCR.FRM	This is the Element Details form for the TACS Controlled Source element.
frmTACSControlledSwitch	TCNTRSWI.FRM	This is the Element Details form for the TACS Controlled Switch element.
frmTempImages	TEMPIMGS.FRM	This is a form which contains images that do not appear on the frmATP form, but must be saved (scrolling images). It is never seen by the user.
frmLeftTopImages	TLEFTIMG.FRM	This form is used to store elements which are in the left top area of the entire circuit region (this form is not visible).
frmTolerancesSpecialRequest	TOLERSR.FRM	The tolerances special request card screen (optional screen in the synchronous machine element). Not available yet.
frmTopImages	TOPIMGS.FRM	This form is used to store elements which are in the top area of the entire circuit region (this form is not visible).
frmTransformer	TRANSFRM.FRM	This is the Element Details form for the Single Phase Transformer element.
frmPhaseWinding	TRFWNDG.FRM	This is the Element Details form for the three phases of the three phase transformers (all three windings on the left/primary side).

Form Name	File Name	Description
frmRightPhaseWinding	TRFWNDGR.FRM	This is the Element Details form for the three phases of the three phase transformers (all three windings on the right/secondary side).
frmRightTopImages	TRGHTIMG.FRM	This form is used to store elements which are in the right top area of the entire circuit region (this form is not visible).
frmTransmissionConductor	TRNSCOND.FRM	This is the Element Details form for the Transmission Line conductor. It is used three times for the three transmission line conductors when creating a Transmission Line element.
frmTransformers	TRNSFLIB.FRM	This is the form which contains the Transformers Library, and allows the user to select an element from this library and then place it in the circuit.
frm3PhaseTransformer	TRNSFR3.FRM	This is the Element Details form for the Three Phase Transformer element.
frmTransmissionLine	TRNSLINE.FRM	This is the Element Details form for the Transmission Line element.
frmTransmissionLines	TRNSMLIB.FRM	This is the form which contains the Transmission Lines Library, and allows the user to select an element from this library and then place it in the circuit.
frmUniversalMachine	UNIVMCHN.FRM	This is the Element Details form of the universal machine (Not available yet).
frmUserDefinedSource	USERDFSR.FRM	This is the Element Details form for the User Defined Source element.

Form Name	File Name	Description
frmWinding	WINDING.FRM	This form is used to enter the winding details in the BCTRAN support program for ATP.

2.3 The frmATP Form

2.3.1 Form Description

This form is the main form of the ATP for Windows program, and contains the main menu, the Circuit Element Panel, the Control Panel / Toolbar, the Work Region (circuit drawing area), the Message Bar, the Time Bar, and the mini-map.

2.3.2 List of Form Methods

Method Name	Description
ControlClick (Index As Integer, Menu As String)	This Procedure checks for which element of the menu, the Control Panel Button corresponds to, and then runs the corresponding Click method.

2.3.3 List of Form Data Members and Constants

Member/Const Name	Description
Dim imgSelected As String	This is a variable which stores the name of the image selected when the user clicks using the left mouse button on an element (thus selecting it for editing).
Dim presentIndex As Integer	This is a variable which stores the actual index of the Image Box selected (from the imgGrid Controls) by the user when performing a rotation of an element in the Work Region. This is then used in order to change the Picture property of this Image Box.
Dim StoredTag As Integer	This variable is used to store the tag number of an element which is being dragged.

Member/Const Name	Description
Dim ClickBlocking As Integer	This is an integer variable which is used to indicate that the user is blocking (ie. dragging the mouse while pressing the <SHIFT> key.
Dim ElementDragging As Integer	This variable is an integer variable which is used to indicate that an element is being dragged. This is used to check whether or not an element was being dragged (not from the Circuit Panel) when it is dropped into the circuit. If it was being dragged, the Tag property of the new image Grid entry is set to the StoredTag value.
Dim Blocking As Integer	This variable is used to indicate that a circuit is presently being blocked.
Dim InitialBlockIndex As Integer	This variable is used to store the index (of the image control array called imgGrid) at which the blocking started.
Dim SixConductors As Integer	This variable is used to indicate to the imgGrid_Mousedown and imgGrid_Dragdrop events whether the transmission line being dropped / dragged is a six conductor (+ two neutrals) or a three conductor transmission line.
Dim filesaved As Integer	This is a boolean variable which is set to False as soon as any changes are made to the circuit since the last time it has been saved. It is later checked by the <i>FILE - New</i> , <i>FILE - Exit</i> etc. methods to ensure that the user is not exiting without saving the latest changes made.
Const ENTER = 0	This is a constant which is used in DragOver events to signify the state of the source. If the source is entering, the state is ENTER.
Const LEAVE = 1	This is a constant which is used in DragOver events to signify the state of the source. If the source is leaving, the state is LEAVE.

2.3.4 List of Form Events

Event Name	Description
Form_Load ()	This event is called at the beginning of the program, and is used to load the information required from the .INI file, as well as to initialise all the variables necessary.
Form_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)	This method is used to inform the user that moving the mouse over specific panel options will give the user more information about these buttons.
Form_Unload	This method is used to end the application since the event occurs when the main screen of the program is unloaded (it simply uses the command 'End').
Form_QueryUnload (Cancel As Integer, UnloadMode As Integer)	This event is automatically called before the form is unloaded. It is used to determine whether or not the user has saved the present simulation. If not, the <i>FILE - Save As</i> option is called.

2.3.5 List of Controls

Control Name	Description
menFile	This is the <i>FILE</i> menu, and contains the miFile array of sub-menus.
miFile(9)	This is the array of sub-menus in the <i>FILE</i> menu, and consists of the <i>New</i> , <i>Open</i> , <i>Save</i> , <i>Save As</i> , <i>Import</i> , <i>Print</i> , and <i>Exit</i> options each of which along with the menu separators is an element of this menu array.
menEdit	This is the <i>EDIT</i> menu, and contains the miEdit array of sub-menus.
miEdit(10)	This is the array of sub-menus in the <i>EDIT</i> menu, and consists of the <i>Cut</i> , <i>Copy</i> , <i>Paste</i> , <i>Block</i> , <i>Select All</i> , and <i>Unselect All</i> options each of which along with the menu separators is an element of this menu array.
menView	This is the <i>VIEW</i> menu, and contains the miView array of sub-menus.

Control Name	Description
miView(6)	This is the array of sub-menus in the <i>VIEW</i> menu, and consists of the <i>Show/Hide Control Panel</i> , <i>Edit Circuit Panel</i> , <i>Zoom In</i> , <i>Zoom Out</i> , and <i>Show/Hide Grid</i> options each of which along with the menu separators is an element of this menu array.
menRun	This is the <i>RUN</i> menu, and contains the miRun array of sub-menus.
miRun(7)	This is the array of sub-menus in the <i>RUN</i> menu, and consists of the <i>Select Element</i> , <i>Setup Simulation</i> , <i>Run Simulation</i> , and <i>Read Output</i> options each of which along with the menu separators is an element of this menu array.
menHelp	This is the <i>HELP</i> menu, and contains the miHelp array of sub-menus.
miHelp(5)	This is the array of sub-menus in the <i>HELP</i> menu, and consists of the <i>Help Contents</i> , <i>Name Search</i> , <i>Tutorial</i> and <i>About</i> options each of which along with the menu separators is an element of this menu array.
picControlPanel	This is a picture box on which the Control Panel toolbar is placed. This is needed to make the <i>Show/Hide Control Panel</i> function simpler, and it is generally a recommended practice when placing cmd3d controls.
ControlPanelFile(3)	This is an array of Control Panel command buttons which includes the toolbar buttons for the <i>FILE</i> menu including the <i>New</i> , <i>Open</i> , <i>Save</i> , <i>Import</i> , and <i>Print</i> functions.
ControlPanelEdit(2)	This is an array of Control Panel command buttons which includes the toolbar buttons for the <i>EDIT</i> menu including the <i>Cut</i> , <i>Copy</i> , and <i>Paste</i> functions.
ControlPanelView(3)	This is an array of Control Panel command buttons which includes the toolbar buttons for the <i>VIEW</i> menu including the <i>Edit Circuit Panel</i> , <i>Zoom In</i> , <i>Zoom Out</i> , and <i>Show/Hide Grid</i> functions.

Control Name	Description
ControlPanelRun(1)	This is an array of Control Panel command buttons which includes the toolbar buttons for the <i>RUN</i> menu including the <i>Run Simulation</i> and <i>Read Output</i> functions.
ControlPanelHelp	This is a Control Panel command button which represents the toolbar option for the <i>Help Contents</i> function.
ControlPanelRotate	This is a Control Panel command button which includes the toolbar buttons for the <i>Rotate</i> function.
picCircuitElementPanel	This is a picture box on which the Circuit Element Panel toolbar is placed.
cmd3dCircuitPanel(36)	This is an array of command buttons which are used for the Circuit Element Panel's circuit elements. Each command button has a different icon in its picture property which represents the particular circuit element required.
VScrollCircuitElement	This is the vertical scroll bar on the left of the Circuit Element Panel which allows the user to scroll through the Circuit Element Panel in order to view and select from the entire range of circuit elements available.
imgGrid(129)	This is an array of image boxes which makes up the Work Region (the area on the screen where one can draw a circuit). Each image box is the same size, and the indexes of the image boxes increase from left to right and then from top to bottom.
lineGridHor(8)	This is an array of Line Controls which have their colour set to grey, and their border style set to dotted. They are used to form the horizontal lines in the grid in the Work Region.
lineGridVert(11)	This is an array of Line Controls which have their colour set to grey, and their border style set to dotted. They are used to form the vertical lines in the grid in the Work Region.
HScrollGrid	This is a horizontal scroll bar which is at the bottom of the Work Region, and is used to scroll horizontally through the circuit in the Work Region.

Control Name	Description
VScrollGrid	This is a vertical scroll bar which is at the bottom of the Work Region, and is used to scroll vertically through the circuit in the Work Region.
cmd	This is a Common Dialog control. It is not visible during the running of the program, but is necessary in order to run functions such as Open, Save As, and Print which all use the common dialog box.
picGarbage	This is a picture which is not visible at run-time unless the user drags a circuit element over the garbage bin inside the Circuit Element Panel. The picture box itself contains a picture of a radio-active bin.
imgGarbage	This is an image box which contains the image of a garbage bin. It is situated in the picCircuitElement control. When the user drags an element over this garbage bin and drops the element into the bin, the element is deleted along with all of it's properties.
Timer1	This is a timer control which is required in order to display the time since it has a Timer event which occurs every time the interval specified at design time is passed (1000 ms).
PanelTime1	This is a 3d panel in which lies the PanelTime2 panel.
PanelTime2	This is a 3d panel on which the time is written.
pnlMessage	This is a 3d panel control on which lies the Message Label lblMessage.
lblMessage	This is a label control on which the messages corresponding to the mouse position are written. If the user for example moves the mouse over the Transformer element, the lblMessage.Caption property is automatically set to 'Single Phase Transformer element'.

2.3.6 The miFile Control

2.3.6.1 Control Description

This is the array of sub-menus in the *FILE* menu, and consists of the *New*, *Open*, *Save*, *Save As*, *Import*, *Print*, and *Exit* options each of which along with the menu separators is an element of this menu array.

2.3.6.2 List of Events Used

Event Name	Description
miFile_Click (Index As Integer)	This method checks which of the <i>FILE</i> menu choices were selected, and then calls the relevant function in the ATP.BAS module. This is done using a Select Case statement.

2.3.6.3 Comments

None

2.3.6.4 Revision History

First Revision: 01/02/95

2.3.7 The miEdit Control

2.3.7.1 Control Description

This is the array of sub-menus in the *EDIT* menu, and consists of the *Cut*, *Copy*, *Paste*, *Block*, *Select All*, and *Unselect All* options each of which along with the menu separators is an element of this menu array.

2.3.7.2 List of Events Used

Event Name	Description
miEdit_Click (Index As Integer)	This method checks which of the <i>EDIT</i> menu choices were selected, and then calls the relevant function in the ATP.BAS module. This is done using a Select Case statement.

2.3.7.3 Comments

None

2.3.7.4 Revision History

First Revision: 01/02/95

2.3.8 The miView Control**2.3.8.1 Control Description**

This is the array of sub-menus in the *VIEW* menu, and consists of the *Show/Hide Control Panel*, *Edit Circuit Panel*, *Zoom In*, *Zoom Out*, and *Show/Hide Grid* options each of which along with the menu separators is an element of this menu array.

2.3.8.2 List of Events Used

Event Name	Description
miView_Click (Index As Integer)	This method checks which of the <i>VIEW</i> menu choices were selected, and then calls the relevant function in the ATP.BAS module. This is done using a Select Case statement.

2.3.8.3 Comments

None

2.3.8.4 Revision History

First Revision: 01/02/95

2.3.9 The miRun Control**2.3.9.1 Control Description**

This is the array of sub-menus in the *RUN* menu, and consists of the *Select Element*, *Setup Simulation*, *Run Simulation*, and *Read Output* options each of which along with the menu separators is an element of this menu array.

2.3.9.2 List of Events Used

Event Name	Description
miRun_Click (Index As Integer)	This method checks which of the <i>RUN</i> menu choices were selected, and then calls the relevant function in the ATP.BAS module. This is done using a Select Case statement.

2.3.9.3 Comments

None

2.3.9.4 Revision History

First Revision: 01/02/95

2.3.10 The miHelp Control**2.3.10.1 Control Description**

This is the array of sub-menus in the *HELP* menu, and consists of the *Help Contents*, *Name Search*, *Tutorial* and *About* options each of which along with the menu separators is an element of this menu array.

2.3.10.2 List of Events Used

Event Name	Description
miHelp_Click (Index As Integer)	This method checks which of the <i>HELP</i> menu choices were selected, and then calls the relevant function in the ATP.BAS module. This is done using a Select Case statement.

2.3.10.3 Comments

None

2.3.10.4 Revision History

First Revision: 01/02/95

2.3.11 The ControlPanelFile Control

2.3.11.1 Control Description

This is an array of Control Panel command buttons which includes the toolbar buttons for the *FILE* menu including the *New*, *Open*, *Save*, *Import*, and *Print* functions.

2.3.11.2 List of Events Used

Event Name	Description
ControlPanelFile_Click (Index As Integer)	This method checks which of the <i>FILE</i> toolbar options choices was selected, and then calls the relevant menu's Click event. This is done using a Select Case statement. This is done using TypeMessage method in the ATP.BAS module.
ControlPanelFile_MouseMove (Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)	This method checks which of the <i>FILE</i> toolbar options the mouse is moving over, and hence determines which comment to write to the ToolMessage.Caption property.

2.3.11.3 Comments

None

2.3.11.4 Revision History

First Revision: 01/02/95

2.3.12 The ControlPanelEdit Control**2.3.12.1 Control Description**

This is an array of Control Panel command buttons which includes the toolbar buttons for the *EDIT* menu including the *Cut*, *Copy*, and *Paste* functions.

2.3.12.2 List of Events Used

Event Name	Description
ControlPanelEdit_MouseMove (Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)	This method checks which of the <i>EDIT</i> toolbar options the mouse is moving over, and hence determines which comment to write to the lblMessage.Caption property. This is done using TypeMessage method in the ATP.BAS module.
ControlPanelEdit_Click (Index As Integer)	This method checks which of the <i>EDIT</i> toolbar options choices was selected, and then calls the relevant menu's Click event. This is done using a Select Case statement.

2.3.12.3 List of Relevant Properties

Property Name	Value
Caption	<BLANK>
Name	ControlPanelFile

2.3.12.4 Comments

None

2.3.12.5 Revision History

First Revision: 01/02/95

2.3.13 The ControlPanelView Control

2.3.13.1 Control Description

This is an array of Control Panel command buttons which includes the toolbar buttons for the *VIEW* menu including the *Edit Circuit Panel*, *Zoom In*, *Zoom Out*, and *Show/Hide Grid* functions.

2.3.13.2 List of Events Used

Event Name	Description
ControlPanelView_MouseMove (Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)	This method checks which of the <i>VIEW</i> toolbar options the mouse is moving over, and hence determines which comment to write to the lblMessage.Caption property. This is done using TypeMessage method in the ATP.BAS module.
ControlPanelView_Click (Index As Integer)	This method checks which of the <i>VIEW</i> toolbar options choices was selected, and then call the relevant menu's Click event. This is done using a Select Case statement.

2.3.13.3 List of Relevant Properties

See 5.3.9.3

2.3.13.4 Comments

None

2.3.13.5 Revision History

First Revision: 01/02/95

2.3.14 The ControlPanelRun Control

2.3.14.1 Control Description

This is an array of Control Panel command buttons which includes the toolbar buttons for the *RUN* menu including the *Run Simulation and Read Output* functions.

2.3.14.2 List of Events Used

Event Name	Description
ControlPanelRun_MouseMove (Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)	This method checks which of the <i>RUN</i> toolbar options the mouse is moving over, and hence determines which comment to write to the lblMessage.Caption property. This is done using TypeMessage method in the ATP.BAS module.
ControlPanelRun_Click (Index As Integer)	This method checks which of the <i>RUN</i> toolbar options choices was selected, and then calls the relevant menu's Click event. This is done using a Select Case statement.

2.3.14.3 List of Relevant Properties

See 5.3.9.3

2.3.14.4 Comments

None

2.3.14.5 Revision History

First Revision: 01/02/95

2.3.15 The ControlPanelHelp Control

2.3.15.1 Control Description

This is a Control Panel command button which represents the toolbar option for the *Help Contents* function.

2.3.15.2 List of Events Used

Event Name	Description
ControlPanelHelp_MouseMove (Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)	This method checks which of the <i>HELP</i> toolbar options the mouse is moving over, and hence determines which comment to write to the lblMessage.Caption property. This is done using TypeMessage method in the ATP.BAS module.
ControlPanelHelp_Click (Index As Integer)	This method checks which of the <i>HELP</i> toolbar options choices was selected, and then calls the relevant menu's Click event. This is done using a Select Case statement.

2.3.15.3 List of Relevant Properties

See 5.3.9.3

2.3.15.4 Comments

None

2.3.15.5 Revision History

First Revision: 01/02/95

2.3.16 The ControlPanelRotate Control

2.3.16.1 Control Description

This is an array of Control Panel command buttons which includes the toolbar buttons for the *Rotate Left*, and *Rotate Right* functions.

2.3.16.2 List of Events Used

Event Name	Description
ControlPanelRotate_MouseMove (Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)	This method checks which of the <i>ROTATE</i> toolbar options the mouse is moving over, and hence determines which comment to write to the lblMessage.Caption property. This is done using TypeMessage method in the ATP.BAS module.
ControlPanelRotate_Click (Index As Integer)	This method checks which of the <i>ROTATE</i> toolbar options choices was selected, and then calls the relevant method in the ATP.BAS module. This is done using a Select Case statement in the ControlClick method of the frmATP form.

2.3.16.3 List of Relevant Properties

See 5.3.9.3

2.3.16.4 Comments

None

2.3.16.5 Revision History

First Revision: 01/02/95

2.3.17 The cmd3dCircuitPanel Control

2.3.17.1 Control Description

This is an array of command buttons which are used for the Circuit Element Panel's circuit elements. Each command button has a different icon in its picture property which represents the particular circuit element required.

2.3.17.2 List of Events Used

Event Name	Description
cmd3dCircuitPanel_MouseMove (Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)	This method checks which of the <i>CIRCUIT ELEMENT</i> toolbar options the mouse is moving over, and hence determines which comment to write to the lblMessage.Caption property describing the particular element required. This is done using TypeMessage method in the ATP.BAS module.
cmd3dCircuitPanel_Click (Index As Integer)	This method clears the lblMessage.Caption property by setting it = "".

2.3.17.3 List of Relevant Properties

Property Name	Value and Description
Caption	<BLANK>
DragIcon	This is set to the same picture as the Picture property.
DragMode	Automatic - this makes dragging of the element automatic. When the user clicks on this element, the elements DragIcon appears automatically.
Name	cmd3dCircuitPanel

2.3.17.4 Comments

None

2.3.17.5 Revision History

First Revision: 01/02/95

2.3.18 The imgGrid Control**2.3.18.1 Control Description**

This is an array of image boxes which makes up the Work Region (the area on the screen where one can draw a circuit). Each image box is the same size, and the indexes of the image boxes increase from left to right and then from top to bottom.

2.3.18.2 List of Events Used

Event Name	Description
imgGrid_DragDrop (Index As Integer, Source As Control, X As Single, Y As Single)	This method is used to set the Picture property of the relevant image equal to the DragIcon property of the source.
imgGrid_MouseDown (Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)	This method is used to enable the drag method of the image box selected if the left button of the mouse is used, and to enable and call the Circuit Element Menu if the right mouse button is clicked.

2.3.18.3 List of Relevant Properties

Property Name	Value and Description
DragMode	Manual
Name	imgGrid

2.3.18.4 Comments

None

2.3.18.5 Revision History

First Revision: 01/02/95

2.3.19 The HscrollGrid Control

2.3.19.1 Control Description

This is a horizontal scroll bar which is at the bottom of the Work Region, and is used to scroll

horizontally through the circuit in the Work Region.

2.3.19.2 List of Events Used

Event Name	Description
HscrollGrid_Change ()	This method sets the value of a temporary integer to the value of HScrollGrid at the time of the change, and calls the <i>ScrollHorImage</i> method passing the tempint as the parameter in ATP.BAS

2.3.19.3 List of Relevant Properties

Property Name	Value and Description
Name	HScrollGrid
Value	16383 - This is the initial position of the scroll change. The range is 32766, making 16383 the middle of the range exactly.

2.3.19.4 Comments

None

2.3.19.5 Revision History

First Revision: 01/02/95

2.3.20 The VscrollGrid Control

2.3.20.1 Control Description

This is a vertical scroll bar which is at the bottom of the Work Region, and is used to scroll vertically through the circuit in the Work Region.

2.3.20.2 List of Events Used

Event Name	Description
VScrollGrid_Change ()	This method sets the value of a temporary integer to the value of VScrollGrid at the time of the change, and calls the <i>ScrollVertImage</i> method passing the tempint as the parameter in ATP.BAS

2.3.20.3 List of Relevant Properties

Property Name	Value and Description
Name	VScrollGrid
Value	3383 - This is the initial position of the scroll change. The value is 32766, making 16383 the middle of the scale.

2.3.20.4 Comments

None

2.3.20.5 Revision History

First Revision: 01/02/95

2.3.21 The /mgGrabage Control**2.3.21.1 Control Description**

This is an image box which contains the image of a garbage bin. It is situated in the picCircuitElement control. When the user drags an element over this garbage bin and drops the element into the bin, the element is deleted along with all of its properties.

2.3.21.2 List of Events

Event Name	Description
imgGarbage_DragDrop	This method is used to set the DragIcon's image back to nothing after the element is dropped into the garbage bin, and then to call the DeleteElement method in ATP.BAS which deletes the element and it's properties.
imgGarbage_DragOver (Source As Control, X As Single, Y As Single, State As Integer)	This method is used to change the DragIcon of the source to the picGarbage.Picture property when the element is being dragged into the Garbage image, and back to it's old DragIcon if the element is dragged out of the garbage bin.
imgGarbage_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)	This method is used to write the string "Trash Bin: Delete Element by dragging and dropping into this bin" to the Caption property of the lblMessage control.

2.3.21.3 List of Relevant Properties

None

2.3.21.4 Comments

None

2.3.21.5 Revision History

First Revision: 01/02/95

2.3.22 The Timer1 Control**2.3.22.1 Control Description**

This is a timer control which is required in order to display the time since it has a Timer event which occurs every time the interval specified at design time is passed (1000 ms).

2.3.22.2 List of Events Used

Event Name	Description
Timer1_Timer ()	This method is used to write the correct time to the time panel. The Timer event occurs every time the interval property is reached (see properties).

2.3.22.3 List of Relevant Properties

Property Name	Value
Interval	1000 - this is the interval (in ms) between each Timer event.
Name	Timer1

2.3.22.4 Comments

None

2.3.22.5 Revision History

First Revision: 01/02/95

2.4 The ATP.BAS Module

2.4.1 Module Description

This module is the main module of the ATP for Windows program, and contains the functionality required for the entire frmATP form, and most of the actual program itself.

2.4.2 List Of Methods

Method Name	Description
ATPPassFileSaved (Filesaved As Integer)	This method allows the passing of the filesaved parameter so that it does not have to be a global variable.
BlockElementsComplete (Index As Integer, InitialBlockIndex As Integer)	This method is used to indicate that the blocking of elements is complete it is called from the frmATP form to indicate that the blocking of elements is complete, and the form passes the initial index and the final index of the imgGrid to this method, and they are set to the correct variables for later use.
BlockImage (Index As Integer)	This image is used to indicate to ATP for Windows that an imgGrid has been blocked if there is a picture in it. It changes the imgGrid's borderstyle to FIXED_SINGLE
Function CheckDraggingDisabled () As Integer	This function is used by the frmATP to check if the dragging has been disabled due to a circuit already being in the clipboard.
CheckForPasting (Index As Integer)	This method allows the frmATP.imgGrid_Dragdrop event to check if ATP For Windows can paste a circuit, ie. if the cutting is complete If it is complete, this method changes the borderstyle property of all the imgGrid images back to NONE.
Function CheckIfFileSaves () As Integer	This function checks if the ATPfileSaved variable is true or false if it is true, then it continues with the FileOpen or FileNew functionality. If not, it calls the Save As screen and then continues.

Method Name	Description
Function ConvertString (InputParameter As String) As String	This method is called by every WriteCircuit method in every element's module to check if there is a "." in the string, and if there is not one, this method places the dot in the correct place since ATP requires the dot in every floating point and real number format entry.
DrawSavedBottomPicture (j As Integer, PassedSavedPicture As Integer)	This method is called by every element's OpenCircuit method to draw the required element in the Bottom screen if this was it's saved position.
DrawSavedLeftBottomPicture (j As Integer, PassedSavedPicture As Integer)	This method is called by every element's OpenCircuit method to draw the required element in the Left Bottom screen if this was it's saved position.
DrawSavedLeftPicture (j As Integer, PassedSavedPicture As Integer)	This method is called by every element's OpenCircuit method to draw the required element in the Left screen if this was it's saved position.
DrawSavedLeftTopPicture (j As Integer, PassedSavedPicture As Integer)	This method is called by every element's OpenCircuit method to draw the required element in the Left Top screen if this was it's saved position.
DrawSavedPicture (j As Integer, PassedSavedPicture As Integer)	This method is used to determine what the picture property of each imgGrid picture and then drawing the right picture into the corresponding imgGrid block.
DrawSavedRightBottomPicture (j As Integer, PassedSavedPicture As Integer)	This method is called by every element's OpenCircuit method to draw the required element in the Right Bottom screen if this was it's saved position.
DrawSavedRightPicture (j As Integer, PassedSavedPicture As Integer)	This method is called by every element's OpenCircuit method to draw the required element in the Right screen if this was it's saved position.
DrawSavedRightTopPicture (j As Integer, PassedSavedPicture As Integer)	This method is called by every element's OpenCircuit method to draw the required element in the Right Top screen if this was it's saved position.
DrawSavedTopPicture (j As Integer, PassedSavedPicture As Integer)	This method is called by every element's OpenCircuit method to draw the required element in the Top screen if this was it's saved position.
EditCopy ()	This method copies the image / images into an array of images which is stores for later pasting.

Method Name	Description
EditCut ()	This method cuts the image / images into an array of images which is stores for later pasting.
EditPaste ()	This method pastes the stored images from the Cut or Copy options into the Work Region at the selected spot.
EditSelectAll ()	This method selects all of the elements in the circuit. It is used to zoom in or out of the entire circuit.
EditUnselectAll ()	This method is the reverse of the select all method, and simply unselects all of the elements.
FileExit ()	This method is used to exit from the simulation. It automatically calls checks the Dirtyflag variable to see whether the user has saved the present simulation since the last changes.
FileImport ()	This method opens the Common Dialog - Open screen, and allow the user to select a file to import. This file is then read into the program using the same functionality as the Open command. The images contained in the file are drawn in the Work Region, and the parameters relating to the elements in the circuit are assigned their own variable names. <i>This function is not available yet.</i>
FileNew ()	This simply checks the Dirtyflag variable to see id the user has saved the simulation since the last changes were made, and calls the Save As function is this has not been done, following which it clears the Work Region and the variables associated with the elements in the Work Region. It calls the InitializeAllVariables method to clear the variables associated with the present simulation.
FileOpen ()	This method calls the Common Dialog (Action = 1) to open a file. It then calls the OpenCircuit() method.
FilePrint ()	This method involves printing the circuit in the Work Region to the printer. This is done by calling the relevant Windows API function.

Method Name	Description
FileSave ()	This method is used to save all the parameters of the present circuit as well as the actual diagram of the circuit. This is done by writing each variable to a standard file open for Output. if the file has not been saved before, the SaveAs_Click event is called.
FileSaveAs ()	This method is used to save all the parameters of the present circuit as well as the actual diagram of the circuit. The Common Dialog box is used to prompt the user for the filename (Action = 2). This is done by writing each variable to a standard file open for Output.
Function GetActiveFileName () As String	This function returns the name of the active file name, which is saved in the File - Save As method.
GetPath ()	This method finds the present path.
HelpAbout ()	This method calls up the frmAbout form.
HelpContents ()	This method calls up the Help Contents screen using the Common Dialog control (Action = 6).
HelpNameSearch ()	This method calls up the Help Name Search screen using the Common Dialog control (Action = 6).
HelpTutorial ()	This method calls the frmHelpTutorial form (not available yet).
InitializeAllVariables ()	This method is used to initialise all the variables which need to be initialised at the beginning of a simulation. It is called by the FileNew method, as well as the frmATP Form_Load event.
OpenCircuit ()	This method is used to open a circuit from file. It opens the database file specified in the Open common dialogue box in the File Save As / File Save methods. It then calls each element's OpenCircuit method, to open its own circuit elements, and then closes the database file.

Method Name	Description
PerformBlocking ()	This method uses the initial index and final index of the blocking procedure (which were passed to the ATP.BAS module in the BlockElementsComplete method, and calls the BlockImage method for each element which has been blocked.
RunATP ()	This method runs a DOS shell of ATP. It is called once the data file has been written, and it passes the parameters of the data file, and output file required.
RunRunSimulation ()	This method is called when the user selects the Run-Run Simulation option. The method starts the running of the simulation by calling the WriteCircuit method, which writes the data files required.
RunSelectElement ()	This method is called when the Run-Select Element option is selected. It calls the frmSelectElement.
RunSetupSimulation ()	This method is called when the Run-Setup Simulation option is selected. It calls the frmSetupSimulation.
SaveCircuit ()	This method is called from the File - Save, and File - Save As methods, in order to save the circuit to the database file specified in the Save As dialogue box. The method loops through each of the 9 screens which make up the entire circuit region, and checks each imgGrid element to see if an element exists there. If one does, it calls that element's SaveCircuit method, passing the method the position (screen wise) of the element, and the index (of the imgGrid in that position) of the element. The element's SaveCircuit method then opens the corresponding table in the already open database file, and adds the element to this table.

Method Name	Description
ScrollHorImageGrid (Value As Integer)	This method allows the user to scroll in any horizontal direction. This is done by first storing the left most or right most pictures, and then setting each image in the image grid equal to the image to the left or right of it (depending on which direction was chosen). The Tag property of each imgGrid is passed to the next imgGrid so that the element's module can determine what number the element is (ie. the TAG property is set to the number of the element in the element's FormInitialize method). Also, a variable called PictureHere is used to determine whether a picture exists in a particular spot before that picture is passed to the left or right imgGrid. PictureHere is set when the element is dropped into the imgGrid.
ScrollVertImageGrid (value As Integer)	This method allows the user to scroll in any vertical direction. This is done by first storing the bottom most or top most pictures, and then setting each image in the image grid equal to the image to below or above it (again depending on which direction was chosen). The Tag property of each imgGrid is passed to the next imgGrid so that the element's module can determine what number the element is (ie. the TAG property is set to the number of the element in the element's FormInitialize method). Also, a variable called PictureHere is used to determine whether a picture exists in a particular spot before that picture is passed to the left or right imgGrid. PictureHere is set when the element is dropped into the imgGrid.

Method Name	Description
SelectFormType ()	This method is called when the <i>CIRCUIT ELEMENT - Element Details</i> menu is required. Since every element has a different <i>Element Details</i> screen, this method is needed to call the correct form corresponding to the selected element. This is done by using a Select Case statement checking the Picture property of the Image Grid, and calling the form which corresponds to the picture in this image.
ShowCircuitPosition ()	This method is used every time an element is dropped on the circuit or the screen is scrolled to show the position of all elements on the mini-map.
StartBlocking (Index As Integer)	This method is called by the frmATP_imgGrid_Mousedown event to indicate that blocking has started, and the index of the first position of blocking.
Sub StoreImage (Index As Integer)	This method is used to actually store the images given in the EditCopy and EditCut methods to the frmTempImages form.
TypeMessage (Index As Integer)	This method determines which circuit panel button the mouse is moving over, and types the correct message in the message label.
ViewEditCircuitPanel ()	This method calls the frmEditCircuitPanel form.
ViewRotateElement ()	This method checks whether the element being rotated is a simple RLC or a Single Transmission Line conductor. If it is, then the imgGrid's Picture property is changed to that of the vertical or horizontal component in question.
ViewShowControlPanel ()	This method checks if the Control Panel is presently visible by checking the picControlPanel control's Visible property. If this is true, then it is set to false and if it is false, it is set to true.
ViewShowGrid ()	This method checks if the Grid Lines are presently visible by checking the lineGridVert and lineGridHor line's Visible property. If this is true, then it is set to false and if it is false, it is set to true using a for loop since these elements are an array.

Method Name	Description
WriteARRDATStart ()	This method is called from the WriteCircuit method. It is used to check if any exponentially modelled surge arrestors exist in the circuit. If they do, it calls the SurgeArrestorWriteARRDAT method.
WriteCircuit ()	This method calls each of the separate Write methods which check which elements exist in the circuit, and then call the corresponding WriteCircuit methods.
WriteCircuitBranchCards ()	This method is used to check the circuit for standard branch card elements, and then call the relevant WriteCircuit method in these circuit elements.
WriteCircuitSourceCards ()	This method is used to check the circuit for standard source card elements, and then call the relevant WriteCircuit method in these circuit elements.
WriteCircuitSwitchCards ()	This method is used to check the circuit for standard switch card elements, and then call the relevant WriteCircuit method in these circuit elements.
WriteDistRLCBranchCards ()	This method is used to check the circuit for distributed RLC / single transmission line conductor elements and then call the relevant WriteCircuit method in these circuit elements. This is done separately to other branch cards because of the use of the \$UNITS line before some single transmission line conductors.
WriteEndBranchCards ()	This method is used to write the line ending the Branch cards in the ATP data case.
WriteEndDataCase ()	This method is used to write the lines required for the end of the data case.
WriteEndPlotCards ()	This method is used to write the lines required for the end of the plot cards section in ATP.
WriteEndSourceCards ()	This method is used to write the lines required for the end of the source cards section in ATP.
WriteEndSwitchCards ()	This method is used to write the lines required for the end of the switch cards section in ATP.

Method Name	Description
WriteLineConstants ()	This method checks if any transmission line elements exist in the circuit. If they do, it calls the Transmission line's WriteLineConstants method passing it the position and index of the element (these are used to determine the number of the element).
WriteNodeNames ()	This method is calls the WriteNodeNames method of each element in the circuit, so that all of the node names are written to the Setup Simulation screen when the user selects the 'Selected Nodes' option in the plot outputs options.
WriteSwitchStatisticsCard ()	This method calls each switch element's WriteMiscCircuit method in order to write the statistical miscellaneous cards needed for random switching simulations.
WriteTransformerCards ()	This method is used to check if any transformers exist in the circuit, and if any do, it calls the transformer's WriteCircuit methods. These are again done separately to regular branch cards because of the need for grouping these elements together.
WriteUnitsEnd ()	This method is used to write the end of the \$UNITS commands needed for using the single transmission line conductor.
WriteUserDefinedSourceCards ()	This method is used to write user defined sources to the circuit because these require v-t or i-t cards separately from the source card section.

2.4.3 List of Module Data Members and Constants

Member/Const Name	Description
Global circuit As Database	This variable is opened as the ActiveFileName when the user opens an existing circuit or saves a circuit.
Global PictureHere(129) As Integer	This variable is used to set or check for a picture existing in a specific position in the imgGrid.
Global LeftPictureHere(180) As Integer	This variable is used to set or check for a picture existing in a specific position in the left imgGrid.
Global RightPictureHere(180) As Integer	This variable is used to set or check for a picture existing in a specific position in the right imgGrid.
Global TopPictureHere(182) As Integer	This variable is used to set or check for a picture existing in a specific position in the top imgGrid.
Global BottomPictureHere(182) As Integer	This variable is used to set or check for a picture existing in a specific position in the bottom imgGrid.
Global LeftBottomPictureHere(252) As Integer	This variable is used to set or check for a picture existing in a specific position in the left bottom imgGrid.
Global RightBottomPictureHere(252) As Integer	This variable is used to set or check for a picture existing in a specific position in the right bottom imgGrid.
Global LeftTopPictureHere(252) As Integer	This variable is used to set or check for a picture existing in a specific position in the left top imgGrid.
Global RightTopPictureHere(252) As Integer	This variable is used to set or check for a picture existing in a specific position in the right top imgGrid.
Global picGridIndex As Integer	This variable is used to store the currently selected or pointed to imgGrid image.
Global presentVertScroll As Integer	This variable is used to store the position of the vertical scroll bar.
Global presentHorScroll As Integer	This variable is used to store the position of the horizontal scroll bar.
Global BlackBoxCount As Integer	This variable stores the number of black box elements in the circuit at any time.

Member/Const Name	Description
Global SimpleRLCCount As Integer	This variable stores the number of simple RLC elements in the circuit at any time.
Global DistRLCCount As Integer	This variable stores the number of distributed RLC / single transmission line conductor elements in the circuit at any time.
Global DiodeCount As Integer	This variable stores the number of diodes or SCRs in the circuit at any time.
Global StepFncCount As Integer	This variable stores the number of step functions in the circuit at any time.
Global SinFncCount As Integer	This variable stores the number of sin functions in the circuit at any time.
Global RampFncCount As Integer	This variable stores the number of ramp functions in the circuit at any time.
Global DRampFncCount As Integer	This variable stores the number of dual ramp functions in the circuit at any time.
Global UserDefinedSourceCount As Integer	This variable stores the number of user defined sources in the circuit at any time.
Global DblExpFncCount As Integer	This variable stores the number of double exponential functions in the circuit at any time.
Global TacsSourceCount As Integer	This variable stores the number of TACS sources in the circuit at any time.
Global SwitchElementCount As Integer	This variable stores the number of switch elements in the circuit at any time.
Global SurgeArrestorCount As Integer	This variable stores the number of surge arrestor elements in the circuit at any time.
Global UniversalMachineCount As Integer	This variable stores the number of universal machine elements in the circuit at any time.
Global TransmissionLineCount As Integer	This variable stores the number of transmission line elements in the circuit at any time.
Global TransformerCount As Integer	This variable stores the number of single phase transformers in the circuit at any time.

Member/Const Name	Description
Global TacsElementCount As Integer	This variable stores the number of TACS elements in the circuit at any time.
Global ThreePhaseTransformerCount As Integer	This variable stores the number of three phase transformers in the circuit at any time.
Global SynchMachineCount As Integer	This variable stores the number of synchronous machines in the circuit at any time.
Global GreyBoxCount As Integer	This variable stores the number of grey box elements in the circuit at any time.
Global ActiveDataFilename As String	This variable is the name of the actual data file to which the circuit is being written.
Global Frequency As String	This variable stores the frequency of the simulation.
Global UnitsDefined As Integer	This variable is used to determine whether or not the SUNITS command has been defined.
Global StatsCardWritten As Integer	This variable is used to determine whether or not the switch statistics miscellaneous card has been written.
Dim Cutting As Integer	This is a variable which is used in the EditCut method to indicate that cutting is in progress
Dim DraggingDisabled As Integer	This variable is used to stop the user from dragging an element if a circuit is in the clipboard since both use the same memory.
Dim filesaved As Integer	This variable is used locally within two methods to check whether the file has been saved.
Dim ActiveFileName As String	This is the name of the file which is currently open in ATP for Windows.
Dim GridNotShown As Integer	This variable is used by the ViewShowGrid method to set / check whether the grid is visible or not.
Dim ATPFileSaved As Integer	This method is used to determine whether the file has been saved to disk
Dim blocked As Integer	This variable is used to indicate that the circuit has or has not been blocked.

Member/Const Name	Description
Dim BlockInitialIndex As Integer	This variable stores the initial index of the blocked circuit.
Dim FinalBlockIndex As Integer	This variable stores the final index of the blocked circuit.
Dim ClipBoardRows As Integer	This variable stores the number of rows in of the circuit which has been cut or copied to the clipboard.
Dim CutComplete As Integer	This variable is used to indicate that the cutting process has been completed.

2.5 The frmAbout form

2.5.1 Form Description

The about form describes the program name and author when the user clicks on the Help - About option.

2.5.2 List of Form Events

None

2.5.3 List of Controls

Control Name	Description
lblAbout	This is a Label Control which is the main label on which the string describing the program name and author name are written.
imgEarth	This is an image control containing an image of the Earth symbol.
imgSource	This is an image control containing an image as a sinusoidal source.

Control Name	Description
cmdOK	This is a command button control which has a Caption property set to 'OK'. It has one event overridden. This is the cmdOK_Click event, which unloads the frmAbout form.

2.6 The frmARRDAT form

2.6.1 Form Description

This is the Element Details ARRDAT form, which corresponds to the ARRDAT add on program to ATP. It contains labels and text boxes for every entry required in this screen, as well as command buttons relevant to this screen. Some of the entries for the ARRDAT support program are included in the exponentially modelled surge arrester Element Details screen since there is no option of not using ARRDAT when using such a surge arrester.

2.6.2 List of Form Events

Event Name	Description
Form_Load ()	This method is run as soon as this form is called. It calls the FormARRDATInitialize method in the SRGARR.BAS module.

2.6.3 List of Important Controls

Control Name	Description
lblArrdatDetails	This is the label which has the caption for the heading 'ARRDAT Details'.
cmdCurrentVoltageCharacteristics	This is a command button which calls the frmCurrentVoltageCharacteristics form (which allows the user to enter the current voltage characteristics of the surge arrester).
lblReferenceRMS	This is the label for the txtReferenceRMS entry.

Control Name	Description
txtReferenceRMS	This is a text box for the entry of the RMS voltage rating of the reference arrestor. This entry corresponds to the A1 entry on the ZnO ratings card in the ARRDAT add on program.
lblDesiredRMS	This is the label for the txtDesiredRMS entry.
txtDesiredRMS	This is a text box for the entry of the RMS voltage rating of the desired arrestor. This entry corresponds to the A2 entry in the ZnO ratings card of the ARRDAT add-on program.
lblVoltageScalingFactor	This is the label for the txtVoltageScalingFactor entry.
txtVoltageScalingFactor	This is a text box for the entry of any additional voltage scaling factor required. This entry corresponds to the A3 entry in the ZnO ratings card of the ARRDAT add-on program.
lblCurrentScalingFactor	This is the label for the txtCurrentScalingFactor entry.
txtCurrentScalingFactor	This is a text box for the entry of the current scaling factor. This entry corresponds to the A3 entry in the ZnO ratings card of the ARRDAT add-on program.
lblCurrentMinimum	This is the label for the txtCurrentMinimum entry.
txtCurrentMinimum	This is a text box for the entry of the lowest current associated with the first segment of the arrestor characteristic (cannot be 0). This entry corresponds to the AMIN entry in the ZnO ratings card of the ARRDAT add-on program.
cmdOk	This is a command button control which gives the functionality to create the element. The cmdOK_Click event contains the functionality to call the ArrdatOK method in the SRGARR.BAS file.
cmdCancel	This is a command button control which allows the user to exit from this screen without saving any of the parameters of this arrestor
cmdHelp	This is a command button control which allows the user to obtain on-line help about the specific ARRDAT screen entries.

2.7 The frmBCTRAN form

2.7.1 Form Description

This is the Element Details BCTRAN form, which corresponds to the BCTRAN add on program to ATP. It contains labels and text boxes for every entry required in this screen, as well as command buttons relevant to this screen.

2.7.2 List of Form Events

Event Name	Description
Form_Load ()	This method is run as soon as this form is called. It calls the FormBctranInitialize method in the TRNSFRM ² JAS module.

2.7.3 List of Important Controls

Control Name	Description
txtNumberOfWindings	This is the text box used for the entry of the number of windings in the transformer.
txtRatedFrequency	This is the text box used for the entry of the rated frequency of the transformer
txtIEXPOS	This is a text box used for the entry of the IEXPOS variable (see User Reference Manual).
txtSPOS	This is a text box used for the entry of the SPOS variable (see User Reference Manual).
txtLEXPOS	This is a text box used for the entry of the LEXPOS variable (see User Reference Manual).
txtIEXZERO	This is a text box used for the entry of the IEXZERO variable (see User Reference Manual).
txtSZERO	This is a text box used for the entry of the SZERO variable (see User Reference Manual).
txtLEXZERO	This is a text box used for the entry of the LEXZERO variable (see User Reference Manual).

Control Name	Description
txtExcitationWindingNumber	This is a text box used for the entry of the winding number used for the excitation during testing.
txtMagnetizingBranch	This is a text box used for the entry of the magnetizing branch winding number.
cmbMatrixOutputType	This is a combo box control used to select the matrix output type in the BCTRAN calculations.
cmdOk	This is a command button control which gives the functionality to create the element. The cmdOK_Click event contains the functionality to call the BctranOK method in the TRNSFRM3.BAS file.
cmdCancel	This is a command button control which allows the user to exit from this screen without saving any of the parameters of this arrestor.
cmdHelp	This is a command button control which allows the user to obtain on-line help about the specific Bctran screen entries.

2.8 The frmBctranMore form

2.8.1 Form Description

This is the Element Details BCTRAN More form, which corresponds to more BCTRAN add on entries in the BCTRAN support program to ATP. It contains labels and text boxes for every entry required in this screen, as well as command buttons relevant to this screen.

2.8.2 List of Form Events

Event Name	Description
Form_Load ()	This method is run as soon as this form is called. It calls the FormBctranMoreInitialize method in the TRNSFRM3.BAS module.

2.8.3 List of Important Controls

Control Name	Description
cmdLeftWindingDetails	This is a command button which calls the frmWinding form allowing the user to enter the left winding details.
cmdRightWindingDetails	This is a command button which calls the frmWinding form allowing the user to enter the right winding details.
cmdExtraWindingDetails	This is a command button which calls the frmWinding form allowing the user to enter the extra winding details if there is an extra winding.
cmdTestData	This is a command button which calls the frmBctranTestData form allowing the user to enter the test data in that form.
cmdOk	This is a command button control which gives the functionality to create the element. The cmdOK_Click event contains the functionality to call the BctranMoreOK method in the TRNSFRM3.BAS file.
cmdCancel	This is a command button control which allows the user to exit from this screen without saving any of the parameters of this arrestor.
cmdHelp	This is a command button control which allows the user to obtain on-line help about the specific BctranMore screen entries.

2.9 The frmBctranTestData form

2.9.1 Form Description

This is the Element Details BCTRAN Test Data form, which corresponds to the BCTRAN add on program to ATP. It contains labels and text boxes for every entry required in this screen, as well as command buttons relevant to this screen.

2.9.2 List of Form Events

Event Name	Description
Form_Load ()	This method is run as soon as this form is called. It calls the FormBctranTestDataInitialize method in the TRNSFRM3.BAS module.

2.9.3 List of Important Controls

Control Name	Description
txtI	This is the text box used for the entry of the reference number of short circuit winding 1 of the 2 used. (i in the BCTRAN add on program.
txtJ	This is the text box used for the entry of the reference number of short circuit winding 2 of the 2 used. (j in the BCTRAN add on program.
txtShortCircuitLoss	This is a text box used to enter the short circuit losses in the form.
txtZPOS	This is a text box used for the entry of the ZPOS variable (see User Reference Manual).
txtSPOS	This is a text box used for the entry of the SPOS variable (see User Reference Manual).
txtZZERO	This is a text box used for the entry of the ZZERO variable (see User Reference Manual).
cmdOk	This is a command button control which gives the functionality to create the element. The cmdOK_Click event contains the functionality to call the BctranOK method in the TRNSFRM3.BAS file.
cmdCancel	This is a command button control which allows the user to exit from this screen without saving any of the parameters of this arrestor.
cmdHelp	This is a command button control which allows the user to obtain on-line help about the specific Bctran screen entries.

2.10 The frmBlackBox Form

2.10.1 Form Description

This is the Element Details form for the Black Box element. It includes functionality to load an existing data file into the element description. The data file can then be edited to include only the branch cards. This way these cards will automatically be included in the simulation.

2.10.2 List of Form Events

Event Name	Description
Form_Load ()	This method is run as soon as this form is called. It calls the FormBlackBoxInitialize method in the BLACKBOX.BAS module.

2.10.3 List Of Important Controls

Control Name	Description
lblExistingDataFile	This is the label which is used as a heading for this screen.
txtBlackBox	This is the text box into which the data file is loaded.
txtLeftNodeName	This is the left node name of the black box elements, and it is used by ATP when determining which element this is for the purpose of automatic node naming (not available yet).
txtRightNodeName	This is the right node name of the black box elements, and it is used by ATP when determining which element this is for the purpose of automatic node naming (not available yet).
cmd	This is a Common Dialog control which is needed in order to Open and Save files in this screen.

Control Name	Description
cmdOpen	This Command Button control is used to open an existing data file by calling the BLACKBOX.BAS BlackBoxOpen method.
cmdSave	This Command Button control is used to save the file related to the black box element once it has been edited by calling the BLACKBOX.BAS BlackBoxSave method.
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the information entered in this screen will be automatically associated with this particular black box element. This is done by calling the BlackBoxOK method.
cmdCancel	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.
cmdHelp	This Command Button is used to open the on-line help screen related to the Black Box element and it's properties.

2.11 The BLACKBOX.BAS Module

2.11.1 Module Description

This module is used to perform functionality relating to the frmBlackBox form. It contains a Type structure of the same name as the module with the members listed below (an array of variables of this type are then created with each instance of an element representing one member of this array).

2.11.2 List of Methods

Method Name	Description
BlackBoxOK ()	This method is called by the cmdOK control on the frmBlackBox form when this command button is clicked. The method is used to save the text details and the file name from the text box on the frmBlackBox form to memory.
BlackBoxOpen ()	This method opens the Open Common Dialog box by setting the cmd Action property to 1. The filename is saved to a variable (FileName), and the txtBlackBox Text property is set to the text of the file.
BlackBoxSave ()	This method saves the text file from the txtBlackBox text box to a file. This is done using the Save Common Dialog box by setting the cmd Action property to 2. The filename is saved to a variable (FileName).
BlackBoxSaveCircuit (Position As String, i As Integer)	This method is called by the SaveCircuit method in the ATP.BAS module. It is used to save the Black Box elements to the Black Box Element table in the database currently being used.
BlackBoxWriteCircuit (Position As String, i As Integer)	This method is called by the WriteBranchCards method in the ATP.BAS module. It is used to write the \$INCLUDE line required to include this element.
BlackBoxOpenCircuit ()	This method is called by the Open Circuit method in the ATP.BAS module. It is used to open the Black Box element table in the database file to which the circuit is being saved, and saving all of the black box elements to this table.

2.11.3 List of Module Data Members and Constants

Member / Const Name	Description
Type BLACKBOX LeftNodeName As String RightNodeName As String BlackBoxText As String BlackBoxFileName As String End Type	This type defines the stored variables of the black box element (see BlackBoxes(100)). The BlackBoxFileName is the name of the file which has been inserted in the text box of the Black Box form (see User Reference Manual), while the BlackBoxText is the actual text which was inserted into the text box of the frmBlackBox form.
Dim BlackBoxNumber As Integer	This variable is used to store the number of Black Box elements which have been created in the present simulation.
Dim BlackBoxes(100) As BlackBox	This array defines the actual black box elements which have been created by storing the relevant variables in memory (see Type BlackBox).
Dim TempFilename As String	This variable is used to temporarily store the filename of the data file which is put into the black box text box.

2.12 The frmLeftBottomImages Form

2.12.1 Form Description

This is the screen used to save pictures when the user scrolls to the left bottom hand side of the circuit region. It has 252 (18 across and 14 down) images (in array format) known as the imgTempGrid control array.

2.13 The frmBottomImages Form

2.13.1 Form Description

This is the screen used to save pictures when the user scrolls to the bottom side of the circuit region. It has 182 (13 across and 14 down) images (in array format) known as the imgTempGrid

control array.

2.14 The frmRightBottomImages Form

2.14.1 Form Description

This is the screen used to save pictures when the user scrolls to the right bottom side of the circuit region. It has 182 (13 across and 14 down) images (in array format) known as the imgTempGrid control array.

2.15 The frmClosingSwitch Form

2.15.1 Form Description

This is the Element Details screen for the Closing Switch element. It is used to save the variables which are specific to a Closing Switch element and are not entered in the General Switch Element Details screen.

2.15.2 List of Form Events

Event Name	Description
Form_Load ()	This method is used to call the FormClosingSwitchInitialize method in the SWELEMEN.BAS module as well as to disable the Message Bar in the frmATP form.

2.15. List Of Important Controls

Control Name	Description
lblClosingSwitchDetails	This is a label control which is used as a heading for the screen entries.
lblAverageClosingTime	This is the label for the txtAverageClosingTime text box.

Control Name	Description
txtAverageClosingTime	This is text box control which is used to enter the Average Closing Time of the switch. This entry corresponds to the TAV entry in the Closing Switch data card.
lblStandardDeviationJitterTime	This is the label for the txtStandardDeviationJitterTime text box.
txtStandardDeviationJitterTime	This is a text box control which is used to enter the Standard Deviation of the jitter in the closing time. This entry corresponds to the JSD entry in the Closing Switch data card.
lblOutputType	This is the label for the cmbOutputType combo box.
cmbOutputType	This is a combo box control which is used to select the type of output required from this element. The entries in this combo box are loaded in the FormClosingSwitchInitialize method in the SWELEMEN.BAS module.
cmdOk	This is a command button control which is used to indicate that the element details description is complete. The cmdOK_Click event calls the ClosingSwitchOK method in the SWELEMEN.BAS module.
cmdCancel	This is a command button control which is used to indicate that the user wishes to exit this screen without saving any of the changes.
cmdHelp	This is a command button control which is used to call the on-line help specific to the closing switch element.

2.16 The frmControlledIntegrator Form

2.16.1 Form Description

This is the Element Details form for the Controlled Integrator element.

2.16.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormControlledIntegratorInitialize method in the TACSELEM.BAS module.

2.16.3 List Of Important Controls

Control Name	Description
lblControlledIntegratorDetails	This is a label control which contains the heading for the controlled integrator entries.
lblStatementType	This is the label for the cmbStatementType combo box.
cmbStatementType	This is a combo box control which is used to enter the statement type for this controlled integrator (Corresponds to the ITYPE variable in the controlled integrator card).
lblOutputParameterName	This is the label for the txtOutputParameter text box.
txtOutputParameterName	This is the text control which is used to enter the output parameter name for the controlled integrator (OUTPUT in the controlled integrator card).
lblGain	This is the label for the txtGain text box.
txtGain	This is the text control which is used to enter the gain of the controlled integrator (A in the controlled integrator card).
lblControlValueName	This is the label control for the txtControlValueName text box.
txtControlValueName	This is the text control which is used to enter the name of the parameter which will supply the CONTROL VALUE parameter for the controlled integrator (D in the controlled integrator card).
lblResetName	This is the label control for the txtResetName text box.

Control Name	Description
txtResetName	This is the text control which is used to enter the name of the parameter which will supply the RESET VALUE parameter for the controlled integrator (E in the controlled integrator card).
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this controlled integrator element. This is done by calling the ControlledIntegratorOK method.
cmdCancel	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.
cmdHelp	This Command Button is used to open the on-line help screen related to the Controlled Integrator element and it's properties.
VscrollInputParameter	This is the vertical scroll bar on the right of the Input Parameter frame. It is used to scroll between input parameter entries. The VscrollInputParameter_Change method saves the names of the input parameters on the screen presently into new variables of the InputParameters() array, and changes the Caption property of each of the lblInputParameter labels to indicate the next input parameter.

2.17 The frmConstantDelaySwitch Form

2.17.1 Form Description

This form is used to display the Constant Delay Switch Element Details screen.

2.17.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormConstDelaySlaveSwitchInitialize method in the SWELEMEN.BAS module.

2.17.3 List of Form Controls

Control Name	Description
lblConstDelaySlaveSwitch	This is the label control which is used to write the heading for the constant delay slave switch entries.
lblConstantDelay	This is the label for the txtConstantDelay text box.
txtConstantDelay	This is the text control which is used to enter the constant delay in closing between the master switch and the constant delay slave switch (TCONST in the constant delay slave switch data card).
lblLeftMasterNodeName	This is the label for the txtLeftMasterNodeName text box.
txtLeftMasterNodeName	This is the text control which is used to enter the name of the left node of the master switch (NODE5 in the constant delay slave switch data card).
lblRightMasterNodeName	This is the label for the txtRightMasterNodeName text box.
txtRightMasterNodeName	This is the text control which is used to enter the name of the right node of the master switch (NODE6 in the constant delay slave switch data card).
lblOutputType	This is the label for the txtOutputType text box.
cmbOutputType	This is the combo box control which is used to enter the type of output required by this branch (P in the constant delay slave switch card).

Control Name	Description
cmdOk	This is a command button control which gives the functionality to create the element. The cmdOK_Click event contains the functionality to call the ConstDelaySlaveSwitchOK method in the CONSTSWI.BAS file.
cmdCancel	This is a command button control which allows the user to exit from this screen without saving any of the parameters of this element.
cmdHelp	This is a command button control which allows the user to obtain on-line help about the specific ARRDAT screen entries.

2.18 The frmControllerTopologyElement Form

2.18.1 Form Description

This is the Element Details form for the Controller Topology element.

2.18.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormControllerTopologyElementInitialize method in the CONTRELE.BAS module.

2.18.3 List Of Important Controls

Control Name	Description
lblControllerTopologyElementDetails	This is a label control which contains the heading for the controller topology element entries.
lblOutputParameterName	This is the label for the txtOutputParameter text box.

Control Name	Description
txtOutputParameterName	This is the text control which is used to enter the output parameter name for the controller topology element (OUTPUT in the controller topology element data card).
lblHighestOrder	This is the label control for the txtHighestOrder text box.
txtHighestOrder	This is the text box control which is used to enter the highest order of the denominator for the controller topology element (N in the controller element data card).
lblGain	This is the label for the txtGain text box.
txtGain	This is the text control which is used to enter the gain of the controlled integrator (K in the controller topology element card).
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this controller topology element. This is done by calling the ControlElementOK method.
cmdCancel	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.
cmdHelp	This Command Button is used to open the on-line help screen related to the Controller Topology element and it's properties.
VscrollInputParameter()	This is an array of the vertical scroll bars on the right of the Input Parameter frame and the numerator and denominator coefficients frame. They are used to scroll between input parameter entries or N & D entries. The VscrollInputParameter_Change method saves the names of the input parameters on the screen presently into new variables of the InputParameters() array or the Numerator() & Denominator() arrays, and changes the Caption property of each of the lblInputParameter or lblD & lbl N labels to indicate the next parameter.

2.19 The frmCurrentFluxCharacteristics Form

2.19.1 Form Description

This form is used to enter the current flux characteristics of the single and three phase transformer windings.

2.19.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormCurrentFluxInitialize method in the TRNSFRM3.BAS module.

2.19.3 List Of Important Controls

Control Name	Description
txtCurrent	This is the text box used to enter each of the current entries.
Control Name	Description
txtFlux	This is the text box used to enter each of the flux values of the current flux characteristic.
cmdNext	This command button is used to move to the next entry point of the current flux characteristic.

2.20 The frmCurrentFluxCharacteristics Form

2.20.1 Form Description

This form is used to enter the current voltage characteristics of the exponentially modelled and piecewise linear model surge arrestors.

2.20.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormCurrentVoltageInitialize method in the SRGARR.BAS module.

2.20.3 List Of Important Controls

Control Name	Description
txtCurrent	This is the text box used to enter each of the current entries.
txtVoltage	This is the text box used to enter each of the voltage values of the current voltage characteristic.
cmdNext	This command button is used to move to the next entry point of the current voltage characteristic.

2.21 The frmDoubleExponentialFunction Form**2.21.1 Form Description**

This form is the screen for the Double Exponential Function - Element Details screen.

2.21.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormDoubleExponFunctionInitialize method in the DBLEXFNC.BAS module.

2.21.3 List Of Important Controls

Control Name	Description
lblDoubleExponFuncDetails	This is the label for the double exponential function details section.
lblNodeName	This is the label for the txtNodeName text box.
txtNodeName	This is the text box entry for the Node name of the node which is connected to the source (the other node is connected to earth automatically) (NODE in the double exponential function data card).
lblStartTime	This is the label for the txtStartTime text box.
txtStartTime	This is the text box entry for the start time of the source (TSTART in the double exponential function data card).
lblStopTime	This is the label for the txtStopTime text box.
txtStopTime	This is the text box entry for the stop time of the source (TSTOP in the double exponential function data card).
lblA	This is the label for the txtA text box.
txtA	This is the text box entry for the variable A in the defining equation of the double exponential function (A in the double exponential function data card).
lblB	This is the label for the txtB text box.
txtB	This is the text box entry for the variable B in the defining equation of the double exponential function (B in the double exponential function data card).
lblC	This is the label for the txtC text box.
txtC	This is the text box entry for the variable C in the defining equation of the double exponential function (C in the double exponential function data card).
lblDoubleExponentialType	This is the label for the txtDoubleExponentialType combo box.

Control Name	Description
cmbDoubleExponentialType	This is the combo box for the type of source (current or voltage) (IV in the double exponential function data card).
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this double exponential function element. This is done by calling the DoubleExponFunctOK method.
cmdCancel	This Command Button is used to open the on-line help screen related to the double exponential function element and its properties.
cmdHelp	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.

2.22 The DBLEXFNC.BAS Module

2.22.1 Module Description

This module provides functionality for the frmDoubleExponentialFunction Form. It contains a Type structure of the same name as the module with the members listed below (an array of variables of this type are then created with each instance of an element representing one member of this array).

2.22.2 List of Methods

Method Name	Description
FormDoubleExponFunctInitialize ()	This method is used to initialize the frmDoubleExponentialFunction form by adding the relevant default values of the different entries are also placed in the relevant text boxes or combo boxes in this method.
DoubleExponFunctOK ()	This method is used to save all the different choices and values from the Double Exponential Function - Element Details screen to memory.

Method Name	Description
DoubleExponFuncOpenCircuit ()	This function is called by the OpenCircuit method in the ATP.BAS module. It opens the Double Exponential Function table, and adds the relevant fields to it.
DoubleExponFuncSaveCircuit (Position As String, i As Integer)	This function is called by the SaveCircuit method in the ATP.BAS module. It opens the Double Exponential Function table, and reads the relevant fields from it.
DoubleExponFuncWriteNodeNames (Number As Integer)	This method adds the node names of all the double exponential functions in the circuit to the frmSetupSimulation's lstAvailableNodes list box.
DoubleExponFuncWriteCircuit (Position As String, i As Integer)	This method is by the WriteSourceCard method in the ATP.BAS module. It opens the ActiveDataFileName in Append mode, and writes the relevant line.

2.22.3 List of Data Members and Constants

Member / Const Name	Description
Dim DblExpFncNumber As Integer	This variable keeps track of the number of the double exponential function which is presently active.
Type DoubleExpFnc NodeName As String StartTime As String StopTime As String A As String B As String C As String FunctionType As Integer End Type	This Type allows for the storage of all the details relevant to the double exponential function including the node name, the start time, the stop time, and the coefficients of the defining equation (A, B, and C). To store these an array is required of this type.
Dim DblExpFncs(100) As DoubleExpFnc	This is an array of double exponential functions (DoubleExpFnc) - see above.

2.23 The frmDCLevel Form

2.23.1 Form Description

This is the form for the DC Level or Single Pulse Element Details screen.

2.23.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormDCLevelInitialize method in the TACSSRCE.BAS module.

2.23.3 List Of Important Controls

Control Name	Description
lblDCLevelDetails	This is the label for the DC Level or Single Pulse details section.
lblOutputName	This is the label for the txtOutputName text box.
txtOutputName	This is the text box for the entry of the output name of the DC Level (OUTPUT in the DC Level data card).
lblDCAmplitude	This is the label for the txtDCAmplitude text box.
txtDCAmplitude	This is the text box for the entry of the Amplitude of the DC Level (A in the DC Level data card).
lblStartTime	This is the label for the txtStartTime text box.
txtStartTime	This is the text box for the entry of the start time of the DC Level (T-START in the DC Level data card).
lblStopTime	This is the label for the txtStopTime text box.
txtStopTime	This is the text box for the entry of the stop time of the DC Level (T-STOP in the DC Level data card).

Control Name	Description
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this DC level element. This is done by calling the DCLevelOK method.
cmdCancel	This Command Button is used to open the on-line help screen related to the DC Level element and it's properties.
cmdHelp	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.

2.24 The frmDiodeSCR Form

2.24.1 Form Description

This form is used to show the Diode / SCR Element Details screen.

2.24.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormDiodeInitialize method in the DIODESCR.BAS module.

2.24.3 List Of Important Controls

Control Name	Description
lblSCRDiodeDetails	This is the label used as a heading for the Diode / SCR details.
lblLeftNodeName	This is the label for the txtLeftNodeName text box.
txtLeftNodeName	This is the text box for the entry of the left node name of the diode or SCR (Node1 in the Diode / SCR card).

Control Name	Description
lblRightNodeName	This is the label for the txtRightNodeName text box.
txtRightNodeName	This is the text box for the entry of the right node name of the diode or SCR (Node2 in the Diode / SCR card).
lblVIG	This is the label for the txtVIG text box.
txtVIG	This is the text box for the entry of the VIG or forward threshold voltage of the diode or SCR (VIG in the Diode / SCR card).
lblIHOLD	This is the label for the txIHOLD text box.
txtIHOLD	This is the text box for the entry of the IHOLD or holding current of the diode or SCR (IHOLD in the Diode / SCR card).
lblRecoveryTime	This is the label for the txtRecoveryTime text box.
txtRecoveryTime	This is the text box for the entry of the recovery time of the diode or SCR (TDEION in the Diode / SCR card).
lblGRID	This is the label for the txtGRID text box.
txtGRID	This is the text box for the entry of the GRID or TACS name of the gate signal for the SCR (GRID in the SCR card).
lblOutputType	This is the label for the cmbOutputType combo box.
cmbOutputType	This is the combo box control for the output type required from this branch (P in the Diode / SCR card).
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this Diode or SCR element. This is done by calling the DiodeOK method.
cmdCancel	This Command Button is used to open the on-line help screen related to the Diode or SCR element and it's properties.

Control Name	Description
cmdlHelp	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.

2.25 The DIODESCR.BAS Module

2.25.1 Module Description

This module is used to provide functionality for the frmDiodeSCR form. It contains a Type structure of the same name as the module with the members listed below (an array of variables of this type are then created with each instance of an element representing one member of this array).

2.25.2 List of Methods

Method Name	Description
FormDiodeInit ()	This method is used to initialize the frmDiodeSCR by adding the relevant default values of the different entries are also placed in the relevant text boxes or combo boxes in this method.
DiodeOK ()	This method is used to save all the different choices and values from the Diode or SCR - Element Details screen to memory.
DiodeSaveCircuit (Position As String, i As Integer)	This method opens the Diode table in the presently active database and saves the existing diodes and SCRs to this table.
DiodeOpenCircuit ()	This method opens the Diode table in the presently active database and opens the existing diodes and SCRs from this table.
DiodeWriteNodeNames (Number As Integer)	This method adds the node names of all the diodes in the circuit to the frmSetupSimulation's 1stAvailableNodes list box.

Method Name	Description
DiodeWriteCircuit ()	This method is called by the WriteSwitchCards method in the ATP.BAS module. It writes the relevant cards to the ActiveDataFileName.

2.25.3 List of Data Members and Constants

Member / Const Name	Description
Dim DiodeNumber As Integer	This variable keeps track of the presently active diode or SCR.
Type DiodeType LeftNodeName As String RightNodeName As String VIG As String IHOLD As String RecoveryTime As String GRID As String OutputType As Integer End Type	This Type allows for the storage of all the details relevant to the diode element including the left and right node names, the forward threshold voltage, the holding current, the recovery time of the SCR, the name of the SCR controlling node (GRID), and the type of output required from this element. To use this an array of type DIODE is required.
Dim Diodes(100) As DiodeType	This is an array of diodes (DiodeType) which allows the user to store all of the details in the above type for each diode or SCR element created by the user.

2.26 The frmDistributedRLC Form

2.26.1 Form Description

This form is used to show the Distributed RLC branch screen

2.26.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormDistributedRLCInitialize method in the DISTRRLC.BAS module.

2.26.3 List Of Important Controls

Control Name	Description
lblDistributedRLCDetails	This is the label control which is used for the heading of the RLC details.
lblLeftNodeName	This is the label for the txtLeftNodeName text box.
txtLeftNodeName	This is the text box for the entry of the left node name of the RLC Branch (Node1 in the distributed RLC card).
lblRightNodeName	This is the label for the txtRightNodeName text box.
txtRightNodeName	This is the text box for the entry of the right node name of the distributed RLC Branch (Node2 in the Distributed RLC card).
lblNode3Name	This is the label for the txtNode3Name text box.
txtNode3Name	This is the text box for the entry of the left node name of a similar RLC Branch (Node3 in the distributed RLC card).
lblNode4Name	This is the label for the txtNode4Name text box.
txtNode4Name	This is the text box for the entry of the right node name of a similar distributed RLC Branch (Node4 in the Distributed RLC card).
lblModalResistance	This is the label for the txtModalResistance text box.
txtModalResistance	This is the text box for the entry of the modal resistance of the distributed RLC Branch (RESIS in the Distributed RLC card).
lblModalCapacitance	This is the label for the txtModalCapacitance text box.
txtModalCapacitance	This is the text box for the entry of the modal capacitance of the distributed RLC Branch (PARB in the Distributed RLC card).
lblModalInductance	This is the label for the txtModalInductance text box.
txtModalInductance	This is the text box for the entry of the modal inductance of the distributed RLC Branch (PARA in the Distributed RLC card).

Control Name	Description
lblLineDistance	This is the label for the txtLineDistance text box.
txtLineDistance	This is the text box for the entry of the line distance of the distributed RLC Branch (DIST in the Distributed RLC card).
lblCapacitanceInput	This is the label for the cmbCapacitanceInput combo box.
cmbCapacitanceInput	This is the combo box for the entry of the capacitance input units (COPT on the Frequency Miscellaneous card).
lblInductanceInput	This is the label for the cmbInductanceInput combo box.
cmbInductanceInput	This is the combo box for the entry of the inductance input units (XOPT on the Frequency Miscellaneous card).
lblLineTransposed	This is the label for the cmbLineTransposed combo box.
cmbLineTransposed	This is the combo box for the line transposed flag (IPOSE on the Frequency Miscellaneous card).
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this distributed RLC element. This is done by calling the DistributedRLCOK method.
cmdCancel	This Command Button is used to open the on-line help screen related to the distributed RLC element and it's properties.
cmdHelp	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.

2.27 The DISTRRLC.BAS Module

2.27.1 Module Description

This module is used to provide functionality for the frmDistributedRLC form. It contains a Type structure of the same name as the module with the members listed below (an array of variables of this type are then created with each instance of an element representing one member of this array).

2.27.2 List of Methods

Method Name	Description
FormDistributedRLCInitialize ()	This method is used to initialize the frmDistributedRLC by adding the relevant default values of the different entries are also placed in the relevant text boxes or combo boxes in this method.
DistributedRLCOK ()	This method is used to save all the different choices and values from the Distributed RLC - Element Details screen in memory.
DistributedRLCOpenCircuit ()	This method is called by the OpenCircuit method in the ATP.BAS module. It opens the Distributed RLC element table in the ActiveFileName database file, and obtains the distributed RLC elements and details from this table.
DistributedRLCSaveCircuit (Position As String, i As Integer)	This method is called by the SaveCircuit method in the ATP.BAS module. It opens the Distributed RLC element table in the ActiveFileName database file, and writes the distributed RLC elements and details to this table.
DistributedRLCWriteNodeNames (Number As Integer)	This method adds the node names of all the distributed RLC elements in the circuit to the frmSetupSimulation's lstAvailableNodes list box.
DistributedRLCWriteCircuit ()	This method is called from the WriteDistRLCBranchCards method in the ATP.BAS module. It writes the required lines to the ActiveDataFileName.

2.27.3 List of Data Members and Constants

Member / Const Name	Description
Dim DistRLCNumber As Integer	This element keeps track of the currently active distributed RLC element.
Type DISTRIBUTEDRLC LeftNodeName As String RightNodeName As String Node3Name As String Node4Name As String ModalResistance As String ModalInductance As String ModalCapacitance As String LineDistance As String ModalParametersType As Integer PhaseNumber As String NumberOfPhases As String End Type	This Type allows for the storage of all the details relevant to the distributed RLC element including the left and right node names, Node3Name, Node4Name, the modal resistance, modal capacitance, modal inductance, line distance, and type of modal parameters required. To use this an array of type DistributedRLC is required.
Dim DRLCs(100) As DISTRIBUTEDRLC	This is an array of distributed RLC elements which allows the user to store the details of all the distributed RLC elements created.

2.28 The frmDualRampFunction

2.28.1 Form Description

This form is the screen for the Double Exponential Function - Element Details screen.

2.28.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormDualRampFunctionInitialize method in the DUALRAMP.BAS module.

2.28.3 List Of Important Controls

Control Name	Description
lblDualRampFunctionDetails	This is the label for the dual slope ramp function details section.
lblNodeName	This is the label for the txtNodeName text box.
txtNodeName	This is the text box entry for the Node name of the node which is connected to the source (the other node is connected to earth automatically) (NODE in the dual slope ramp function data card).
lblStartTime	This is the label for the txtStartTime text box.
txtStartTime	This is the text box entry for the start time of the source (TSTART in the dual slope ramp function data card).
lblStopTime	This is the label for the txtStopTime text box.
txtStopTime	This is the text box entry for the stop time of the source (TSTOP in the dual slope ramp function data card).
lblAmplitude	This is the label for the txtAmplitude text box.
txtAmplitude	This is the text box entry for the Amplitude of the dual slope ramp function (A in the double exponential function data card).
lblPeakValueTime	This is the label for the txtPeakValueTime text box.
txtPeakValueTime	This is the text box entry for the time at which the peak value is reached in the dual ramp function (TIME-0 in the dual slope ramp function data card).
lblTailValueTime	This is the label for the txtTailValueTime text box.
txtTailValueTime	This is the text box entry for the time taken to reach the tail value of the dual slope ramp function (TIME-1 in the dual slope ramp function data card).
lblTailValue	This is the label for the txtTailValue text box.
txtTailValue	This is the text box entry for the tail value of the dual slope ramp function (TAILVAL in the dual slope ramp function data card).

Control Name	Description
lblDualRampFunctionType	This is the label for the txtDualRampFunctionType combo box.
cmbDualRampFunctionType	This is the combo box for the type of source (current or voltage) (IV in the dual slope ramp function data card).
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this dual slope ramp function element. This is done by calling the DualRampFuncOK method.
cmdCancel	This Command Button is used to open the on-line help screen related to the dual slope ramp function element and it's properties.
cmdHelp	This Command button control is used to Cancel this operation without saving any of the details to file or memory.

2.29 The DUALRAMP.BAS Module

2.29.1 Module Description

This module provides functionality for the frmDualRampFunction Form. It contains a Type structure of the same name as the module with the members listed below (an array of variables of this type are then created with each instance of an element representing one member of this array).

2.29.2 List of Methods

Method Name	Description
FormDualRampFuncInitialize ()	This method is used to initialize the frmDualRampFunction form by adding the relevant default values of the different entries are also placed in the relevant text boxes or combo boxes in this method.

Method Name	Description
DualRampFunctionOK ()	This method is used to save all the different choices and values from the Dual Slope Ramp Function - Element Details screen to memory.
DrampFuncOpenCircuit ()	This method is called by the OpenCircuit method in the ATP.BAS module. It opens the Dual Ramp Function table in the ActiveFileName database file, and obtains the dual ramp functions details from this table.
DrampFuncSaveCircuit (Position As String, i As Integer)	This method is called by the SaveCircuit method in the ATP.BAS module. It opens the Dual Ramp Function table in the ActiveFileName database file, and writes the dual ramp function details to this table.
DRampFuncWriteNodeNames	This method adds the node name of all the dual ramp functions in the circuit to the frmsSetupSimulation's lstAvailableNodes list box.
DualRampFunctionWriteCircuit ()	This method is called from the WriteSourceCards method in the ATP.BAS module. It writes the required lines to the ActiveDataFileName.

2.29.3 List of Data Members and Constants

Member / Const Name	Description
Dim DrampFuncNumber As Integer	This variable keeps track of the number of the dual ramp functions which is presently active.
Type DualRampFunc NodeName As String StartTime As String StopTime As String Amplitude As String PeakValueTime As String TailValue As String TailValueTime As String FunctionType As Integer End Type	This Type allows for the storage of all the details relevant to the dual ramp function including the node name, the start time, the stop time, the amplitude, the peak value time, the tail value time, the tail value, and the type of function required. To store these an array is required of this type.

Member / Const Name	Description
Dim DrampFncs (100) As DualRampFnc	This array of dual ramp functions allows the user to store the details relevant to this element for each dual ramp function in the circuit.

2.30 The frmEditCircuitPanel Form

2.30.1 Form Description

This form is used to show the Edit Circuit Panel screen.

2.30.2 List of Form Data Members and Constants

Member / Const Name	Description
Dim ExistingElementCount As Integer	This is a variable which is used to store the number of existing elements in the program.
Dim PanelSettingsCount As Integer	This is a variable which is used to store the number of settings which have been created. The default is 1 (ATP Default setting).
Dim PanelSettings() As CircuitPanelSetting	This is the array of actual circuit element settings. The CircuitPanelSetting type is in the EDITCCTP.BAS module.
Dim ElementDragAllowed As Integer	This is a boolean variable which is set to True to indicate that drag and drop between the forms list boxes is allowed. Drag and drop is only allowed once a setting has been selected, and the <i>Edit Settings</i> option has been selected.
Dim SelectSetting As Integer	This is a boolean variable which is set to true when a panel setting has been selected.

2.30.3 List Of Important Controls

Control Name	Description
lblExistingPanelSettings	This is the label for the heading of the Panel Settings list box.
lstPanelSettings	This is a list box control which is filled with the list of panel settings which have been created up to now.
lblPresentPanelSetting	This is the label for the heading of the Present Panel Settings text box.
txtPresentPanelSetting	This is the text box in which the present panel setting is listed.
lblExistingElements	This is the label for the heading of the Existing Elements list box.
lstExistingElements	This is a list box control which is filled with the list of existing ATP for Windows circuit elements. The user is then capable of dragging and dropping elements from this box to the lstSelectedPanelElements.
lblSelectSetting	This is a label control which is used in a similar fashion to a standard command button. It is used to select a setting from the Panel Settings List.
lblEditSetting	This is a label control which is used in a similar fashion to a standard command button. It is used to edit a setting from the Panel Settings List. This option can only be selected after a setting has been selected using the select setting option.
lblCreateSetting	This is a label control which is used in a similar fashion to a standard command button. It is used to create a new setting for the Panel Settings List.
lblDeleteSetting	This is a label control which is used in a similar fashion to a standard command button. It is used to delete a setting from the Panel Settings List.
lblDeleteElement	This is a label control which is used in a similar fashion to a standard command button. It is used to delete an element from an existing set of elements for a specific panel setting.

Control Name	Description
ElementIcon	This is an icon control which has it's visible property initially set to False. It is then used to indicate that the user is dragging a file in an area in which this is allowed.
NoIcon	This is an icon control which has it's visible property initially set to False. It is then used to indicate that the user is dragging a file in an area in which this is not allowed.
lblSelectedPanelElements	This is the label for the Selected Panel Elements list.
lstPanelElements	This is a list box control which is filled with the list of selected ATP for Windows circuit elements for this particular setting.
cmdOk	This is a command button which is used to indicate that the user has completed the process of editing the Circuit Panel.
cmdCancel	This is a command button which is used when the user wishes to exit from this screen without saving any of the changes made while in this screen.
cmdHelp	This is a command button which is used when the user wishes to get on-line help specific to this screen.

2.30.4 The lblSelectSetting Control

2.30.4.1 Control Description

This is a label control which is used in a similar fashion to a standard command button. It is used to select a setting from the Panel Settings List.

2.30.4.2 List of Events Used

Event Name	Description
lblSelectSetting_Click ()	This method is used to check which of the settings in the Existing Panel Settings list is active, and to set the ActivePanelSetting variable to this setting.

2.30.5 The lblEditSetting Control**2.30.5.1 Control Description**

This is a label control which is used in a similar fashion to a standard command button. It is used to edit a setting from the Panel Settings List. This option can only be selected after a setting has been selected using the select setting option.

2.30.5.2 List of Events Used

Event Name	Description
lblEditSetting_Click ()	This method is used to edit a setting once it has already been selected. The user cannot edit the ATP Default Setting, and a message box warning the user of this is shown if the user selects the Edit Setting option when the ATP Default Setting has been selected. When a valid setting has been selected, this event clears all of the available elements from the Circuit Element Panel, and then stores whichever elements are dragged into the Selected Panel Elements list into the this particular PannelSettings() array element.

2.30.6 The lblCreateSetting Control**2.30.6.1 Control Description**

This is a label control which is used in a similar fashion to a standard command button. It is used to create a new setting for the Panel Settings List.

2.30.6.2 List of Events Used

Event Name	Description
lblCreateSetting_Click ()	This method is used to create a new setting. This is done by opening an InputBox to the user, and prompting the user for the name of the new setting. This name is then given as the .Name variable in the PannelSettings() array element. The setting initially starts with no associated selected elements at all.

2.30.7 The lblDeleteSetting Control**2.30.7.1 Control Description**

This is a label control which is used in a similar fashion to a standard command button. It is used to delete a setting from the Panel Settings List.

2.30.7.2 List of Events Used

Event Name	Description
lblDeleteSetting_Click ()	This method is used to delete a new setting. This is done by simply erasing the particular array element selected. A MsgBox is shown before this is done to ensure that the user would like to erase the particular setting selected.

2.30.8 The lblDeleteElement Control**2.30.8.1 Control Description**

This is a label control which is used in a similar fashion to a standard command button. It is used to delete an element from an existing set of elements for a specific panel setting.

2.30.8.2 List of Events Used

Event Name	Description
lblDeleteElement_Click ()	This method is used to delete an element from the Selected Elements list. This is done by simply erasing the particular element selected from the elements associated with the setting. A MsgBox is shown before this is done to ensure that the user would like to erase the particular element selected.

2.31 The EDITCCTP.BAS Module

2.31.1 Module Description

This module is used to provide functionality for the frmEditCircuitPanel form.

2.31.2 List of Methods

Method Name	Description
SetupNewCircuitPanel (CircuitElement As String)	This method inserts each new element which is selected for a new setting. This method is called every time an element is dropped in the Panel Elements list.

2.31.3 List of Data Members and Constants

Member / Const Name	Description
PanelElements(100) As String	This is an array of elements which is stored for every Circuit Element Setting. It is in the CircuitPanelSetting type.
Name As String	This is the name of the Circuit Element Panel Setting stored for each setting. It is in the CircuitPanelSetting type.
ElementCount As Integer	This is a count of the number of elements in the setting created. It is in the CircuitPanelSetting type.

2.32 The frmExponSgArr Form

2.32.1 Form Description

This form is used to show the Exponentially modelled surge arrestor Element Details screen.

2.32.2 List of Events

Event Name	Description
Form_Load ()	This form is used to call the FormExponSgArrInitialize method in the SRGARR.BAS module.

2.32.3 List Of Important Controls

Control Name	Description
lblArrestorDetails	This label control is used to write the heading for the Arrestor Details section.
lblLeftNodeName	This is the label for the txtLeftNodeName text box.
txtLeftNodeName	This is the text box for the entry of the left node name of the Exponentially modelled surge arrestor (Node1 in the exponentially modelled surge arrestor card).
lblRightNodeName	This is the label for the txtRightNodeName text box.
txtRightNodeName	This is the text box for the entry of the right node name of the exponentially modelled surge arrestor (Node2 in the Exponentially modelled surge arrestor card).
lblNode3Name	This is the label for the txtNode3Name text box.
txtNode3Name	This is the text box for the entry of the left node name of a similar Exponentially modelled surge arrestor (Node3 in the exponentially modelled surge arrestor card).
lblNode4Name	This is the label for the txtNode4Name text box.

Control Name	Description
txtNode4Name	This is the text box for the entry of the right node name of a similar exponentially modelled surge arrester (Node4 in the Exponentially modelled surge arrester card).
lblFlashoverVoltage	This is the label for the txtFlashoverVoltage text box.
txtFlashoverVoltage	This is the text box for the entry of the flashover voltage of the exponentially modelled surge arrester (VFLASH in the Exponentially modelled surge arrester card).
lblVREF	This is the label for the txtVREF text box.
txtVREF	This is the text box for the entry of the VREF in the defining function of the exponentially modelled surge arrester (VREF in the Exponentially modelled surge arrester card).
lblNoOfSegments	This is the label for the txtNoOfSegments text box.
txtNoOfSegments	This is the text box for the entry of the number of segments of the exponentially modelled surge arrester.
lblOutputType	This is the label for the cmbOutputType combo box.
cmbOutputType	This is the combo box for the entry of the output type required from this branch of the simulation (P in the Exponentially modelled surge arrester card).
txtMaximumRelativeError	This is the text box for the relative maximum error.
txtNumberOfPhases	This is the text box for the number of phases.
txtReferenceRMS	This is the text box for the reference RMS entry.
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this exponentially modelled surge arrester element. This is done by calling the ExponSgArrOK method.

Control Name	Description
cmdCancel	This Command Button is used to open the on-line help screen related to the exponentially modelled surge arrestor element and it's properties.
cmdHelp	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.

2.33 The frmFortranElement form

2.33.1 Form Description

This form is used to display the Fortran Relationship Element - Element Details screen.

2.33.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormFortranElementInitialize method in the FORTRANE.BAS module.

2.33.3 List Of Important Controls

Control Name	Description
lblFortranRelationships	This is the label control used for the heading of the txtFortranRelationships text box.
txtFortranRelationships	This text box is a multi-line text box which is used to enter the Fortran relationships associated with this element row by row. Each row is then written as a separate card in the TACS section of the data case.
lblFortranCommands	This is the label control used for the heading of the txtFortranCommands text box.

Control Name	Description
txtFortranCommands	This text box is a multi-line text box which is used to list the available Fortran commands which the user can then click on and insert into the Fortran Relationships section.
cmdInsertCommand	This command button is used to insert commands selected in the Fortran Commands list, into the Fortran Relationships section at the selected spot. This command button makes use of the Click () event to do this.
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this Fortran Relationship element. This is done by calling the FortranElementOK method.
cmdCancel	This command button is used to exit from this screen without saving any of the changes made.
cmdHelp	This Command Button is used to open the on-line help screen related to the Fortran Relationship element and it's properties.

2.34 The FORTRANE.BAS Module

2.34.1 Module Description

This module is used to provide functionality for the frmFortranElement form. It contains a Type structure of the same name as the module with the members listed below (an array of variables of this type are then created with each instance of an element representing one member of this array).

2.34.2 List of Methods

Method Name	Description
FormFortranElementInitialize ()	This method is used to initialize the frmFortranElement by adding the relevant default values of the different entries are also placed in the relevant text boxes or combo boxes in this method.
FortranElementOK ()	This method is used to save all the different choices and values from the Fortran Relationship Element - Element Details screen to memory.
FortranElementWriteAllValues ()	This method is called by the RunRur Simulation method in order to write the relevant exponential model at surge arrestor details to card. This method calls the relevant methods in the Fortran_Rel_Elem class in the Data Case Module.

2.34.3 List of Data Members and Constants

Member / Const Name	Description
Dim FortranStatements() As String	This array of variables stores all of the Fortran Relationships relating to this particular element.

2.35 The frmImpedanceElements Form**2.35.1 Form Description**

This form is used to display the Impedance Elements Library screen.

2.35.2 List of Form Events

None

2.35.3 List Of Important Controls

Control Name	Description
lblLibraryElements	This is the label control used as a heading for the lstLibraryElements list box.
lstLibraryElements	This is a list box control which is used to list the impedance elements and surge arrestors available in ATP for Windows.
cmdOk	This command button is used to indicate that the user has selected the element required, and would like to place it in the circuit. This command button's Click () event calls the ImpShowElementIcon method in the IMPELIB.BAS module.
cmdCancel	This command button is used to indicate that the user has decided not to select an element from this library, and to exit without any changes.
cmdHelp	This command button is used to get on-line help on the Impedance Elements and Surge Arrestors Library.

2.36 The IMPELIB.BAS Module**2.36.1 Module Description**

This module is used to provide functionality for the frmImpedanceElements form.

2.36.2 List of Methods

Method Name	Description
ImpShowElementIcon ()	This method checks which element was selected from the Element List and selects the correct icon for this element.

2.36.3 List of Data Members and Constants

None

2.37 The frmInputParameters Form

2.37.1 Form Description

This form is user to enter the input parameters of the Controller Topology element. The inputs include U1-U5, N0-N7, and D0-D7. The frmSpecDevInputParameters is user for the relay operated switch, the level triggered switch, the simple differentiator, and the controlled integrator.

2.37.2 List of Form Events

Event Name	Description
Form_Load (	This form is used to call the FormLevelTriggerredSwitchIntialize method in the TACSELEM.BAS module.

2.37.3 List Of Important Controls

Control Name	Description
txtU1 - txtU5	These text boxes are used to enter the input parameters U1 - U5 in the controller topology element.
txtD0 - txtD7	These text boxes are used to enter the coefficients of the denominator of the controller topology element.
txtN0 - txtN7	These text boxes are used to enter the coefficients of the numerator of the controller topology element.

2.38 The frmJMarti Form

2.38.1 Form Description

This form is user to show the Element Details screen for the JMARTI support program details.

2.38.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormLevelTriggeredSwitchInitialize method in the LVTRGSW1.BAS module.

2.38.3 List Of Important Controls

Control Name	Description
txtDiagonalizationFrequency	This is the text box used to enter the diagonalization frequency for JMARTI.
txtSteadyStateFrequency	This is the text box used to enter the steady state frequency for JMARTI.
txtLowestFrequency	This is the text box used to enter the lowest frequency for JMARTI.
txtNumberOfDecades	This is the text box used to enter the number of decades (of frequency sweep) required.
txtPointsPerDecade	This is a text box to indicate the points per decade.
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this level triggered switch element. This is done by calling the JMartiOK method.
cmdCancel	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.
cmdHelp	This Command Button is used to open the on-line help screen related to the JMARTI screen and it's fields.

2.39 The frmLevelTriggeredSwitch Form**2.39.1 Form Description**

This form is user to show the Element Details screen for the level triggered switch.

2.39.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormLevelTriggerredSwitchInitialize method in the LVTRGSWI.BAS module.

2.39.3 List Of Important Controls

Control Name	Description
lblLevelTrigerredSwitchDetails	This is a label control which contains the heading for the level triggered switch entries.
lblStatementType	This is the label for the cmbStatementType combo box.
cmbStatementType	This is a combo box control which is used to enter the statement type for this level triggered switch (Corresponds to the ITYPE variable in the level triggered switch card).
lblOperatingMode	This is the label for the txtOperatingMode combo box.
cmbOperatingMode	This is the combo box which is used to select the operating mode of the switch (C in the level triggered switch data card).
lblOutputParameterName	This is the label for the txtOutputParameter text box.
txtOutputParameterName	This is the text control which is used to enter the output parameter name for the level triggered switch (OUTPUT in the level triggered switch card).
lbiGain	This is the label for the txtGain text box.

Control Name	Description
txtGain	This is the text control which is used to enter the gain of the level triggered switch (A in the level triggered switch card).
lblFixedThreshold	This is the label for the txtFixedThreshold text box.
txtFixedThreshold	This is the text control which is used to enter the value of the fixed threshold of the level triggered switch (B in the level triggered switch card).
lblNamedThreshold	This is the label for the txtNamedThreshold text box.
txtNamedThreshold	This is the text control which is used to enter the value of the TACS parameter which will supply the NAMED THRESHOLD in the level triggered switch (C in the level triggered switch card).
lblDrivingSignal	This is the label control for the txtDrivingSignal text box.
txtDrivingSignal	This is the text control which is used to enter the name of the parameter which will supply the DRIVING SIGNAL parameter for the level triggered switch (D in the level triggered switch card).
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this level triggered switch element. This is done by calling the LevelTrigSwitchOK method.
cmdCancel	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.
cmdHelp	This Command Button is used to open the on-line help screen related to the Level triggered switch element and it's properties.
cmdInputParameters	This command button calls the frmInputParameters form which allows the user to enter the required input parameters (U1-U5).

2.40 The frmOpeningSwitch Form

2.40.1 Form Description

This is the Element Details screen for the Opening Switch element. It is used to save the variables which are specific to a Opening Switch element and are not entered in the General Switch Element Details screen.

2.40.2 List of Form Events

Event Name	Description
Form_Load ()	This method is used to call the FormOpeningSwitchInitialize method in the OPENSWL.BAS module as well as to disable the Message Bar in the frmATP form.

2.40.3 List Of Important Controls

Control Name	Description
lblOpeningSwitchDetails	This is a label control which is used as a heading for the screen entries.
lblAverageOpeningTime	This is the label for the txtAverageOpeningTime text box.
txtAverageOpeningTime	This is text box control which is used to enter the Average Opening Time of the switch. This entry corresponds to the TAV entry in the Opening Switch data card.
lblStandardDeviationJitterTime	This is the label for the txtAverageJitterTime text box.
txtStandardDeviationJitterTime	This is a text box control which is used to enter the Standard Deviation of the jitter in the switch opening time. This entry corresponds to the TSD entry in the Opening Switch data card.
lblCurrentTheshold	This is the label for the txtCurrentThreshold text box.

Control Name	Description
txtCurrentThreshold	This is a text box control which is used to enter the current threshold of the switch. This entry corresponds to the IE entry in the Opening Switch data card.
lblOutputType	This is the label for the cmbOutputType combo box.
cmbOutputType	This is a combo box control which is used to select the type of output required from this element. The entries in this combo box are loaded in the <code>FormOpeningSwitchInitialize</code> method in the <code>OPENSWI.BAS</code> module.
cmdOk	This is a command button control which is used to indicate that the element details description is complete. The <code>cmdOK_Click</code> event calls the <code>OpeningSwitchOK</code> method in the <code>OPENSWI.BAS</code> module.
cmdCancel	This is a command button control which is used to indicate that the user wishes to exit this screen without saving any of the changes.
cmdHelp	This is a command button control which is used to call the on-line help specific to the opening switch element.

2.41 The frmRampFunction Form

2.41.1 Form Description

This form is the screen for the Ramp Function - Element Details screen.

2.41.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the <code>FormRampFunctionInitialize</code> method in the <code>RAMPFUNC.BAS</code> module.

2.41.3 List Of Important Controls

Control Name	Description
lblRampFunctionDetails	This is the label for the ramp function details section.
lblNodeName	This is the label for the txtNodeName text box.
txtNodeName	This is the text box entry for the Node name of the node which is connected to the source (the other node is connected to earth automatically) (NODE in the ramp function data card).
lblStartTime	This is the label for the txtStartTime text box.
txtStartTime	This is the text box entry for the start time of the source (TSTART in the ramp function data card).
lblStopTime	This is the label for the txtStopTime text box.
txtStopTime	This is the text box entry for the stop time of the source (TSTOP in the ramp function data card).
lblAmplitude	This is the label for the txtAmplitude text box.
txtAmplitude	This is the text box entry for the Amplitude of the ramp function (A in the double exponential function data card).
lblFinalValueTime	This is the label for the txtFinalValueTime text box.
txtFinalValueTime	This is the text box entry for the time at which the final value is reached in the Ramp function (TIME-0 in the ramp function data card).
lblRampFunctionType	This is the label for the txtRampFunctionType combo box.
cmbRampFunctionType	This is the combo box for the type of source (current or voltage) (IV in the ramp function data card).
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this ramp function element. This is done by calling the RampFuncOK method.

Control Name	Description
cmdCancel	This Command Button is used to open the on-line help screen related to the ramp function element and its properties.
cmdHelp	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.

2.42 The RAMPFUNC.BAS Module

2.42.1 Module Description

This module provides functionality for the frmRampFunction Form. It contains a Type structure of the same name as the module with the members listed below (an array of variables of this type are then created with each instance of an element representing one member of this array).

2.42.2 List of Methods

Method Name	Description
FormRampFuncInitialize ()	This method is used to initialize the frmRampFunction form by adding the relevant default values of the different entries are also placed in the relevant text boxes or combo boxes in this method.
RampFunctionOK ()	This method is used to save all the different choices and values from the Ramp Function - Element Details screen to memory.
RampFuncOpenCircuit ()	This method is called by the OpenCircuit method in the ATP.BAS module. It opens the Ramp Function table in the ActiveFileName database file, and obtains the dual ramp functions details from this table.
RampFuncSaveCircuit (Position As String, i As Integer)	This method is called by the SaveCircuit method in the ATP.BAS module. It opens the Ramp Function table in the ActiveFileName database file, and writes the dual ramp function details to this table.

Method Name	Description
RampFuncWriteNodeNames	This method adds the node names of all the ramp functions in the circuit to the frmSetupSimulation's lstAvailableNodes list box.
RampFunctionWriteCircuit ()	This method is called from the WriteSourceCards method in the ATP.BAS module. It writes the required lines to the ActiveDataFileName.

2.42.3 List of Data Members and Constants

Member / Const Name	Description
Dim RampFuncNumber As Integer	This variable keeps track of the number of the ramp functions which is presently active.
Type RampFunc NodeName As String StartTime As String StopTime As String Amplitude As String FinalValueTime As String FunctionType As Integer End Type	This Type allows for the storage of all the details relevant to the ramp function including the node name, the start time, the stop time, the amplitude, the final value time, and the type of function required. To store these an array is required of this type.
Dim RampFncs (100) As RampFnc	This array of ramp functions allows the user to store the details relevant to this element for each dual ramp function in the circuit.

2.43 The frmReadOutput Form

Not available yet

2.44 The READOUTP.BAS Module

Not available yet.

2.45 The frmRelayOperatedSwitch Form

2.45.1 Form Description

This form is user to show the Element Details screen for the relay operated switch.

2.45.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormRelayOperatedSwitchInitialize method in the RELOPSWI BAS module.

2.45.3 List Of Important Controls

Control Name	Description
lblRelayOperatedSwitchDetails	This is a label control which contains the heading for the relay operated switch entries.
lblStatementType	This is the label for the cmbStatementType combo box.
cmbStatementType	This is a combo box control which is used to enter the statement type for this relay operated switch (Corresponds to the ITYPE variable in the relay operated switch card).
lblOperatingMode	This is the label for the txtOperatingMode combo box.
cmbOperatingMode	This is the combo box which is used to select the operating mode of the switch (C in the relay operated switch data card).
lblOutputParameterName	This is the label for the txtOutputParameter text box.
txtOutputParameterName	This is the text control which is used to enter the output parameter name for the relay operated switch (OUTPUT in the relay operated switch card).
lblGain	This is the label for the txtGain text box.

Control Name	Description
txtGain	This is the text control which is used to enter the gain of the relay operated switch (A in the relay operated switch card).
lblFixedThreshold	This is the label for the txtFixedThreshold text box.
txtFixedThreshold	This is the text control which is used to enter the value of the fixed threshold of the relay operated switch (B in the relay operated switch card).
lblNamedThreshold	This is the label for the txtNamedThreshold text box.
txtNamedThreshold	This is the text control which is used to enter the value of the TACS parameter which will supply the NAMED THRESHOLD in the relay operated switch (C in the relay operated switch card).
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this relay operated switch element. This is done by calling the RelayopswitchOK method.
cmdCancel	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.
cmdHelp	This Command Button is used to open the on-line help screen related to the Relay operated switch element and it's properties.
cmdInputParameters	This command button is used to call the frmInputParameters form, which allows the user to enter the inputs of the relay operated switch (ie. U1 - U5 etc.)

2.46 The frmRepetitivePulse Form

2.46.1 Form Description

This is the form for the Repetitive pulse Element Details screen.

2.46.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormRepetitivePulseInitialize method in the TACSSRCE.BAS module.

2.46.3 List Of Important Controls

Control Name	Description
lblRepetitivePulseDetails	This is the label for the Repetitive pulse details section.
lblOutputName	This is the label for the txtOutputName text box.
txtOutputName	This is the text box for the entry of the output name of the Repetitive pulse (OUTPUT in the Repetitive pulse data card).
lblPulseAmplitude	This is the label for the txtPulseAmplitude text box.
txtPulseAmplitude	This is the text box for the entry of the Amplitude of the Repetitive pulse (A in the Repetitive pulse data card).
lblPeriod	This is the label for the txtPeriod text box.
txtPeriod	This is the text box for the entry of the period of the Repetitive pulse (B in the Repetitive pulse data card).
lblPulseWidth	This is the label for the txtPulseWidth text box.
txtPulseWidth	This is the text box for the entry of the pulse width of the Repetitive pulse (C in the Repetitive pulse data card).
lblStartTime	This is the label for the txtStartTime text box.
txtStartTime	This is the text box for the entry of the start time of the Repetitive pulse (T-START in the Repetitive pulse data card).
lblStopTime	This is the label for the txtStopTime text box.

Control Name	Description
txtStopTime	This is the text box for the entry of the stop time of the Repetitive pulse (T-STOP in the Repetitive pulse data card).
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this Repetitive pulse element. This is done by calling the RepetitivePulseOK method.
cmdCancel	This Command Button is used to open the on-line help screen related to the Repetitive pulse element and it's properties.
cmdHelp	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.

2.47 The frmRepetitiveRamp Form

2.47.1 Form Description

This is the form for the Repetitive ramp Element Details screen.

2.47.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormRepetitiveRampIntialize method in the REPRamp.BAS module.

2.47.3 List Of Important Controls

Control Name	Description
lblRepetitiveRampDetails	This is the label for the Repetitive ramp details section.
lblOutputName	This is the label for the txtOutputName text box.

Control Name	Description
txtOutputName	This is the text box for the entry of the output name of the Repetitive ramp (OUTPUT in the Repetitive ramp data card).
lblPeakAmplitude	This is the label for the txtPeakAmplitude text box.
txtPeakAmplitude	This is the text box for the entry of the Amplitude of the Repetitive ramp (A in the Repetitive ramp data card).
lblPeriod	This is the label for the txtPeriod text box.
txtPeriod	This is the text box for the entry of the period of the Repetitive ramp (B in the Repetitive ramp data card).
lblStartTime	This is the label for the txtStartTime text box.
txtStartTime	This is the text box for the entry of the start time of the Repetitive ramp (T-START in the Repetitive ramp data card).
lblStopTime	This is the label for the txtStopTime text box.
txtStopTime	This is the text box for the entry of the stop time of the Repetitive ramp (T-STOP in the Repetitive ramp data card).
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this Repetitive ramp element. This is done by calling the RepetitiveRampOK method.
cmdCancel	This Command Button is used to open the on-line help screen related to the Repetitive ramp element and it's properties.
cmdHelp	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.

2.48 The frmSelectElement Form

Form Description

This form is used to display the Select Element screen.

2.48.1 List of Form Events

None

2.48.2 List Of Important Controls

Control Name	Description
lblElementLibraries	This is the label control used as a heading for the lstElementLibrary list box.
lstElementLibraries	This is a list box control which is used to list the element libraries available in ATF for Windows.
cmdOk	This command button is used to indicate that the user has selected the element required, and would like to place it in the circuit. This command button's Click () event calls the ShowLibraryForm in the SELECELE.BAS module.
cmdCancel	This command button is used to indicate that the user has decided not to select a library from this list and to exit without any changes.
cmdHelp	This command button is used to get on-line help on the Element libraries.

2.49 The frmRightImages Form

2.49.1 Form Description

This is the screen used to save pictures when the user scrolls to the right hand side of the circuit region. It has 182 (14 across and 13 down) images (in array format) known as the imgTempGrid

control array.

2.50 The SELECELE.BAS Module

2.50.1 Module Description

This module is used to provide functionality for the frmSelectElement form.

2.50.2 List of Methods

Method Name	Description
ShowElementLibrary ()	This method checks which library was selected from the Element Libraries List and selects the correct form for this library.

2.50.3 List of Data Members and Constants

None

2.51 The frmSetupSimulation Form

2.51.1 Form Description

This form is used to display the *RUN - Setup Simulation* option.

2.51.2 Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormSetupSimulationInitialize method in the SETUPSIM.BAS module.

2.51.3 List Of Important Controls

Control Name	Description
lblEntryList	This is the label control for the Entry List heading.
lblSimulationTimeStep	This is the label control for the txtSimulationTimeStep text box.
txtSimulationTimeStep	This text box is used to enter the simulation time step of the simulation (DELTAT in the floating-point miscellaneous card).
lblDurationOfSimulation	This is the label for the txtDurationOfSimulation text box.
txtDurationOfSimulation	This text box is used to enter the duration of the simulation (TMAX in the floating-point miscellaneous card).
lblIOUT	This is the label for the txtIOUT text box.
txtIOUT	This text box is used to enter the frequency of output relative to the simulation time step (IOUT in the integer miscellaneous card).
lblIPILOT	This is the label for the txtIPILOT text box.
txtIPILOT	This text box is used to enter the frequency of saving simulation values for later plotting (IPILOT in the integer miscellaneous card).
lblNENERG	This is the label for the txtNENERG text box.
txtNENERG	This text box is used to enter the NENERG variable (NENERG in the integer miscellaneous card).
lblFrequencyOfSimulation	This is the label for the txtFrequencyOfSimulation text box.
txtFrequencyOfSimulation	This text box is used to enter the frequency of the simulation.
lblBinaryFlags	This is a label control for the heading of the Binary Flags section of the Setup Simulation screen.
lblSteadyStateSolnType	This is the label for the cmbSteadyStateSolnType combo box.

Control Name	Description
cmbSteadyStateSolnType	This combo box is used to select the type of steady state solution printout required (KSSOUT in the integer-miscellaneous card)
lblNetworkConnectivity	This is the label for the optNetworkConnectivity option boxes.
optNetworkConnectivityYes	This option box is used to indicate that the circuit has network connectivity (IDOUBL in the integer miscellaneous card).
optNetworkConnectivityNo	This option box is used to indicate that the circuit has no network connectivity (IDOUBL in the integer miscellaneous card).
lblPrintoutExtrema	This is the label for the optPrintoutExtrema option boxes.
optPrintoutExtremaYes	This option box is used to indicate that the user would like a printout of extrema (MAXOUT in the integer miscellaneous card).
optPrintoutExtremaNo	This option box is used to indicate that the user would not like a printout of extrema (MAXOUT in the integer miscellaneous card).
lblManPrintoutFrequency	This is the label for the optManPrintoutFrequency option boxes.
optManPrintoutFrequencyYes	This option box is used to indicate that the user would like to add extra cards to manipulate the printout frequency (IPUN in the integer miscellaneous card).
optManPrintoutFrequencyNo	This option box is used to indicate that the user would not like to add extra cards to manipulate the printout frequency (IPUN in the integer miscellaneous card).
lblOpVoltAllNodes	This is the label for the optOpVoltAllNodes option boxes.
optOpVoltAllNodesYes	This option box is used to indicate that the user would like to printout the output voltage of all the nodes in the circuit.

Control Name	Description
optOpVoltAllNodesNo	This option box is used to indicate that the user would not like to printout the output voltage of all the nodes in the circuit.
lblOpVoltSpecNodes	This is a label control for labelling the lstOpVoltSpecNodes list box.
lstOpVoltSpecNodes	This list box contains a list of all the node names of the circuit elements in the present simulation. It's MultiSelect property is set to extended to allow for multiple selection. This list is used as the Output Variable Request card. The user is allowed to click on (select) as many of the nodes as he/she wishes.
cmdOk	This command button is used to indicate that the user has completed the entries of this screen. When selected, the Click () event calls the SetupSimOK method in the SETUPSIM.BAS module.
cmdOpen	This command button is used to open an existing simulation setup if one has been saved on a previous occasion. The Click () event calls the SetupSimLoadAllValues method in the SETUPSIM.BAS module.
cmdSave	This command button is used to save an existing simulation setup to file. The Click () event calls the SetupSimFileAllValues method in the SETUPSIM.BAS module.
cmdCancel	This command button is used to exit the screen without saving any of the changes made. Since none of the changes are saved until either the <i>SAVE</i> or <i>OK</i> buttons have been selected, this simply exits from this form by unloading it.
cmdHelp	This command button is used to indicate that the user would like on-line help on the Setup Simulation screen.

2.52 The SETUPSIM.BAS Module

2.52.1 Module Description

This module is used to provide functionality for the frmSetupSimulation form.

2.52.2 List of Methods

Method Name	Description
FormSetupSimulationInitialize ()	This method is used to setup all the default values in the relevant controls. The default values are stored in a Default type which contains one variable of every type on the Setup Simulation screen.
ErrorMessage (errortype As Integer)	This method is used to display an error message of the required type. Each control on the form is checked in the SetupSimOK method, and if there is an error, this method is called to display the correct error message.
SetupSimFileAllValues ()	This method is used to save all of the entries in the Setup Simulation screen to file. This is done using the standard file IO commands and printing the variables to the file. The Common Dialog control is used to provide the functionality for the Save screen.
SetupSimOpenAllValues ()	This method is used to open an existing file of all of the entries in the Setup Simulation screen to the screen. This is done using the standard file IO commands and printing the variables to the screen. The Common Dialog control is used to provide the functionality for the Open screen.
SetupSimOK ()	This method is used to save all of the entries in the Setup Simulation screen to memory. Each variable has a corresponding data member which is stored in memory and then used to write the relevant data cards during the Run Simulation command.

2.52.3 List of Data Members and Constants

Member / Const Name	Description
Dim SimulationTimeStep As String	This is the variable which is used to store the simulation time step.
Dim SimDuration As String	This is the variable which is used to store the duration of the simulation.
Dim IOUT As String	This is the variable which is used to store the IOUT entry.
Dim IPLOT As String	This is the variable which is used to store the IPLOT entry.
Dim NENERG As String	This is the variable which is used to store the NENERG entry.
Dim SteadyStateSolnType As Integer	This is the variable which is used to store the steady state solution type flag.
Dim NetConnectivity As Integer	This is the variable which is used to store the network connectivity flag.
Dim PrintoutExtrema As Integer	This is the variable which is used to store the printout extrema flag.
Dim ManPrintoutFreq As Integer	This is the variable which is used to store the manipulate printout frequency flag.
Dim OpVoltAllNodes As Integer	This is the variable which is used to store the output voltage of all nodes flag.
Dim OpVoltSpecNodes() As String	This array of strings is used to store the names of all the nodes for which the voltage is required in the ATP solution file.
Dim NumberOfNodes As Integer	This variable is needed to maintain the number of nodes in the present circuit.
Dim SetupSimulationSaved As Integer	This is a boolean variable which is set to True if the Setup Simulation screen has been saved.
Dim ErrorFound As Integer	This is a boolean variable which is set to True whenever an error is found in the entries of the Setup Simulation screen.

2.53 The frmSimpleDifferentiator Form

2.53.1 Form Description

This is the Element Details form for the Simple differentiator element.

2.53.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormSimpleDifferentiatorInitialize method in the TACSELEM.BAS module.

2.53.3 List Of Important Controls

Control Name	Description
lblSimpleDifferentiatorDetails	This is a label control which contains the heading for the simple differentiator entries.
lblStatementType	This is the label for the cmbStatementType combo box.
cmbStatementType	This is a combo box control which is used to enter the statement type for this simple differentiator (Corresponds to the ITYPE variable in the simple differentiator card).
lblOutputParameterName	This is the label for the txtOutputParameter text box.
txtOutputParameterName	This is the text control which is used to enter the output parameter name for the simple differentiator (OUTPUT in the simple differentiator card).
lblGain	This is the label for the txtGain text box.
txtGain	This is the text control which is used to enter the gain of the simple differentiator (A in the simple differentiator card).

Control Name	Description
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this simple differentiator element. This is done by calling the SimpleDifferentiatorOK method.
cmdCancel	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.
cmdHelp	This Command Button is used to open the on-line help screen related to the Simple differentiator element and it's properties.
VscrollInputParameter	This is the vertical scroll bar on the right of the Input Parameter frame. It is used to scroll between input parameter entries. The VscrollInputParameter_Change method saves the names of the input parameters on the screen presently into new variables of the InputParameters() array, and changes the Caption property of each of the lbInputParameter labels to indicate the next input parameter.

2.54 The frmSimpleSources Form

2.54.1 Form Description

This form is used to display the Simple sources Library screen.

2.54.2 List of Form Events

None

2.54.3 List Of Important Controls

Control Name	Description
lbLibraryElements	This is the label control used as a heading for the lstLibraryElements list box.

Control Name	Description
lstLibraryElements	This is a list box control which is used to list the simple sources available in ATP for Windows
cmdOk	This command button is used to indicate that the user has selected the element required, and would like to place it in the circuit. This command button's Click () event calls the SourceShowElementIcon method in the SIMPLLIB.BAS module.
cmdCancel	This command button is used to indicate that the user has decided not to select an element from this library, and to exit without any changes.
cmdHelp	This command button is used to get on-line help on the Simple sources Library.

2.55 The SIMPLLIB.BAS Module

2.55.1 Module Description

This module is used to provide functionality for the frmSimpleSources form.

2.55.2 List of Methods

Method Name	Description
SourceShowElementIcon ()	This method checks which element was selected from the Element List and selects the correct icon for this element.

2.55.3 List of Data Members and Constants

None

2.56 The frmSimpleRLC Form

2.56.1 Form Description

This form is used to show the Simple RLC branch screen.

2.56.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormSimpleRLCInitialize method in the SIMPLRLC.BAS module.

2.56.3 List Of Important Controls

Control Name	Description
lblSimpleRLCDetails	This is the label control used for the heading of the Simple RLC branch details.
lblLeftNodeName	This is the label for the txtLeftNodeName text box.
txtLeftNodeName	This is the text box for the entry of the left node name of the RLC Branch (Node1 in the simple RLC card)
lblRightNodeName	This is the label for the txtRightNodeName text box.
txtRightNodeName	This is the text box for the entry of the right node name of the simple RLC Branch (Node2 in the Simple RLC card).
lblNode3Name	This is the label for the txtNode3Name text box.
txtNode3Name	This is the text box for the entry of the left node name of a similar RLC Branch (Node3 in the simple RLC card).
lblNode4Name	This is the label for the txtNode4Name text box.
txtNode4Name	This is the text box for the entry of the right node name of a similar simple RLC Branch (Node4 in the Simple RLC card).
lblResistance	This is the label for the txtResistance text box.
txtResistance	This is the text box for the entry of the resistance of the simple RLC Branch (R in the Simple RLC card).

Control Name	Description
lblCapacitance	This is the label for the txtCapacitance text box.
txtCapacitance	This is the text box for the entry of the capacitance of the simple RLC Branch (C in the Simple RLC card).
lblInductance	This is the label for the txtInductance text box.
txtInductance	This is the text box for the entry of the inductance of the simple RLC Branch (L in the Simple RLC card).
lblCapacitanceInput	This is the label for the cmbCapacitanceInput combo box.
cmbCapacitanceInput	This is the combo box for the entry of the capacitance input units (COPT on the floating point miscellaneous card).
lblInductanceInput	This is the label for the cmbInductanceInput combo box (XOPT ON THE floating-point miscellaneous card).
cmbInductanceInput	This is the combo box for the entry of the inductance input units (XOPT on the Frequency Miscellaneous card).
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this simple RLC element. This is done by calling the SimpleRLCOK method.
cmdCancel	This Command Button is used to open the on-line help screen related to the simple RLC element and it's properties.
cmdHelp	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.

2.57 The SIMPLRLC.BAS Module

2.57.1 Module Description

This module is used to provide functionality for the frmSimpleRLC form. It contains a Type

structure of the same name as the module with the members listed below (an array of variables of this type are then created with each instance of an element representing one member of this array).

2.57.2 List of Methods

Method Name	Description
FormSimpleRLCInitialize ()	This method is used to initialize the frmSimpleRLC by adding the relevant default values of the different entries are also placed in the relevant text boxes or combo boxes in this method.
SimpleRLCOK ()	This method is used to save all the different choices and values from the Simple RLC - Element Details screen to memory.
SimpleRLCOpenCircuit ()	This method is called by the OpenCircuit method in the ATP.BAS module. It opens the Simple RLC element table in the ActiveFileName database file, and obtains the distributed RLC elements and details from this table.
SimpleRLCSaveCircuit (Position As String, i As Integer)	This method is called by the SaveCircuit method in the ATP.BAS module. It opens the Simple RLC element table in the ActiveFileName database file, and writes the distributed RLC elements and details to this table.
SimpleRLCWriteNodeNames (Number As Integer)	This method adds the node names of all the simple RLC elements in the circuit to the frmSetupSimulation's lstAvailableNodes list box.
SimpleRLCWriteCircuit ()	This method is called from the WriteBranchCards method in the ATP.BAS module. It writes the required lines to the ActiveDataFileName.

2.57.3 List of Data Members and Constants

Member / Const Name	Description
Dim SimpleRLCNumber As	

Member / Const Name	Description
Integer Type SimpleRLCElement LeftNodeName As String RightNodeName As String Node3Name As String Node4Name As String Resistance As String Inductance As String Capacitance As String OutputType As Integer End Type	This type is the defining type of the simple RLC element. The array of RLC elements (see next entry) contains every field in this type which includes the left and right node names of the branch, the names of Node 3 and Node 4 (similar RLC branch node names), the resistance, capacitance, and inductance, and the type of output which is required from this branch
Dim RLCs(100) As SimpleRLCElement	This array allows for the saving of every field of each simple RLC branch which the user creates since it is an array of simple RLC elements.

2.58 The frmSinFunction Form

2.58.1 Form Description

This form is the screen for the Sin function - Element Details screen.

2.58.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormSinFunctionInitialize method in the SinFUNC.BAS module.

2.58.3 List Of Important Controls

Control Name	Description
lblSinFunctionDetails	This is the label for the sin function details section.
lblNodeName	This is the label for the txtNodeName text box.
txtNodeName	This is the text box entry for the Node name of the node which is connected to the source (the other node is connected to earth automatically) (NODE in the sin function data card).
lblStartTime	This is the label for the txtStartTime text box.
txtStartTime	This is the text box entry for the start time of the source (TSTART in the sin function data card).
lblStopTime	This is the label for the txtStopTime text box.
txtStopTime	This is the text box entry for the stop time of the source (TSTOP in the sin function data card).
lblAmplitude	This is the label for the txtAmplitude text box.
txtAmplitude	This is the text box entry for the Amplitude of the sin function (A in the double exponential function data card).
lblSinusoidalFrequency	This is the label for the txtSinusoidalFrequency text box.
txtSinusoidalFrequency	This is the text box entry for the frequency of the Sin function (FREQ in the sin function data card).
lblInitialShiftType	This is the label for the cmbInitialShiftType combo box.
cmbInitialShiftType	This is the combo box which is used to select the initial shift type (A1 in the sin function data card).
lblInitialCondition	This is the label for the txtInitialCondition text box.
txtInitialCondition	This is the text box entry for the initial condition (time shift or phase shift depending on the shift type selected of the Sin function (TIME-0 in the sin function data card).

Control Name	Description
lblSinFunctionType	This is the label for the txtSinFunctionType combo box.
cmbSinFunctionType	This is the combo box for the type of source (current or voltage) (IV in the sin function data card).
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this sin function element. This is done by calling the SinFunctOK method.
cmdCancel	This Command Button is used to open the on-line help screen related to the sin function element and it's properties.
cmdHelp	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.

2.59 The SINFUNC.BAS Module

2.59.1 Module Description

This module provides functionality for the frmSinFunction Form. It contains a Type structure of the same name as the module with the members listed below (an array of variables of this type are then created with each instance of an element representing one member of this array).

2.59.2 List of Methods

Method Name	Description
FormSinFunctInitialize ()	This method is used to initialize the frmSinFunction form by adding the relevant default values of the different entries are also placed in the relevant text boxes or combo boxes in this method.
SinFunctionOK ()	This method is used to save all the different choices and values from the Sin function - Element Details screen to memory.

Method Name	Description
SinFuncOpenCircuit ()	This method is called by the OpenCircuit method in the ATP.BAS module. It opens the Sin Function table in the ActiveFileName database file, and obtains the dual ramp functions details from this table.
SinFuncSaveCircuit (Position As String, i As Integer)	This method is called by the SaveCircuit method in the ATP.BAS module. It opens the Sin Function table in the ActiveFileName database file, and saves the dual ramp function details to the file.
SinFuncWriteNodeNames	This method adds the node names of all the Sin functions in the circuit to the frmSetupSimulation's lstAvailableNodes list box.
SinFunctionWriteCircuit ()	This method is called by the RunRunSimulation method in order to write the relevant sin function details to card. This method calls the relevant methods in the Sin_Funct class in the Data Case Module.

2.59.3 List of Data Members and Constants

Member / Const Name	Description
Dim SinFncNumber As Integer	This variable is used to keep track of the currently active sin function.
Type SinFunc NodeName As String StartTime As String StopTime As String Amplitude As String SinusoidalFreq As String InitialShiftType As Integer InitialShift As String FunctionType As Integer End Type	This type is the defining type of the sin function element. It includes variables to store all the properties of a sin function including the node name, the start time, the stop time, the amplitude, the frequency, initial shift type, initial shift, and function type. To access these variables, an array of variables of this type must be created (see below).
Dim SinFncs(100) As SinFunc	This is the array of sin functions which is used to store the details relating to a sin function (see above).

2.60 The frmSpecDevInputParameters Form

2.60.1 Form Description

This form is user to enter the input parameters of the relay operated switch, the level triggered switch, the simple differentiator, and the controlled integrator. The entries are U1-U5

2.60.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormLevelTriggerredSwitchInitialize method in the TACSELEM.BAS module.

2.60.3 List Of Important Controls

Control Name	Description
txtU1 - txtU5	These text boxes are used to enter the input parameters U1 - U5 in the controller topology element.

2.61 The frmSimusoidalWaveformGenerator Form

2.61.1 Form Description

This form is the screen for the Sin waveform generator - Element Details screen.

2.61.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormSinWaveGenInitialize method in the TACSSRCE.BAS module.

2.61.3 List Of Important Controls

Control Name	Description
lblSinWaveGenDetails	This is the label for the sin waveform generator details section.
lblOutputName	This is the label for the txtOutputName text box.
txtOutputName	This is the text box entry for the output name of the source (OUTPUT in the sin waveform generator data card).
lblStartTime	This is the label for the txtStartTime text box.
txtStartTime	This is the text box entry for the start time of the source (TSTART in the sin waveform generator data card).
lblStopTime	This is the label for the txtStopTime text box.
txtStopTime	This is the text box entry for the stop time of the source (TSTOP in the sin waveform generator data card).
lblAmplitude	This is the label for the txtAmplitude text box.
txtAmplitude	This is the text box entry for the Amplitude of the sin waveform generator (A in the double exponential function data card).
lblSinusoidalFrequency	This is the label for the txtSinusoidalFrequency text box.
txtSinusoidalFrequency	This is the text box entry for the frequency of the Sin waveform generator (FREQ in the sin waveform generator data card).
lblInitialPhaseAngle	This is the label for the txtInitialPhaseAngle text box.
txtInitialPhaseAngle	This is the text box entry for the initial phase angle of the Sin waveform generator (C in the sin waveform generator data card).
lblSinWaveGenType	This is the label for the txtSinWaveGenType combo box.

Control Name	Description
ctrlSinWaveGenType	This is the combo box for the type of source (current or voltage) (IV in the sin waveform generator data card).
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this sin waveform generator element. This is done by calling the SinWaveFormOK method.
cmdCancel	This Command Button is used to open the on-line help screen related to the sin waveform generator element and it's properties.
cmdHelp	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.

2.62 The frmSlaveClosingSwitch Form

2.62.1 Form Description

This is the Element Details screen for the Slave closing switch element. It is used to save the variables which are specific to a slave closing switch element and are not entered in the General Switch Element Details screen.

2.62.2 List of Form Events

Event Name	Description
Form_Load ()	This method is used to call the FormSlaveClosingSwitchInitialize method in the SWELEMEN.BAS module as well as to disable the Message Bar in the frmATP form.

2.62.3 List Of Important Controls

Control Name	Description
lblSlaveClosingSwitchDetails	This is a label control which is used as a heading for the screen entries.
lblAverageDelayTime	This is the label for the txtAverageDelayTime text box.
txtAverageDelayTime	This is text box control which is used to enter the average delay in closing of this switch with respect to the master switch. This entry corresponds to the TDELAY(AV) entry in the slave closing switch data card.
lblStandardDeviationDelay	This is the label for the txtStandardDeviationDelay text box.
txtStandardDeviationDelay	This is a text box control which is used to enter the Standard Deviation of the delay in the switch closing time. This entry corresponds to the TDELAY(SD) entry in the slave closing switch data card.
lblLeftMasterNodeName	This is the label for the txtLeftMasterNodeName text box.
txtLeftMasterNodeName	This text box control is used to enter the name of the left node of the master switch (Node5 in the slave closing switch card).
lblRightMasterNodeName	This is the label for the txtRightMasterNodeName text box.
txtRightMasterNodeName	This text box control is used to enter the name of the left node of the master switch (Node6 in the slave closing switch card).
lblOutputType	This is the label for the cmbOutputType combo box.
cmbOutputType	This is a combo box control which is used to select the type of output required from this element. The entries in this combo box are loaded in the FormSlaveClosingSwitchInitialize method in the SLCLSSWI.BAS module.

Control Name	Description
cmdOk	This is a command button control which is used to indicate that the element details description is complete. The cmdOK_Click event calls the SlaveClosingSwitchOK method in the SLCLSSWIBAS module.
cmdCancel	This is a command button control which is used to indicate that the user wishes to exit this screen without saving any of the changes.
cmdHelp	This is a command button control which is used to call the on-line help specific to the slave closing switch element.

2.63 The frmSlaveOpeningSwitch Form

2.63.1 Form Description

This is the Element Details screen for the Slave opening switch element. It is used to save the variables which are specific to a slave opening switch element and are not entered in the General Switch Element Details screen.

2.63.2 List of Form Events

Event Name	Description
Form_Load ()	This method is used to call the FormSlaveOpeningSwitchInitialize method in the SWELEMEN.BAS module as well as to disable the Message Bar in the frmATP form.

2.63.3 List Of Important Controls

Control Name	Description
lblSlaveOpeningSwitchDetails	This is a label control which is used as a heading for the screen entries.

Control Name	Description
lblAverageDelayTime	This is the label for the txtAverageDelayTime text box.
txtAverageDelayTime	This is text box control which is used to enter the average delay in opening of this switch with respect to the master switch. This entry corresponds to the TDELAY(AV) entry in the slave opening switch data card.
lblStandardDeviationDelay	This is the label for the txtStandardDeviationDelay text box.
txtStandardDeviationDelay	This is a text box control which is used to enter the Standard Deviation of the delay in the switch opening time. This entry corresponds to the TDELAY(SD) entry in the slave opening switch data card.
lblLeftMasterNodeName	This is the label for the txtLeftMasterNodeName text box.
txtLeftMasterNodeName	This text box control is used to enter the name of the left node of the master switch (Node5 in the slave opening switch card).
lblRightMasterNodeName	This is the label for the txtRightMasterNodeName text box.
txtRightMasterNodeName	This text box control is used to enter the name of the left node of the master switch (Node6 in the slave opening switch card).
lblCurrentThreshold	This is the label control for the txtCurrentThreshold text box.
txtCurrentThreshold	This is the text box control used to enter the current threshold corresponding to the switch actually opening (IE in the slave opening switch card).
lblOutputType	This is the label for the cmbOutputType combo box.

Control Name	Description
cmdOk	This is a command button control which is used to indicate that the element details description is complete. The cmdOK Click event calls the SlaveOpeningSwitchOK method in the SWELEMEN.BAS module.
cmdCancel	This is a command button control which is used to indicate that the user wishes to exit this screen without saving any of the changes.
cmdHelp	This is a command button control which is used to call the on-line help specific to the slave opening switch element.

2.64 The frmSpecialDevice Form

2.64.1 Form Description

This form is used to display the Special Device - Element Details screen.

2.64.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormSpecialDeviceInitialize method in the SPECDEV.BAS module.

2.64.3 List Of Important Controls

Control Name	Description
lblSpecialDevices	This is the label control for the Special Devices: heading.
lblDeviceType	This is the label control used as a label for the cmbDeviceType combo box.
cmbDeviceType	This is a combo box which is used to select the type of special device required.

Control Name	Description
cmdOk	This Command Button is used to indicate that the user has selected the specific special device required. The Click () event of this command button calls the SelectSpecialDevice method in the SPECDEV.BAS module.
cmdCancel	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.
cmdHelp	This Command Button is used to open the on-line help screen related to the special devices available.

2.65 The frmSurgeArrestor Form

2.65.1 Form Description

This form is used to show the Piecewise-linear modelled surge arrestor Element Details screen.

2.65.2 List of Events

Event Name	Description
Form_Load ()	This form is used to call the FormSgArrInitialize method in the SRGARR.BAS module.

2.65.3 List Of Important Controls

Control Name	Description
lblArrestorDetails	This label control is used to write the heading of the Arrestor Details section.
lblLeftNodeName	This is the label for the txtLeftNodeName text box.

Control Name	Description
txtLeftNodeName	This is the text box for the entry of the left node name of the Piecewise-linear modelled surge arrester (Node1 in the piecewise-linear modelled surge arrester card).
lblRightNodeName	This is the label for the txtRightNodeName text box.
txtRightNodeName	This is the text box for the entry of the right node name of the piecewise-linear modelled surge arrester (Node2 in the Piecewise-linear modelled surge arrester card).
lblNode3Name	This is the label for the txtNode3Name text box.
txtNode3Name	This is the text box for the entry of the left node name of a similar Piecewise-linear modelled surge arrester (Node3 in the piecewise-linear modelled surge arrester card).
lblNode4Name	This is the label for the txtNode4Name text box.
txtNode4Name	This is the text box for the entry of the right node name of a similar piecewise-linear modelled surge arrester (Node4 in the Piecewise-linear modelled surge arrester card).
lblFlashoverVoltage	This is the label for the txtFlashoverVoltage text box.
txtFlashoverVoltage	This is the text box for the entry of the flashover voltage of the piecewise-linear modelled surge arrester (VFLASH in the Piecewise-linear modelled surge arrester card).
lblExtinguishingTime	This is the label for the txtExtinguishingTime text box.
txtExtinguishingTime	This is the text box for the entry of the earliest time after flashover that the arc can extinguish (TDLY in the Piecewise-linear modelled surge arrester card).
lblThresholdVoltage	This is the label for the txtThresholdVoltage text box.
txtThresholdVoltage	This is the text box for the entry of the threshold voltage of the arrester (VSEAL in the Piecewise-linear modelled surge arrester card).

Control Name	Description
lblConductionSegment	This is the label for the txtConductionSegment text box.
txtConductionSegment	This is the text box for the entry of the conduction segment of the piecewise-linear characteristic of the arrester (JUMP in the Piecewise-linear modelled surge arrester card).
lblNoOfSegments	This is the label for the txtNoOfSegments text box.
txtNoOfSegments	This is the text box for the entry of the number of segments of the piecewise-linear characteristic.
lblOutputType	This is the label for the cmbOutputType combo box.
cmbOutputType	This is the combo box for the entry of the output type required from this branch of the simulation (P in the Piecewise-linear modelled surge arrester card).
cmdCurrVoltChar	This command button is used to call the frmCurrentVoltageCharacteristics form which allows the user to enter the piecewise-linear characteristic of the surge arrester.
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this piecewise-linear modelled surge arrester element. This is done by calling the ExponSgArrOK method.
cmdCancel	This Command Button is used to open the on-line help screen related to the piecewise-linear modelled surge arrester element and its properties.
cmdHelp	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.

2.66 The SRGARR.BAS Module

2.66.1 Module Description

This module is used to provide functionality for the surge arrestor and the exponentially modelled surge arrestor elements and forms. It contains a Type structure of the same name as the module with the members listed below (an array of variables of this type are then created with each instance of an element representing one member of this array).

2.66.2 List of Methods

Method Name	Description
ArrdatOK ()	This method is used to save the frmArrdat values to memory when the user clicks on the OK command button in that form.
CurrentVoltageSaveValue (EntryCount As Integer)	This method is used by the frmCurrentVoltageCharacteristics form to save each entry of the current - voltage characteristic. It is called by the cmdNext command button's click event.
ExponSurgeArrestorOK ()	This method is called by the frmExponSurgeArrestor to save the details from that form to memory.
Function FindIfSurgeExists () As Integer	This function is used by the WriteLineConstants method to determine if a particular surge arrestor is an exponential arrestor, or a piecewise-linear surge arrestor.
FormArrdatInitialize ()	This method is used to initialize the frmArrdat form, including placing values which the user has already completed in the correct controls.
FormCurrentVoltageInitialize ()	This method is used to initialize the frmCurrentVoltage form, including placing values which the user has already completed in the correct controls.
FormExponSurgeArrestorInitialize ()	This method is used to initialize the frmExponSurgeArrestor form, including placing values which the user has already completed in the correct controls.

Method Name	Description
FormSurgeArrestorInitialize ()	This method is used to initialize the frmSurgeArrestor form, including placing values which the user has already completed in the correct controls.
Function GetCurrent (i As Integer) As String	This function is used by the frmCurrentVoltageCharacteristics form to obtain the next current entry when the cmdNext command button is pressed.
Function GetVoltage (i As Integer) As String	This function is used by the frmCurrentVoltageCharacteristics form to obtain the next voltage entry when the cmdNext command button is pressed.
SurgeArrestorOK ()	This method is called by the frmSurgeArrestor to save the details from that form to memory.
SurgeArrestorOpenCircuit ()	This method is used to open the surge arrestor table, and write the relevant surge arrestor element to this table. The method is called by the OpenCircuit method in the ATP.BAS module.
SurgeArrestorSaveCircuit (Position As String, i As Integer)	This method is used to open the surge arrestor table, and obtain the relevant surge arrestor elements from this table. The method is called by the SaveCircuit method in the ATP.BAS module.
SurgeArrestorWriteArrdatStart (Position As String, i As Integer)	This method is used to write the ARRDAT details to the ZnoFitter / ARRDAT ATP support program. It is called by the WriteArrdat method in the ATP.BAS file.
SurgeArrestorWriteCircuit (Position As String, i As Integer)	This method is used to write the details of the surge arrestor to the active data file name. The method is called from the WriteBranchCards method in the ATP.BAS module.
SurgeArrestorWriteNodeNames (Number As Integer)	This method is used to write the node names of all the surge arrestor's in the circuit to the frmSetupSimulation's lstAvailableNodes list box.

2.66.3 List of Data Members and Constants

Member / Const Name	Description
Dim SurgeArrestorNumber As Integer	This variable is used to keep track of the particular surge arrestor which is currently active.
Dim ArrestorTypeArr As Integer	This variable is used to distinguish between the piecewise-linear and exponential surge arrestors.

Member / Const Name	Description
Type SurgeArrestorType LeftNodeName As String RightNodeName As String Node3Name As String Node4Name As String ExtinguishingTime As String ConductionSegment As String ThresholdVoltage As String OutputType As Integer NumberOfSegments As Integer NumberOfPhases As Integer MaximumRelativeError As String ReferenceVoltage As String FlashoverVoltage As String ReferenceRMS As String DesiredRMS As String VoltageScale As String CurrentScale As String CurrentMinimum As String ArrestorType As Integer Current(50) As String Voltage(50) As String NumberOfSteps As Integer WrittenToFile As Integer ArrdatRequired As Integer NumberOfSimilarArrestors As Integer LeftNodeNames(50) As String RightNodeNames(50) As String ZnOFitterFileName As String ActiveDataSupportName As String End Type	This is the defining type of the surge arrestor elements. It contains fields for every entry in both the regular surge arrestor (piecewise-linear) and the exponentially modelled surge arrestor. Using an array of surge arrestors, each field can be accessed by each element (see next entry).
Dim SurgeArrestors(100) As SurgeArrestorType	This is an array of surge arrestors each element of which includes many of the fields in the surge arrestor type (above).

Member / Const Name	Description
Dim NewElement As Integer	This variable is used to indicate whether or not an element is a new element (ie. has it been defined previously).
Global NewSurgeElement	This variable is also used to indicate whether an element is a new element or an existing element. It is required in the frmExponSurgeArrestor and the frmSurgeArrestor.

2.67 The frmStepFunction Form

2.67.1 Form Description

This form is the screen for the Step function - Element Details screen.

2.67.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormStepFunctionIntialize method in the STEPFUNC.BAS module.

2.67.3 List Of Important Controls

Control Name	Description
lblStepFunctionDetails	This is the label for the step function details section.
lblNodeName	This is the label for the txtNodeName text box.
txtNodeName	This is the text box entry for the Node name of the node which is connected to the source (the other node is connected to earth automatically) (NODE in the step function data card).
lblStartTime	This is the label for the txtStartTime text box.

Control Name	Description
txtStartTime	This is the text box entry for the start time of the source (TSTART in the step function data card).
lblStopTime	This is the label for the txtStopTime text box.
txtStopTime	This is the text box entry for the stop time of the source (TSTOP in the step function data card).
lblAmplitude	This is the label for the txtAmplitude text box.
txtAmplitude	This is the text box entry for the Amplitude of the step function (A in the double exponential function data card).
lblStepFunctionType	This is the label control used to label the cmbStepFunctionType combo box.
cmbStepFunctionType	This combo box is used to select the type of step function required (voltage or current) (IV in the step function card).
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this step function element. This is done by calling the StepFuncOK method.
cmdCancel	This Command Button is used to open the on-line help screen related to the step function element and it's properties.
cmdHelp	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.

2.68 The STEPFUNC.BAS Module

2.68.1 Module Description

This module provides functionality for the frmStepFunction Form. It contains a Type structure of the same name as the module with the members listed below (an array of variables of this type are then created with each instance of an element representing one member of this array).

2.68.2 List of Methods

Method Name	Description
FormStepFuncInitialize ()	This method is used to initialize the frmStepFunction form by adding the relevant default values of the different entries are also placed in the relevant text boxes or combo boxes in this method.
StepFunctionOK ()	This method is used to save all the different choices and values from the Step function - Element Details screen to memory.
StepFuncOpenCircuit ()	This method is called by the OpenCircuit method in the ATP.BAS module. It opens the step function table in the ActiveFileName database file, and obtains the dual ramp functions details from this table.
StepFuncSaveCircuit (Position As String, i As Integer)	This method is called by the SaveCircuit method in the ATP.BAS module. It opens the step function table in the ActiveFileName database file, and writes the dual ramp function details to this table.
StepFuncWriteNodeNames	This method adds the node names of all the step functions in the circuit to the frmSetupSimulation's lstAvaialableNodes list box.
StepFunctionWriteCircuit ()	This method is called from the WriteSourceCards method in the ATP.BAS module. It writes the requiried lines to the ActiveDataFileName.

2.68.3 List of Data Members and Constants

Member / Const Name	Description
Dim StepFuncNumber As Integer	This variable keeps track of the number of the step functions which is presently active.
Type StepFunc NodeName As String StartTime As String StopTime As String Amplitude As String FunctionType As Integer End Type	This Type allows for the storage of all the details relevant to the step function including the node name, the start time, the stop time, the amplitude, and the type of function required. To store these an array is required of this type.
Dim StepFuncs(100) As StepFunc	This array of step functions allows the user to store the details relevant to this element for each dual ramp function in the circuit.

2.69 The frmStandardSwitch Form

2.69.1 Form Description

This is the Element Details screen for the Standard Switch element. It is used to save the variables which are specific to a Standard Switch element and are not entered in the General Switch Element Details screen.

2.69.2 List of Form Events

Event Name	Description
Form_Load ()	This method is used to call the FormStandardSwitchInitialize method in the SWELEMEN.BAS module as well as to disable the Message Bar in the frmATP form.

2.69.3 List Of Important Controls

Control Name	Description
lblStandardSwitchDetails	This is a label control which is used as a heading for the screen entries.
lblCloseTime	This is the label for the txtCloseTime text box.
txtCloseTime	This is text box control which is used to enter the Standard Time of the switch (TCLOSE entry in the Standard Switch data card).
lblOpenTime	This is the label for the txtOpenTime text box.
txtOpenTime	This is a text box control which is used to enter the switch operating time (TOPEN entry in the Standard Switch data card).
lblCurrentThreshold	This is the label for the txtCurrentThreshold text box.
txtCurrentThreshold	This is a text box which is used to enter the current threshold of the switch (IE in the standard switch data card).
lblOutputType	This is the label for the cmbOutputType combo box.
cmbOutputType	This is a combo box control which is used to select the type of output required from this element. The entries in this combo box are loaded in the FormStandardSwitchInitialize method in the STNDSWI.BAS module.
cmdOk	This is a command button control which is used to indicate that the element details description is complete. The cmdOK_Click event calls the StandardSwitchOK method in the STNDSWI.BAS module.
cmdCancel	This is a command button control which is used to indicate that the user wishes to exit this screen without saving any of the changes.
cmdHelp	This is a command button control which is used to call the on-line help specific to the standard switch element.

2.70 The SWELEMEN.BAS Module

2.70.1 Module Description

This is the module which contains methods needed by the all of the switch elements in the ATP for Windows. It contains a Type structure of the same name as the module with the members listed below (an array of variables of this type are then created with each instance of an element representing one member of this array).

2.70.2 List of Methods

Method Name	Description
ClosingSwitchOK ()	This method is called by the frmClosingSwitch cmdOK_click event when all of the details of the closing switch have been entered. It saves the details to memory.
ConstDelaySlaveSwitchOK ()	This method is called by the frmClosingSwitch cmdOK_click event when all of the details of the switch have been entered. It saves the details to memory.
FormClosingSwitchInitialize ()	This is method checks if the closing switch element is a new element. If it is not a new element, the text box for this element is filled in with whatever was in the element before. Also, if it is a new element, the black box count element is incremented.
FormConstDelaySlaveSwitchInitialize ()	This is method checks if the const delay slave switch element is a new element. If it is not a new element, the text box for this element is filled in with whatever was in the element before. Also, if it is a new element, the black box count element is incremented.
FormOpeningSwitchInitialize ()	This is method checks if the opening switch element is a new element. If it is not a new element, the text box for this element is filled in with whatever was in the element before. Also, if it is a new element, the black box count element is incremented.

Method Name	Description
FormSlaveClosingSwitchInitialize ()	This is method checks if the slave closing switch element is a new element. If it is not a new element, the text box for this element is filled in with whatever was in the element before. Also, if it is a new element, the black box count element is incremented.
FormSlaveOpeningSwitchInitialize ()	This is method checks if the slave opening switch element is a new element. If it is not a new element, the text box for this element is filled in with whatever was in the element before. Also, if it is a new element, the black box count element is incremented.
FormStandardSwitchInitialize ()	This is method checks if the standard switch element is a new element. If it is not a new element, the text box for this element is filled in with whatever was in the element before. Also, if it is a new element, the black box count element is incremented.
FormSwitchedInitialize ()	This is method checks if the general switch element is a new element. If it is not a new element, the text box for this element is filled in with whatever was in the element before. Also, if it is a new element, the black box count element is incremented.
FormSystematicClosingSwitchInitialize ()	This is method checks if the systematic closing switch element is a new element. If it is not a new element, the text box for this element is filled in with whatever was in the element before. Also, if it is a new element, the black box count element is incremented.
OpeningSwitchOK ()	This method is called by the frmOpeningSwitch cmdOK_click event when all of the details of the switch have been entered. It saves the details to memory.
SlaveClosingSwitchOK ()	This method is called by the frmSlaveClosingSwitch cmdOK_click event when all of the details of the switch have been entered. It saves the details to memory.
SlaveOpeningSwitchOK ()	This method is called by the frmSlaveOpeningSwitch cmdOK_click event when all of the details of the switch have been entered. It saves the details to memory.

Method Name	Description
StandardSwitchOK ()	This method is called by the frmStandardSwitch cmdOK_click event when all of the details of the switch have been entered. It saves the details to memory.
SwitchElementCreate ()	This method is called from the general frmSwitchesElement form, when the user clicks on the cmdCreate command button. This calls the select ^{ed} form, and saves the already selected details to memory.
SwitchElementModify ()	This method is called from the general frmSwitchesElement form, when the user clicks on the cmdModify command button. This calls the selected form, and saves the already selected details to memory, however, the form will automatically include the details of the predefined switch.
SwitchElementOpenCircuit ()	This is called from the OpenCircuit method in the ATP.BAS module. It opens the Switch Element table in the ActiveDataFileName database file, and retrieves the relevant switches from this table.
SwitchElementSaveCircuit (Position As String, i As Integer)	This is called from the SaveCircuit method in the ATP.BAS module. It opens the Switch Element table in the ActiveDataFileName database file, and writes the relevant switches to this table.
SwitchElementWriteCircuit (Position As String, i As Integer)	This method is called by the WriteCircuitSwitchCards method in the ATP.BAS module. It opens the ActiveDataFileName and writes the details of all the present switches to the file.
SwitchElementWriteMiscCard	This method is called by the WriteMiscSwitchCards method in the ATP.BAS module. It opens the ActiveDataFileName and writes the statistics miscellaneous card required for statistical analysis of switching time, to the file.
SwitchElementWriteNodeNames (Number As Integer)	This method adds the node names of all the switch elements to the frmSetupSimulation's lstAvailableNodeNames list box.

Method Name	Description
SystematicClosingSwitchOK ()	This method is called by the frmSystematicClosingSwitch cmdOK_click event when all of the details of the switch have been entered. It saves the details to memory.

2.79.3 List of Module Data Members and Constants

Member / Const Name	Description
Dim SwitchElementNumber As Integer	This element keeps track of which switch is the currently active element.

Member / Const Name	Description
Type GeneralSwitchElement LeftNodeName As String RightNodeName As String SwitchType As Integer CloseTime As String OpenTime As String CurrentThreshold As String OutputType As Integer AveClosingTime As String StandDevJitter As String JitterDeviation As String AveOpeningTime As String StandDevDelay As String AveDelayTime As String LeftMasterNodeName As String RightMasterNodeName As String ConstantDelay As String MeanClosingTime As String TimeIncrement As String NoOfIncrements As String End Type	This Type allows for the storage of all the details relevant to the switch elements. To store these an array is required of this type.
Dim SwitchElements(100) As GeneralSwitchElement	This is the array of switch elements which allows for the storage of all fields relate to switch elements.
Dim NewSwitch As Integer	This is a variable which is used to indicate or check whether or not an element is new
Dim ListOperatingTimes As Integer	This is a flag which is associated with all the switch elements in the circuit if a statistical analysis of switch operating times is performed.
Dim DistributionType As Integer	This is a flag associated with the type of distribution required in a statistical analysis of switch operating times.
Dim IMAX As Integer	This is a flag asso ciated with the statistical analysis of switch operating times.
Dim KSTOUT As Integer	This is a flag associated with the statistical analysis of switch operating times.

Member / Const Name	Description
Dim ITEST2 As Integer	This is a flag associated with the statistical analysis of switch operating times

2.71 The frmSwitchesDiodes Form

2.71.1 Form Description

This form is used to display the Switches and Diodes Library screen.

2.71.2 List of Form Events

None

2.71.3 List Of Important Controls

Control Name	Description
lblLibraryElements	This is the label control used as a heading for the lstLibraryElements list box.
lstLibraryElements	This is a list box control which is used to list the switch element and diode element which the user can select to place in the circuit. Each of these elements has a number of types which can be seen in the lblTypeList label at the bottom of the screen.
lblElementType	This is the label for the list of types of elements lblelementTypeList.
lblElementTypeList	This is a label control which is used to list the types of each element available in ATP for Windows.
cmdOk	This command button is used to indicate that the user has selected the element required, and would like to place it in the circuit. This command button's Click () event calls the SwiShowElementIcon method in the SWDIOLIB.BAS module.

Control Name	Description
cmdCancel	This command button is used to indicate that the user has decided not to select an element from this library, and to exit without any changes.
cmdHelp	This command button is used to get on-line help on the Switches and Diodes Library.

2.72 The SWDIOLIB.BAS Module

2.72.1 Module Description

This module is used to provide functionality for the frmSwitchesDiodes form.

2.72.2 List of Methods

Method Name	Description
SwiShowElementIcon ()	This method checks which element was selected from the Element List and selects the correct icon for this element.

2.72.3 List of Data Members and Constants

None

2.73 The frmSwitchElement Form

2.73.1 Form Description

This form is used to display the Switch Element - Element Details screen.

2.73.2 Form Events

Event Name	Description
Form_Load ()	This method is used to call the FormSwitchElementInitialize method in the SWITCELE.BAS module as well as to disable the Message Bar in the frmATP form.

2.73.3 List Of Important Controls

Control Name	Description
lblSwitchDetails	This is a label control used as a heading for the Switch Details section.
lblLeftNodeName	This is a label control used to label the txtLeftNodeName text box.
txtLeftNodeName	This is the text box used for the entry of the left node name of the switch (Node1 on the switch data card).
lblRightNodeName	This is a label control used to label the txtRightNodeName text box.
txtRightNodeName	This is the text box used for the entry of the left node name of the switch (Node2 on the switch data card).
lblSwitchType	This is a label control used to label the cmdSwitchType combo box.
cmbSwitchType	This is a combo box control which is used to select the switch type.
lblSwitchOperatingTimes	This is a label control used to label the cmdSwitchOperatingTimes combo box.
cmbSwitchOperatingTimes	This is a combo box control which is used to select the listing of switch operating times (ISW on the statistics miscellaneous card).
lblITEST	This is a label control used to label the cmdITEST combo box.

Control Name	Description
cmbITEST	This is a combo box control which is used to select the flag for controlling offset times for random switches (ITEST on the statistics miscellaneous card).
lblDistributionType	This is a label control used to label the cmdDistributionType combo box.
cmbDistributionType	This is a combo box control which is used to select the flag for controlling type of distribution for the random switch operating times (IDIST on the statistics miscellaneous card).
lblIMAX	This is a label control used to label the cmdIMAX combo box.
cmbIMAX	This is a combo box control which is used to select the flag for controlling the listing of deterministic extrema (IMAX on the statistics miscellaneous card).
lblKSSOUT	This is a label control used to label the cmdKSSOUT text box.
cmbKSSOUT	This is a combo box control which is used to select the flag for controlling the printout of additional time step data for each energisation (KSSOUT on the statistics miscellaneous card).
lblITEST2	This is a label control used to label the cmdITEST2 combo box.
cmbITEST2	This is a combo box control which is used to select the flag for controlling whether the systematic switch times T are to be the median or minimum times. (ITEST on the systematic miscellaneous card).
cmdCreate	This command button is used to indicate that the user has completed the entries in this screen relevant to the selected switch, and the Click () event of this button calls the CreateSwitch method in the SWITCELE.BAS module.
cmdModify	This command button is used to user would like to modify the existing switch. The Click () event of this button calls the ModifySwitch method in the SWITCELE.BAS module.

Control Name	Description
cmdCancel	This command button is used to indicate that the user has decided not to select an element from this screen, and to exit without any changes.
cmdHelp	This command button is used to get on-line help on the Switch Element Details screen.

2.74 The frmSystematicClosingSwitch Form

2.74.1 Form Description

This is the Element Details screen for the Systematically closing switch element. It is used to save the variables which are specific to a Systematically closing switch element and are not entered in the General Switch Element Details screen.

2.74.2 List of Form Events

Event Name	Description
Form_Load ()	This method is used to call the FormSystematicClosingSwitchInitialize method in the SYSCLOSW.BAS module as well as to disable the Message Bar in the frmATP form.

2.74.3 List Of Important Controls

Control Name	Description
lblSystematicClosingSwitchDetails	This is a label control which is used as a heading for the screen entries.
lblMeanClosingTime	This is the label for the txtMeanClosingTime text box.
txtMeanClosingTime	This is text box control which is used to enter the mean closing time of the switch.(T entry in the systematically closing switch data card).
lblTimeIncrement	This is the label for the txtTimeIncrement text box.

Control Name	Description
txtTimeIncrement	This is a text box control which is used to enter the time increment for the systematic increase of closing time (DELT entry in the systematically closing switch data card).
lblNoOfIncrements	This is the label for the txtNoOfIncrements text box.
txtNoOfIncrements	This is a text box which is used to enter the number of increments required for the systematic increasing of closing time (NSTEP in the systematically closing switch data card).
lblOutputType	This is the label for the cmbOutputType combo box.
cmbOutputType	This is a combo box control which is used to select the type of output required from this element. The entries in this combo box are loaded in the FormSysCloseSwitchInitialize method in the SYSCLOSW.BAS module.
cmdOk	This is a command button control which is used to indicate that the element details description is complete. The cmdOK_Click event calls the SysCloseSwitchOK method in the SYSCLOSW.BAS module.
cmdCancel	This is a command button control which is used to indicate that the user wishes to exit this screen without saving any of the changes.
cmdHelp	This is a command button control which is used to call the on-line help specific to the systematically closing switch element.

2.75 The frmTACSElementElement Form

2.75.1 Form Description

This form is used to display the TACS Element - Element Details screen.

2.75.2 Form Events

Event Name	Description
Form_Load ()	This method is used to call the FormTACSElementInitialize method in the TACSELEM.BAS module as well as to disable the Message Bar in the frmATP form.

2.75.3 List Of Important Controls

Control Name	Description
lblTACSDetails	This is a label control used as a heading for the TACS Details section.
lblElementName	This is a label control used to label the txtElementName text box.
txtElementName	This is the text box used for the entry of the element name of the TACS element.
lblLeftNodeName	This is a label control used to label the txtLeftNodeName text box.
txtLeftNodeName	This is the text box used for the entry of the left node name of the TACS element (Node1 on the TACS data card).
lblRightNodeName	This is a label control used to label the txtRightNodeName text box.
txtRightNodeName	This is the text box used for the entry of the right node name of the TACS element (Node2 on the TACS data card).
lblTACSType	This is a label control used to label the cmdTACSType combo box.
cmbTACSType	This is a combo box control which is used to select the TACS element type.

Control Name	Description
cmdCreate	This command button is used to indicate that the user has completed the entries in this screen relevant to the selected TACS element, and the Click () event of this button calls the CreateTACSElement method in the TACSELEM.BAS module.
cmdModify	This command button is used to user would like to modify the existing TACS element. The Click () event of this button calls the ModifyTACSElement method in the TACSELEM.BAS module.
cmdCancel	This command button is used to indicate that the user has decided not to select an element from this screen, and to exit without any changes.
cmdHelp	This command button is used to get on-line help on the TACS Element Details screen.

2.76 The TACSELEM.BAS Module

2.76.1 Module Description

This module is used to provide functionality for the frmTACSElement form.

2.76.2 List of Methods

Method Name	Description
ControlElementOK ()	This method is called by the frmControlElement cmdOK_click event when all of the details of the controller topology element have been entered. It saves the details to memory.

Method Name	Description
ControlledIntegratorOK ()	This method is called by the frmControlledIntegrator cmdOK click event when all of the details of the controlled Integrator element have been entered. It saves the details to memory.
FormControlElementInitialize ()	This method is used to initialize the Control Element form. It checks whether the element is a new element or an existing element. If it an existing element, it places the previously stored values in the relevant text boxes.
FormControlledIntegratorInitialize ()	This method is used to initialize the controlled integrator form. It checks whether the element is a new element or an existing element. If it an existing element, it places the previously stored values in the relevant text boxes.
FormFortranInitialize ()	This method is used to initialize the Fortran relationship Element form. It checks whether the element is a new element or an existing element. If it an existing element, it places the previously stored values in the relevant text boxes.
FormInputParametersInitialize ()	This method is used to initialize the Input parameters form. It checks whether the element is a new element or an existing element. If it an existing element, it places the previously stored values in the relevant text boxes.
FormLevelTriggerredSwitchInitialize ()	This method is used to initialize the level triggered switch form. It checks whether the element is a new element or an existing element. If it an existing element, it places the previously stored values in the relevant text boxes.
FormRelayOperatedSwitchInitialize ()	This method is used to initialize the relay operated switch form. It checks whether the element is a new element or an existing element. If it an existing element, it places the previously stored values in the relevant text boxes.

Method Name	Description
FormSimpleDifferentiatorInitialize ()	This method is used to initialize the simple differentiator form. It checks whether the element is a new element or an existing element. If it an existing element, it places the previously stored values in the relevant text boxes.
FormSpecDevInputParametersInitialize ()	This method is used to initialize the special Devices Input Parameters form. It checks whether the element is a new element or an existing element. If it an existing element, it places the previously stored values in the relevant text boxes.
FormSpecialDeviceInitialize ()	This method is used to initialize the TACS Special Device form. It checks whether the element is a new element or an existing element. If it an existing element, it places the previously stored values in the relevant text boxes.
FormTACSCntrCmpntInitialize ()	This method is used to initialize the TACS controlled component form. It checks whether the element is a new element or an existing element. If it an existing element, it places the previously stored values in the relevant text boxes.
FormTACSControlledResistorInitialize ()	This method is used to initialize the TACS controlled resistor form. It checks whether the element is a new element or an existing element. If it an existing element, it places the previously stored values in the relevant text boxes.
FormTACSControlledSwitchInitialize ()	This method is used to initialize the TACS controlled switch form. It checks whether the element is a new element or an existing element. If it an existing element, it places the previously stored values in the relevant text boxes.
FormTACSElementElementInitialize ()	This method is used to initialize the TACS Element main form. It checks whether the element is a new element or an existing element. If it an existing element, it places the previously stored values in the relevant text boxes.

Method Name	Description
FortranElementOK ()	This method is called by the frmFortranElement cmdOK_click event when all of the details of the element have been entered. It saves the details to memory.
InputParametersOK ()	This method is called by the frmInputParameters cmdOK_click event when all of the details of the control element input parameters element have been entered. It saves the details to memory.
LevelTriggerredSwitchOK ()	This method is called by the frmLevelTriggerredSwitch cmdOK_click event when all of the details of the element have been entered. It saves the details to memory.
RelayOperatedSwitchOK ()	This method is called by the frmRelayOperatedSwitch cmdOK_click event when all of the details of the element have been entered. It saves the details to memory.
SimpleDifferentiatorOK ()	This method is called by the frmSimpleDifferentiator cmdOK_click event when all of the details of the element have been entered. It saves the details to memory.
SpecDevInputParametersOK ()	This method is called by the frmSpecDevInputParameters cmdOK_click event when all of the details of the element input parameters have been entered. It saves the details to memory.
SpecialDeviceCreate ()	This method is used to call the element which has been created. It is called by the cmd_Create command button in the frmSpecialDevice form.
SpecialDeviceModify ()	This method is used to call the element which has been created for modification. It is called by the cmd_Modify command button in the frmSpecialDevice form.
TACSControlledComponentCreate ()	This method is used to call the element which has been created. It is called by the cmd_Create command button in the frmTACSControlledComponent form.

Method Name	Description
TACSControlledComponentModify ()	This method is used to call the element which has been created for modification. It is called by the cmd_Modify command button in the frmTACSControlledComponent form.
TACSControlledResistorOK ()	This method is called by the frmTACSControlledResistor cmdOK_click event when all of the details of the element input parameters have been entered. It saves the details to memory.
TACSControlledSwitchCK ()	This method is called by the frmTACSControlledSwitch cmdOK_click event when all of the details of the element input parameters have been entered. It saves the details to memory.
TACSElementCreate ()	This method is used to call the element which has been created. It is called by the cmd_Create command button in the frmTACSElementElement form.
TACSElementModify ()	This method is used to call the element which has been created for modification. It is called by the cmd_Modify command button in the frmTACSElementElement form.
TACSElementOpenCircuit ()	This is called from the OpenCircuit method in the ATP.BAS module. It opens the TACS Element table in the ActiveDataFileName database file, and retrieves the relevant TACS elements from this table.
TACSElementSaveCircuit (Position As String, i As Integer)	This is called from the SaveCircuit method in the ATP.BAS module. It opens the TACS Element table in the ActiveDataFileName database file, and writes the relevant TACS elements to this table.
TACSElementWriteCircuit (Position As String, i As Integer)	This method is called by the WriteBranchCards method in the ATP.BAS module. It opens the ActiveDataFileName and writes the details of all the present TACS elements to the file.
TACSElementWriteNodeNames (Number As Integer)	This method adds the node names of all the TACS elements to the frmSetupSimulation's lstAvailableNodeNames list box.

2.76.3 List of Data Members and Constants

Member / Const Name	Description
Dim TACSElementNumber As Integer	This element keeps track of which switch is the currently active element.
Type TACSElementType ElementName As String LeftNodeName As String RightNodeName As String TypeOfElement As Integer FortranRelationships As String U1 As String (up to U 5) N0 As String (up to N7) D0 As String (up to D7) Gain As String OutputParameterName As String HighestOrderD As String DeviceType As Integer OperatingMode As Integer StatementType As Integer FixedThreshold As String NamedThreshold As String DrivingSignal As String ControlValueName As String ResetName As String ControlledComponentType As String TACSPParameterName As String OutputRequired As Integer CNTRLTACSPParameterName As String End Type	This Type allows for the storage of all the details relevant to the TACS elements. To store these an array is required of this type.
Dim TACSElements(100) As TACSElementType	This is the array of TACS Elements which allows for the storage of all TACS elements including their properties.

Member / Const Name	Description
Dim NewElement As Integer	This is a variable which is used to determine whether a TACS element is a new or already existing element.

2.77 The frmTACSElements Form

2.77.1 Form Description

This form is used to display the TACS element Library screen.

2.77.2 List of Form Events

None

2.77.3 List Of Imports

Control Name	Description
lblLibraryElements	This is the label control used as a heading for the lstLibraryElements list box.
lstLibraryElements	This is a list box control which is used to list the TACS elements which the user can select to place in the circuit. Each of these elements has a number of types which can be seen in the lblTypeList label at the bottom of the screen.
lblElementType	This is the label for the list of types of elements lblelementTypeList.
lblElementTypeList	This is a label control which is used to list the types of each element available in ATP for Windows.
cmdOk	This command button is used to indicate that the user has selected the element required, and would like to place it in the circuit. This command button's Click () event calls the TACS.ShowElementIcon method in the TACSELIB.BAS module.

Control Name	Description
cmdCancel	This command button is used to indicate that the user has decided not to select an element from this library, and to exit without any changes.
cmdHelp	This command button is used to get on-line help on the TACS element Library

2.78 The TACSELIB.BAS Module

2.78.1 Module Description

This module is used to provide functionality for the frmTACSElements form.

2.78.2 List of Methods

Method Name	Description
TACSShowElementIcon ()	This method checks which element was selected from the Element List and selects the correct icon for this element.

2.78.3 List of Data Members and Constants

None

2.79 The frmTACSSources Form

2.79.1 Form Description

This form is used to display the TACS source Library screen.

2.79.2 List of Form Events

None

2.79.3 List Of Important Controls

Control Name	Description
lblLibrarySources	This is the label control used as a heading for the lstLibrarySources list box.
lstLibrarySources	This is a list box control which is used to list the sources which the user can select to place in the circuit. Each of these sources has a number of types which can be seen in the lblTypeList label at the bottom of the screen.
lblSourceType	This is the label for the list of types of sources lblSourceTypeList.
lblSourceTypeList	This is a label control which is used to list the types of each source available in ATP for Windows.
cmdOk	This command button is used to indicate that the user has selected the source required, and would like to place it in the circuit. This command button's Click () event calls the TACSSourceShowSourceIcon method in the TACSSLIB.BAS module.
cmdCancel	This command button is used to indicate that the user has decided not to select an source from this library, and to exit without any changes.
cmdHelp	This command button is used to get on-line help on the TACS source Library.

2.80 The TACSSLIB.BAS Module**2.80.1 Module Description**

This module is used to provide functionality for the frmTACSSources form.

2.80.2 List of Methods

Method Name	Description
TACSSourceShowSourceIcon ()	This method checks which source was selected from the Element List and selects the correct icon for this source.

2.80.3 List of Data Members and Constants

None

2.81 The frmTACSSource Form**2.81.1 Form Description**

This form is used to display the TACS Source - Element Details screen.

2.81.2 Form Events

Event Name	Description
Form_Load ()	This method is used to call the FormTACSSourceInitialize method in the TACSSRCE.BAS module as well as to disable the Message Bar in the frmATP form.

2.81.3 List Of Important Controls

Control Name	Description
lblTACSDetails	This is a label control used as a heading for the TACS Details section.
lblSourceName	This is a label control used to label the txtSourceName text box.
txtSourceName	This is the text box used for the entry of the element name of the TACS source.

Control Name	Description
lblNodeName	This is a label control used to label the txtNodeName text box.
txtNodeName	This is the text box used for the entry of the node name of the TACS source (the node which isn't connected to ground) (Node on the TACS data card).
lblTACSType	This is a label control used to label the cmdTACSType combo box.
cmbTACSType	This is a combo box control which is used to select the TACS source type.
cmdCreate	This command button is used to indicate that the user has completed the entries in this screen relevant to the selected TACS, and the Click () event of this button calls the CreateTACSSource method in the TACSSRCE.BAS module.
cmdModify	This command button is used to user would like to modify the existing TACS source. The Click () event of this button calls the ModifyTACSSource method in the TACSSRCE.BAS module.
cmdCancel	This command button is used to indicate that the user has decided not to select an element from this screen, and to exit without any changes.
cmdHelp	This command button is used to get on-line help on the TACS Source Details screen.

2.82 The TACSSRCE.BAS Module

2.82.1 Module Description

This module is used to provide functionality for the frmTACSSource form.

2.82.2 List of Methods

Method Name	Description
CreateTACSSource ()	This method is used to create a TACS source. This is done by simply checking which type of TACS source was selected, and then opening the relevant form with the default values used. The form for that TACS source will save the values in the frmTACSSource form.
ModifyTACSSource ()	This method is used to modify an existing TACS source. This is done by simply checking which TACS source was selected, and then opening the relevant form with the saved values used. The form for that TACS source will save the values in the frmTACSSource form again.

2.82.3 List of Data Members and Constants

Member / Const Name	Description
Dim NodeName As String	This variable is used to store the name of the node of the TACS source which is not connected to earth.
Dim SourceName As String	This variable is used to store the name of the TACS source for use in the description of other elements if necessary.
Dim TACSSourceType As Integer	This variable is used to select the TACS type required.

2.83 The frmTACSControlledComponent Form**2.83.1 Form Description**

This form is used to display the TACS Controlled Element - Element Details screen.

2.83.2 List of Form Events

Event Name	Description
Form_Load ()	This method is used to call the FormTACSCtrlComponentInitialize method in the TCNTRLEL.BAS module as well as to disable the Message Bar in the frmATP form.

2.83.3 List Of Important Controls

Control Name	Description
lblTACSCtrlComponentType	This is the label control used for the heading of the TACS Controlled Component Types.
lblDeviceType	This is the label control used to label the cmdDeviceType combo box.
cmbDeviceType	This combo box is used to select the type of TACS controlled component required.
cmdOk	This command button is selected once the user has completed the selection of the type of TACS controlled element, and would like to create such an element. The Click () event of this control calls the CreateTACSCtrlComponent method in the TCNTRLEL.BAS module.
cmdCancel	This command button is used to indicate that the user has decided not to select an element from this screen, and to exit without any changes.
cmdHelp	This command button is used to get on-line help on the TACS Controlled Component screen.

2.84 The TCNTRLEL.BAS Module

2.84.1 Module Description

This module is used to provide functionality for the frmTACSCtrlComponent form.

2.84.2 List of Methods

Method Name	Description
FormTACSCntrlComponentInitialize ()	This method is used to initialize the frmTACSCntrlComponent form by adding the relevant default values of the different entries to the relevant text boxes or combo boxes in this method.
CreateTACSCntrlComponent ()	This method is used to create a TACS controlled component. This is done by checking which type of component was selected and then loading the relevant form.

2.84.3 List of Data Members and Constants

None

2.85 The frmTACSCntrlResistor Form**2.85.1 Form Description**

This form is used to display the TACS Controlled Resistor - Element Details screen.

2.85.2 List of Form Events

Event Name	Description
Form_Load ()	This method is used to call the FormTACSCntrlResistorInitialize method in the TCNTRRES.BAS module as well as to disable the Message Bar in the frmATP form.

2.85.3 List Of Important Controls

Control Name	Description
lblTACSCntrlResistorDetails	This is the label control used for the heading of the TACS Controlled Resistor Details section.

Control Name	Description
lblTACSPParameterName	This is the label control used to label the txtTACSPParameterName text box.
cmdOk	This command button is selected once the user has completed the details of this screen. The Click () event of this control calls the TACSCntrlResisOK method in the TCNTRRES.BAS module.
cmdCancel	This command button is used to indicate that the user has decided not to select an element from this screen, and to exit without any changes.
cmdHelp	This command button is used to get on-line help on the TACS Controlled Resistor screen.

2.86 The TCNTRRES.BAS Module

2.86.1 Module Description

This module is used to provide functionality for the frmTACSControlledResistor form.

2.86.2 List of Methods

Method Name	Description
FormTACSCntrlResistorInitialize ()	This method is used to place the default value of the TACS Parameter name in the text box for this entry.
TACSCntrlResistorOK ()	This method is used to save all the values entered in the TACS controlled resistor screen.
TACSCntrlResistorWriteAllValues ()	This method is called by the RunRunSimulation method in order to write the details of the TACS Controlled Resistor to the TACS_Controlled_Resistor class in the Data Case Module.

2.86.3 List of Data Members and Constants

Member / Const Name	Description
Dim TACSPParameterName As String	This is the name of the TACS parameter which will provide the value of the TACS controlled resistor.

2.87 The frmTACSControlledSource Form**2.87.1 Form Description**

This form is used to display the TACS Controlled Source - Element Details screen.

2.87.2 List of Form Event

Event Name	Description
Form_Load ()	This method is used to call the FormTACSCntrlSourceInitialize method in the TCNTRSRC.BAS module as well as to disable the Message Bar in the frmATP form.

2.87.3 List Of Important Controls

Control Name	Description
lblTACSControlledSourceDetails	This is the label control used for the heading of the TACS Controlled Source Details section.
lblStartTime	This is the label control used to label the txtStartTime text box.
txtStartTime	This text box is used to enter the start time of the source (T-START in the TACS controlled source card).
lblStopTime	This is the label control used to label the txtStopTime text box.

Control Name	Description
txtStopTime	This text box is used to enter the stop time of the source (T-STOP in the TACS controlled source card).
lblSourceType	This is the label control used to label the cmbSourceType combobox.
cmbSourceType	This is the combo box control used to select the type of source required (IV in the TACS controlled source card).
lblTACSPParameterName	This is the label control used to label the txtTACSPParameterName text box.
cmdOk	This command button is selected once the user has completed the data of this screen. The Click () event of this control calls the TACSCntrlSourceOK method in the TCNTRSRC.BAS module.
cmdCancel	This command button is used to indicate that the user has decided not to select an element from this screen, and to exit without any changes.
cmdHelp	This command button is used to get on-line help on the TACS Controlled Source screen.

2.88 The TCNTRSRC.BAS Module

2.88.1 Module Description

This module is used to provide functionality for the frmTACSControlledSource form.

2.88.2 List of Methods

Method Name	Description
FormTACSCntrlSourceInitialize ()	This method is used to place the default value of the entries in this screen in the correct controls.
TACSCntrlSourceOK ()	This method is used to save all the values entered in the TACS controlled source screen.

Method Name	Description
TACSCntrlSourceWriteAllValues ()	This method is called by the RunRunSimulation method in order to write the details of the TACS Controlled Source to the TACS_Controlled_Source class in the Data Case Module.

2.88.3 List of Data Members and Constants

Member / Const Name	Description
Dim TACSPParameterName As String	This is the name of the TACS parameter which will provide the value of the TACS controlled source.
Dim StartTime As String	This variable stores the starting time of the source.
Dim StopTime As String	This variable stores the stopping time of the source.
Dim SourceType As Integer	This variable is used to store the type of source required.

2.89 The frmTACSControlledSwitch Form

2.89.1 Form Description

This form is used to display the TACS Controlled Switch - Element Details screen.

2.89.2 List of Form Events

Event Name	Description
Form_Load ()	This method is used to call the FormTACSCntrlSwitchInitialize method in the TCNTRSWI.BAS module as well as to disable the Message Bar in the frmATP form.

2.89.3 List Of Important Controls

Control Name	Description
lblTACSControlledSwitchDetails	This is the label control used for the heading of the TACS Controlled Switch Details section.
lblCONTROLParameterName	This is the label control used to label the txtCONTROLParameterName text box.
txtCONTROLParameterName	This is a text box which is used to enter the name of the parameter which will supply the CONTROL value for the switch (CONTROL in the TACS controlled switch card).
lblOutputRequired	This is the label used for the cmbOutputRequired combo box.
cmbOutputRequired	This combo box control is used to select the type of output required from this source (P in the TACS controlled switch card).
cmdOk	This command button is selected once the user has completed the details of this screen. The Click () event of this control calls the TACSCntrlSwitchOK method in the TCNTRSWI.BAS module.
cmdCancel	This command button is used to indicate that the user has decided not to select an element from this screen, and to exit without any changes.
cmdHelp	This command button is used to get on-line help on the TACS Controlled Switch screen.

2.90 The TCNTRSWI.BAS Module**2.90.1 Module Description**

This module is used to provide functionality for the frmTACSControlledSwitch form.

2.90.2 List of Methods

Method Name	Description
FormTACSCntrlSwitchInitialize ()	This method is used to place the default values of the entries on this screen.
TACSCntrlSwitchOK ()	This method is used to save all the values entered in the TACS controlled switch screen.
TACSCntrlSwitchWriteAllValues ()	This method is called by the RunRunSimulation method in order to write the details of the TACS Controlled Switch to the TACS_Controlled_Switch class in the Data Case Module.

2.90.3 List of Data Members and Constants

Member / Const Name	Description
Dim CONTROLName As String	This is the name of the TACS parameter which will provide the value of the CONTROL parameter for the TACS controlled switch.
Dim OutputRequired As Integer	This variable is used to store the type of output required from this switch.

2.91 The frmTempImages Form**2.91.1 Form Description**

This form is used as a clip board to store images when they are cut or copied from frmATP. This form also contains all the images which are not in the actual cmd3dCircuitPanel, but are required for certain elements such as the three phase transformer (top, bottom, stardelta, etc), and the top and bottom images of the transmission line element and 6 conductor transmission line.

2.92 The frmLeftTopImages Form**2.92.1 Form Description**

This form is used to store images in the left top area of the entire circuit region when the user

scrolls circuit elements into the left top region.

2.93 The frmTopImages Form

2.93.1 Form Description

This form is used to store images in the left top area of the entire circuit region when the user scrolls circuit elements into the top region.

2.94 The frmTransformer Form

2.94.1 Form Description

This form is the screen for the Single phase Transformer - Element Details screen.

2.94.2 List of Form Data Members and Constants

Same as Module data members for all entries in frames with scroll bars.

2.94.3 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormSinglePhaseTransformerInitialize method in the TRNSFRMR.BAS module.

2.94.4 List Of Important Controls

Control Name	Description
lblTransformerDetails	This is the label for the heading of the Transformer entries.
lblLeftNodeName1	This is the label for the txtLeftNodeName1 text box.

Control Name	Description
txtLeftNodeName1	This is the text control which is used to enter the left node name of the 1st winding of this phase of the transformer (NODE1 in the winding cards for the Transformer data card).
lblRightNodeName1	This is the label for the txtRightNodeName1 text box.
txtRightNodeName1	This is the text control which is used to enter the right node name of the 1st winding of this phase of the transformer (NODE2 in the winding cards for the Transformer data card).
lblWindingResistance1	This is the label for the txtWindingResistance1 text box.
txtWindingResistance1	This is the text control which is used to enter the winding resistance of the 1st winding of this phase of the transformer (Rk in the winding cards for the Transformer data card).
lblLeakageInductance1	This is the label for the txtLeakageInductance1 text box.
txtLeakageInductance1	This is the text control which is used to enter the leakage inductance of the 1st winding of this phase of the transformer (Lk in the winding cards for the Transformer data card).
lblNoOfTurns1	This is the label for the txtNoOfTurns1 text box.

Control Name	Description
txtNoOfTurns1	This is the text control which is used to enter the number of turns of the 1st winding of this phase of the transformer (Nk in the winding cards for the Transformer data card).
lblLeftNodeName2	This is the label for the txtLeftNodeName2 text box.
txtLeftNodeName2	This is the text control which is used to enter the left node name of the 2nd winding of this phase of the transformer (NODE1 in the winding cards for the Transformer data card).
lblRightNodeName2	This is the label for the txtRightNodeName2 text box.
txtRightNodeName2	This is the text control which is used to enter the right node name of the 2nd winding of this phase of the transformer (NODE1 in the winding cards for the Transformer data card).
lblWindingResistance2	This is the label for the txtWindingResistance2 text box.
txtWindingResistance2	This is the text control which is used to enter the winding resistance of the 2nd winding of this phase of the transformer for the current-flux characteristics of the transformer (Rk in the winding cards for the Transformer data card).
cmbCurrentFluxCharacteristics	This is the command button used to call the frmCurrentFluxCharacteristics form which allows the user to enter the current flux characteristics required for this transformer.
lblOutputType	This is the label for the cmbOutputType combo box.
cmbOutputType	This is a combo box for the type of output required from this branch of the circuit (P in the Transformer data card).
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this single phase transformer element. This is done by calling the SinglePhaseTransformerOK method.

Control Name	Description
cmdCancel	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.
cmdHelp	This Command Button is used to open the on-line help screen related to the Single phase Transformer element and it's properties.

2.95 The TRNSFRMR.BAS Module

2.95.1 Module Description

This module provides functionality for the frmSinglePhaseTransformer form. It contains a Type structure of the same name as the module with the members listed below (an array of variables of this type are then created with each instance of an element representing one member of this array).

2.95.2 List Of Methods

Method Name	Description
FormTransformerInitialize ()	This method is used to initialize the frmSinglePhaseTransformer form by adding the relevant default values of the different entries are also placed in the relevant text boxes or combo boxes in this method.
TransformerOK ()	This method is used to save all the different choices and values from the Single phase Transformer - Element Details screen to memory.
TransformerOpenCircuit ()	This method is called by the OpenCircuit method in the ATP.BAS module. It opens the Transformer table in the active database file, and retrieves the relevant transformer elements from this file.
TransformerSaveCircuit (Position As String, i As Integer)	This method is called by the SaveCircuit method in the ATP.BAS module. It opens the Transformer table in the active database file, and saves the relevant transformer elements to this file.

Method Name	Description
TransformerWriteNodeNames (Number As Integer)	This method is called by the WriteNodeNames method in the ATP.BAS module. It adds the node names of all transformers to the frmSetupSimulation's lstAvailableNodeNames list box.
TransformerWriteCircuit ()	This method is called by the WriteTransformerCards method in the ATP.BAS module. It opens the ActiveDataFileName and writes the required lines to this file.

2.95.3 List of Data Members and Constants

Member / Const Name	Description
Dim TransformerNumber As Integer	This variable is used to keep track of the number of the transformer element which is presently active.
Type TransformerType TransformerName As String ISTDY As String FluxSTDY As String ReferenceName As String CoreLoss As String LeftNodeName As String RightNodeName As String WindingResistanceLeft As String WindingResistanceRight As String LeakageInductanceLeft As String LeakageInductanceRight As String NumberOfTurnsLeft As String NumberOfTurnsRight As String OutputType As Integer CurrentFluxValues As String End Type	This is the defining type of the transformer element. It includes variables to store all details relevant to the transformer element. To use this an array of this type must be created (see below).

Member / Const Name	Description
Dim Transformers(100) As TransformerType Dim NewElement As Integer	This is the array of transformers to which every field in the transformer details is saved.

2.96 The frmTransformer3Phase Form

2.96.1 Form Description

This form is used to display the Transformer Element - Element Details screen which appears just before the Single Phase, Shell Type, or Core Type transformers.

2.96.2 Form Events

Event Name	Description
Form_Load ()	This method is used to call the FormTransformerInitialize method in the TRANSFRM.BAS module as well as to disable the Message Bar in the frmATP form.

2.96.3 List Of Important Controls

Control Name	Description
lblTransformerDetails	This is the label control which is used as the heading for the Transformer Details section.
lblReferenceName	This is the label for the txtReferenceName text box.
txtReferenceName	This text box is used to enter the reference name of an identical transformer if such a transformer exists.
lblISTDY	This is the label for the txtISTDY text box.
txtISTDY	This text box is used to enter the current component of the point on the linear characteristic of the flux vs magnetising current characteristic to be used in the steady state initialisation (ISTDY in the transformer card).

Control Name	Description
lbl ϕ STDY	This is the label for the txt ϕ STDY text box.
txt ϕ STDY	This text box is used to enter the flux component of the point on the linear characteristic of the flux vs magnetising current characteristic to be used in the steady state initialisation (ϕ STDY in the transformer card).
lblTransformerName	This is the label for the txtTransformerName text box.
txtTransformerName	This text box is used to enter the unique name of the transformer which can then later be used as a reference name for another transformer (BUSTOP in the transformer card).
lblCoreLoss	This is the label for the txtCoreLoss text box.
txtCoreLoss	This text box is used to enter the core loss of the transformer (RMAG in the transformer card).
lblTransformerType	This is the label for the cmbTransformerType combo box.
cmbTransformerType	This combo box is used to indicate what type of three phase transformer is required (shell type or core type) in the case of the user selecting the three phase transformer. This does not appear in the case of the single phase transformer.
txtZeroSequencyFluxPath	This is the text box used for the entry of the zero sequence flux path reluctance in the case of core type transformers.
lblOutputType	This is the label for the cmbOutputType combo box.
cmbOutputType	This is the combo box which is used to select the type of output required from this branch (P in the transformer card).
cmdCurrentFluxCharacteristics	This is the command button used to call the frmCurrentFluxCharacteristics form which allows the user to enter the current flux characteristics required for this transformer.

Control Name	Description
cmdLeftWindingA	This command button calls the form for the element details of the left / primary winding A.
cmdLeftWindingB	This command button calls the form for the element details of the left / primary winding B.
cmdLeftWindingC	This command button calls the form for the element details of the left / primary winding C.
cmdRightWindingA	This command button calls the form for the element details of the right / secondary winding A.
cmdRightWindingB	This command button calls the form for the element details of the right / secondary winding B.
cmdRightWindingC	This command button calls the form for the element details of the right / secondary winding C.
cmdExtraWindingA	This command button calls the form for the element details of any extra winding A.
cmdExtraWindingB	This command button calls the form for the element details of any extra winding B.
cmdExtraWindingC	This command button calls the form for the element details of any extra winding C.
cmdBctran	This command button is used to call the frmBctran form. If this choice is made, the rest of the element details on this form should not be filled in.
cmdModify	This command button is used to user would like to modify an existing transformer. The Click () event of this button calls the ModifyTransformer method in the TRNSFRM3.BAS module.
cmdCancel	This command button is used to indicate that the user has decided not to select an element from this screen, and to exit without any changes.
cmdHelp	This command button is used to get on-line help on the Transformer Details screen.

2.97 The TRNSFRM3.BAS Module

2.97.1 Module Description

This module is used to provide functionality for the frmTransformer form.

2.97.2 List of Methods

Method Name	Description
BctranMoreOk ()	This method is called from the frmBctranMore form to indicate that the user has completed the entries in this form. The entries are saved to memory.
BctranOK ()	This method is called from the frmBctran form to indicate that the user has completed the entries in this form. The entries are saved to memory.
BctranTestDataOK ()	This method is called from the frmBctranTestData form to indicate that the user has completed the entries in this form. The entries are saved to memory.
CurrentFluxSaveValues ()	This method is called from the cmdNext command button in the frmCurrentFluxCharacteristics. It stores the value in the txtCurrent and txtFlux text boxes in memory.
FormBctranInitialize ()	This method is used to check whether or not the current element is active, and then initialize the entries on the form accordingly.
FormBctranMoreInitialize ()	This method is used to check whether or not the current element is active, and then initialize the entries on the form accordingly.
FormBctranTestDataInitialize ()	This method is used to check whether or not the current element is active, and then initialize the entries on the form accordingly.
FormCurrentFluxInitialize ()	This method is used to check whether or not the current element is active, and then initialize the entries on the form accordingly.

Method Name	Description
FormThreePhaseInitialize ()	This method is used to check whether or not the current element is active, and then initialize the entries on the form accordingly.
FormWindingInitialize ()	This method is used to check whether or not the current element is active, and then initialize the entries on the form accordingly. This is used for the frmWinding form for all the windings.
Function Get3PhaseCurrent (i As Integer) As String	This function is called from the frmCurrentFluxCharacteristics form to determine what the next current entry is when the user clicks on the cmdNext button in an already existing element.
Function Get3PhaseFlux (i As Integer) As String	This function is called from the frmCurrentFluxCharacteristics form to determine what the next flux entry is when the user clicks on the cmdNext button in an already existing element.
ThreePhaseTransformerOK ()	This method is called from the frm3PhaseTransformer form to indicate that the user has completed the entries in this form. The entries are saved to memory.
ThreePhaseTransformerOpenCircuit ()	This method is called by the OpenCircuit method in the ATP.BAS module. It opens the Three Phase Transformer table in the active database file, and retrieves the relevant transformer elements from this file.
ThreePhaseTransformerSaveCircuit (Position As String, i As Integer)	This method is called by the SaveCircuit method in the ATP.BAS module. It opens the Three Phase Transformer table in the active database file, and saves the relevant transformer elements to this file.
ThreePhaseTransformerWriteCircuit (Position As String, i As Integer)	This method is called by the WriteTransformerCards method in the ATP.BAS module. It opens the ActiveDataFileName and writes the required lines to this file.
ThreePhaseTransformerWriteNodeNames (Number As Integer)	This method is called by the WriteNodeNames method in the ATP.BAS module. It adds the node names of all transformers to the frmSetupSimulation's lstAvailableNodeNames list box.

Method Name	Description
WindingOK (Number As Integer)	This method is called from the frmWinding forms to indicate that the user has completed the entries in these forms. The entries are saved to memory. The number of the winding form is passed to this method to indicate which variable to store the values to in the three phase transformer array (see data members)

2.97.3 List of Data Members and Constants

Member / Const Name	Description
Dim ThreePhaseTransformerNumber As Integer	This variable is used to keep track of the number of the transformer element which is presently active.
Type ThreePhTransformerType ReferenceName As String ISTDY As String FluxSTDY As String TransformerName As String CoreLoss As String TransformerType As Integer OutputType As Integer LeftNodeName(8) As String RightNodeName(8) As String WindingResistance(8) As String LeakageInductance(8) As String NumberOfTurns(8) As String OutputRequired(8) As Integer Current(50) As String Flux(50) As String NumberOfSteps As Integer ZeroSequenceFluxReluctance As String End Type	This is the defining type of the transformer element. It includes variables to store all details relevant to the transformer element. To use this an array of this type must be created (see below).
Dim ThreePhaseTransformers(100) As ThreePhTransformerType	This is the array of three phase transformers which is used to store all the fields of every transformer in the circuit.

Member / Const Name	Description
Dim NewElement As Integer	This variable is used to determine whether a three phase transformer is a new element or not.
Dim NewWinding1 As Integer	This variable is used to determine whether left winding 1 has been previously entered for this transformer.
Dim NewWinding2 As Integer	This variable is used to determine whether left winding 2 has been previously entered for this transformer.
Dim NewWinding3 As Integer	This variable is used to determine whether left winding 3 has been previously entered for this transformer.
Dim NewWindingRight1 As Integer	This variable is used to determine whether right winding 1 has been previously entered for this transformer.
Dim NewWindingRight2 As Integer	This variable is used to determine whether right winding 2 has been previously entered for this transformer.
Dim NewWindingRight3 As Integer	This variable is used to determine whether right winding 3 has been previously entered for this transformer.
Global New3PhaseElement	This variable is used by the frmCurrentFluxCharacteristics to determine whether or not the current flux characteristics have previously been defined.

2.98 The frmTransmissionConductor Form

2.98.1 Form Description

This form is the screen for the transmission Conductor Element Details screen.

2.98.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormTransmissionConductorInitialize method in the TRNSLINE.BAS module.

2.98.3 List Of Important Controls

Control Name	Description
txtLeftNodeName	This is the text box used to enter the left node name of the conductor.
txtRightNodeName	This is the text box used to enter the right node name of the conductor.
txtPhaseNumber	This is the text box used to enter the phase number of this conductor.
txtSkinRatio	This is the text box used to enter the skin ration of the conductor.
txtResistance	This is the text box used to enter the resistance of the conductor.
txtDiameter	This is the text box used to enter the diameter of the conductor.
txtHorizontalSeparation	This is the text box used to enter the horizontal separation of the conductor from some arbitrary reference point.
txtTowerHeight	This is the text box used to enter the tower height of the conductor.
txtMidspanHeight	This is the text box used to enter the midspan height of the conductor.

Control Name	Description
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this user defined source element. This is done by calling the TransmissionConductorOK method.
cmdCancel	This Command Button is used to open the screen related to the transmission conductor properties.
cmdHelp	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.

2.99 The frmTransformers Form

2.99.1 Form Description

This form is used to display the Transformer element Library screen.

2.99.2 List of Form Events

None

2.99.3 List Of Important Controls

Control Name	Description
lblLibraryElements	This is the label control used as a heading for the lstLibraryElements list box.
lstLibraryElements	This is a list box control which is used to list the Transformer elements which the user can select to place in the circuit. Each of these elements has a number of types which can be seen in the lblTypeList label at the bottom of the screen.

Control Name	Description
lblElementType	This is the label for the list of types of elements lblelementTypeList.
lblElementTypeList	This is a label control which is used to list the types of each element available in ATP for Windows.
cmdOk	This command button is used to indicate that the user has selected the element required, and would like to place it in the circuit. This command button's Click () event calls the TransformerShowElementIcon method in the TransformerELIB.BAS module.
cmdCancel	This command button is used to indicate that the user has decided not to select an element from this library, and to exit without any changes.
cmdHelp	This command button is used to get on-line help on the Transformer element Library.

2.100 The TRNSFLIB.BAS Module

2.100.1 Module Description

This module is used to provide functionality for the frmTransformers form.

2.100.2 List of Methods

Method Name	Description
TransformerShowElementIcon ()	This method checks which element was selected from the Element List and selects the correct icon for this element.

2.100.3 List of Data Members and Constants

None

2.101 The frmTransmissionLine Form**2.101.1 Form Description**

This form is the screen for the transmission Line Element Details screen.

2.101.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormTransmissionLineInitialize method in the TRNSLINE.BAS module.

2.101.3 List Of Important Controls

Control Name	Description
txtLineLength	This is user for the entry of the length of the transmission line.
txtGroundResistivity	This text box is used to enter the ground resistivity.
txtCalculationFrequency	This text box is used to enter the calculation frequency for the transformation matrix calculations in the transmission line model.

Control Name	Description
cmbTransmissionLineModel	This combo box is used to offer the user the three model options available to the user.
cmbEquivalentPiOutput	This combo box is used to offer the user the options for the output of an equivalent pi model of the transmission line.
cmbContinuousGroundWires	This combo box allows the user to select whether or not the ground wires in the transmission lines are continuous (either continuous or segmented).
cmbParallelCommunicationCircuit	This combo box allows the user to select whether or not there is a communication circuit parallel to the line.
cmbLineUntransposed	This combo box allows the user to select whether or not the line is untransposed.
cmdTransmissionLineConductor1	This command button is used to open the frmTransmissionConductor for the 1st conductor.
cmdTransmissionLineConductor2	This command button is used to open the frmTransmissionConductor for the 2nd conductor.
cmdTransmissionLineConductor3	This command button is used to open the frmTransmissionConductor for the 3rd conductor.
cmdTransmissionLineConductor4	This command button is used to open the frmTransmissionConductor for the 4th conductor.
cmdTransmissionLineConductor5	This command button is used to open the frmTransmissionConductor for the 5th conductor.
cmdTransmissionLineConductor6	This command button is used to open the frmTransmissionConductor for the 6th conductor.
cmdTransmissionLineConductor7	This command button is used to open the frmTransmissionConductor for the 7th conductor.
cmdTransmissionLineConductor8	This command button is used to open the frmTransmissionConductor for the 8th conductor.
cmdJMarti	This command button is used to open the frmJMarti form which allows the user to enter parameters for the JMARTI add on program.

Control Name	Description
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this user defined source element. This is done by calling the TransmissionLineOK method.
cmdCancel	This Command Button is used to open the on-line help screen related to the transmission line and it's properties.
cmdHelp	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.

2.102 The TRNSLINE.BAS Module

2.102.1 Module Description

This module is used to provide functionality for the frmTransmissionLine, frmJMarti and frmTransmissionConductor forms.

2.102.2 List of Methods

Method Name	Description
FormJMartiInitialize ()	This method is used to check whether or not the current element is active, and then initialize the entries on the form accordingly.
FormTransmissionConductorInitialize ()	This method is used to check whether or not the current element is active, and then initialize the entries on the form accordingly.
FormTransmissionLineInitialize ()	This method is used to check whether or not the current element is active, and then initialize the entries on the form accordingly.
JmartiOK ()	This method is called from the frmJMarti form to indicate that the user has completed the entries in this form. The entries are saved to memory.

Method Name	Description
TransmissionConductorOK ()	This method is called from the frmTransmissionConductor form to indicate that the user has completed the entries in this form. The entries are saved to memory
TransmissionLineOK ()	This method is called from the frmTransmissionLine form to indicate that the user has completed the entries in this form. The entries are saved to memory.
TransmissionLineOpenCircuit ()	This method is called by the OpenCircuit method in the ATP.BAS module. It opens the Transmission Line table in the active database file, and retrieves the relevant transformer elements from this file.
TransmissionLineSaveCircuit (Position As String, i As Integer)	This method is called by the SaveCircuit method in the ATP.BAS module. It opens the Transmission Line table in the active database file, and saves the relevant transformer elements to this file.
TransmissionLineWriteCircuit (Position As String, i As Integer)	This method is called by the WriteBranchCards method in the ATP.BAS module. It opens the ActiveDataFileName and writes the required lines to this file.
TransmissionLineWriteLineConstants (Position As String, i As Integer)	This method is called by the WriteLineConstantsCards method in the ATP.BAS module. It opens the ActiveDataFileName and writes the required lines to this file.
TransmissionLineWriteNodeNames (Number As Integer)	This method is called by the WriteNodeNames method in the ATP.BAS module. It adds the node names of all transmission lines to the frmSetupSimulation's lstAvailableNodeNames list box.

2.102.3 List of Data Members and Constants

Member / Const Name	Description
Dim TransmissionLineNumber As Integer	This variable is used to keep track of the number of the transmission line element which is presently active.

Member / Const Name	Description
Type TransmissionLineType ElementName As String LineLength As String GroundResistivity As String CalculationFrequency As String TransmissionLineModel As Integer ContinuousGroundWires As Integer SegmentedGroundWires As Integer ParallelCommsCct As Integer LineUntransposed As Integer DiagonalizationFrequency As String SteadyStateFrequency As String LowestFrequency As String NumberOfDecades As String PointsPerDecade As String ConductorLeftNodeName(7) As String ConductorRightNodeName(7) As String ConductorPhaseNumber(7) As String ConductorSkinRatio(7) As String ConductorResistance(7) As String ConductorDiameter(7) As String HorizontalSeparation(7) As String TowerHeight(7) As String MidspanHeight(7) As String EquivalentPiType As Integer LineConstantsName As String IncludeFileName As String End Type	This is the defining type of the transmission line element. It includes variables to store all details relevant to the transmission line element. To use this an array of this type must be created (see below).

Member / Const Name	Description
Dim TransmissionLines(100) As TransmissionLineType	This is the array of transmission lines which is used to store all the fields of every transmission line in the circuit.
Dim NewElement As Integer	This variable is used to determine whether a transmission line is a new element or not.
Dim NewJMarti As Integer	This variable is used to determine whether the Jmarti details have been previously entered for this transmission line.
Dim NewConductor(8) As Integer	This is used to determine if the any of the conductors' details have been previously entered for this element.
Global ConductorNumber As Integer	This variable is used by the frmJMarti to determine whether or not the current flux characteristics have previously been defined.

2.103 The frmUserDefinedSource Form

2.103.1 Form Description

This form is the screen for the User Defined Source - Element Details screen.

2.103.2 List of Form Events

Event Name	Description
Form_Load ()	This form is used to call the FormUserDefSourceInitialize method in the USERDFSR.BAS module.

2.103.3 List Of Important Controls

Control Name	Description
lblSourceDetails	This is the label for the user defined source details section.

Control Name	Description
lblNodeName	This is the label for the txtNodeName text box.
txtNodeName	This is the text box entry for the Node name of the node which is connected to the source (the other node is connected to earth automatically) (NODE in the user defined source data card).
lblStartTime	This is the label for the txtStartTime text box.
txtStartTime	This is the text box entry for the start time of the source (TSTART in the user defined source data card).
lblStopTime	This is the label for the txtStopTime text box.
txtStopTime	This is the text box entry for the stop time of the source (TSTOP in the user defined source data card).
lblNoOfTimeSteps	This is the label for the txtNoOfTimeSteps text box.
txtNoOfTimeSteps	This is the text box entry for the number of time steps required for the definition of the source.
lblUserDefSourceType	This is the label control used to label the cmbUserDefSourceType combo box.
cmbUserDefSourceType	This combo box is used to select the type of user defined source required (voltage or current) (IV in the user defined source card).
cmdOk	This Command Button is used to indicate that the data file has now been edited and once this button is selected, the data in this screen will automatically be associated with this user defined source element. This is done by calling the UserDefSourceOK method.
cmdCancel	This Command Button is used to open the on-line help screen related to the user defined source element and it's properties.
cmdHelp	This Command Button control is used to Cancel this operation without saving any of the details to file or memory.

2.104 The USERDFSR.BAS Module

2.104.1 Module Description

This module provides functionality for the frmUserDefinedSource Form. It contains a Type structure of the same name as the module with the members listed below (an array of variables of this type are then created with each instance of an element representing one member of this array).

2.104.2 List of Methods

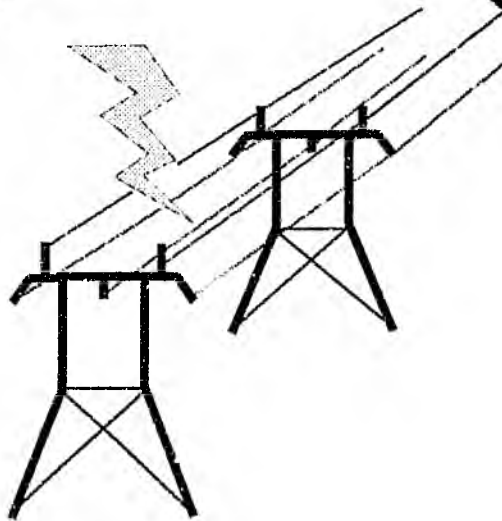
Method Name	Description
FormUserDefSourceInitialize ()	This method is used to initialize the frmUserDefinedSource form by adding the relevant default values of the different entries are also placed in the relevant text boxes or combo boxes in this method.
UserDefSourceOK ()	This method is used to save all the different choices and values from the User Defined Source - Element Details screen to memory.
UserDefSourceOpenCircuit ()	This method is called by the OpenCircuit method in the ATP.BAS module. It opens the User Defined Source table, and retrieves the details of all the user defined sources in the circuit.
UserDefSourceSaveCircuit (Position As String, i As Integer)	This method is called by the SaveCircuit method in the ATP.BAS module. It opens the User Defined Source table, and writes the details of all the user defined sources in the circuit to the table.
UserDefSourceWriteNodeNames (Number As Integer)	This method is called by the WriteNodeNames method in the ATP.BAS module. It adds the node names of all the user defined sources in the circuit to the frmSetupSimulation's lstAvailableNodes list box.
UserDefSourceWriteCircuit (Position As String, i As Integer)	This method is called by the WriteSourceCards method in the ATP.BAS module. It writes the relevant lines to the currently open ActiveDataFileName database file.

2.104.3 List of Data Members and Constants

Member / Const Name	Description
Dim UserDefinedSourceNumber As Integer	This variable is used to keep track of the number of the currently active user defined source.
Type UserDefinedType NodeName As String StartTime As String StopTime As String TimeStepValues As String NumberOfTimeSteps As Integer SourceType As Integer End Type	This is the defining type of the user defined source. It contains all the properties of the user defined source. To use it, an array of user defined sources is required (see next entry).
Dim UserDefinedSources(100) As UserDefinedType	This is the array of user defined sources used to save all the fields of all of the user defined sources in the circuit.
Dim NewElement As Integer	This variable is used to determine whether the presently active element has been previously defined or not.

Volume 3 - Appendix D

ATP FOR WINDOWS



THE DATA CARD UNIT

ACKNOWLEDGEMENTS

I would like to acknowledge that use was made of 'AN INTRODUCTION TO EMTP (ELECTROMAGNETIC TRANSIENTS PROGRAM)' by J. M. Van Coller (1993) for many of the descriptions of elements used in ATP / data members used in ATP for Windows.

DOCUMENT STATUS CONTROL

Configuration Control

Project / Owner:	Tamir Orbach
Title:	<i>ATP for Windows</i>
Doc. Ref. Number:	ATP003
Version Number:	1.0
Version Release Date:	16 June 1995
Author:	Tamir Orbach
Location and Filename:	C:\OFFICE\WPWIN\WPDOCS\ DCU.001

Chapter Status Control

Chapter	Revision	Date
1	1.0	17/06/95

Document Change History

DCR	Date	Chapter	Page	Change Description / Reason	Old Rev	New Rev
ATP for Windows	01/07/95	All	All			0.1
ATP for Windows	17/06/95	All	All	Single Spacing, fixing up errors.	0.1	1.0

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	ii
------------------------	----

DOCUMENT STATUS CONTROL	ix
-------------------------------	----

TABLE OF CONTENTS	v
-------------------------	---

1	THE DATA CARD UNIT	1-1
	1.1 List of Classes	1-1
	1.2 The Data_Case Class	1-16
	1.3 The Data_Card Class	1-17
	1.4 The Power_System_Data_Card Class	1-19
	1.5 The ATP_Control_Card Class	1-21
	1.6 The ATP_Setup_Card Class	1-22
	1.7 The TACS_Element Class	1-26
	1.8 The Branch_Card Class	1-27
	1.9 The Switch_Card Class	1-28
	1.10 The Source_Card Class	1-30
	1.11 The Output_Variable_Request_Card Class	1-32
	1.12 The RLC_Branch Class	1-33
	1.13 The Piecewise_SgArr Class	1-34
	1.14 The Exponential_SgArr Class	1-35
	1.15 The Xformer Class	1-38
	1.16 The Xmission_Line Class	1-40
	1.17 The Fortran_Rel_Element Class	1-46
	1.18 The Controller_Topology_Elem Class	1-48
	1.19 The TACS_Special_Device Class	1-50
	1.20 The TACS_Controlled_Element Class	1-51
	1.21 The Standard_Switch Class	1-52
	1.22 The Open/Close_Switch Class	1-53
	1.23 The Open/Close_Slave_Switch Class	1-54
	1.24 The Systematic_Closing_Switch Class	1-55
	1.25 The Const_Delay_Slave_Switch Class	1-57
	1.26 The TACS_Source Class	1-58
	1.27 The User_Defined_Source Class	1-59
	1.28 The Sinusoidal_Funct_Source Class	1-60
	1.29 The Step_Funct_Source Class	1-62
	1.30 The Ramp_Funct_Source Class	1-62
	1.31 The Dual_Ramp_Funct_Source Class	1-63
	1.32 The Double_Expon_Funct_Source Class	1-65
	1.33 The Simple_RLC Class	1-66
	1.34 The Distributed_RLC Class	1-67

1.35	The Single_Phase_Xformer Class	1-69
1.36	The Shell_Type_Xformer Class	1-69
1.37	The Core_Type_Xformer Class	1-70
1.38	The Nominal_Pi_Xmission_Line Class	1-72
1.39	The Equivalent_Pi_Xmission_Line Class	1-73
1.40	The Modal_Parameter_Xmission_Line Class	1-74
1.41	The MARTI_Xmission_Line Class	1-75
1.42	The Relay_Operated_Switch Class	1-77
1.43	The Level_Triggered_Switch Class	1-78
1.44	The Controlled_Integrator Class	1-80
1.45	The Simple_Differentiator Class	1-81
1.46	The Opening_Switch Class	1-82
1.47	The Closing_Switch Class	1-83
1.48	The Opening_Slave_Switch Class	1-84
1.49	The Closing_Slave_Switch Class	1-85
1.50	The DC_Level Class	1-86
1.51	The TACS_Sinusoid Class	1-87
1.52	The Repetitive_Pulse Class	1-88
1.53	The Repetitive_Ramp Class	1-89
1.54	The TACS_Controlled_Source Class	1-90
1.55	List of Files	1-91

1 THE DATA CARD UNIT

This Unit contains all the methods required to write the ATP data files / data cards from the numerical and graphical information received by the interface.

1.1 List of Classes

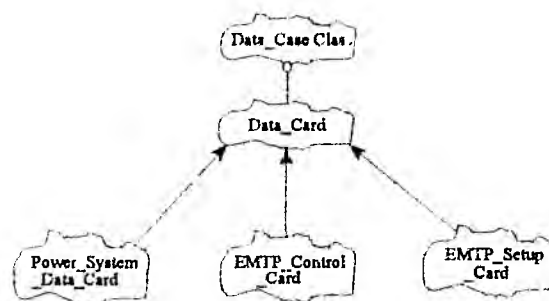


Figure 2 - The Data_Case Class Structure Hierarchy

Class Name	Description
Data_Case	The Data_Case class is a generic class which contains all of the Data_Card classes and their components (see figure 2 above).
Data_Card	The Data_Card class is a generic class which includes methods to write a data card / line in the data case in the correct position and to write characters in a specific line and position in the data case.
Power_System_Data_Card	This class inherits the methods and data members of the Data_Card class. It contains functionality to write all details which are specific to Power System Data cards and are not available in the Data_Card class.
ATP_Control_Card	This class inherits the methods and data members of the Data_Card class. It contains functionality to write all details which are specific to ATP Control cards and are not available in the Data_Card class.

Class Name	Description
ATP_Setup_Card	This class inherits the methods and data members of the Data_Card class. It contains functionality to write all details which are specific to ATP Setup cards and are not available in the Data_Card class.

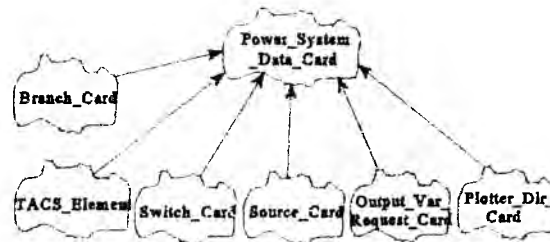


Figure 3- The Power_System_Data_Card Class Structure Hierarchy

Class Name	Description
Branch_Card	This class inherits the methods and data members of the Power_System_Data_Card class. It contains functionality to write all details which are specific to ATP Branch cards and are not available in the Power_System_Data_Card class (see List of Methods).
TACS_Element	This class inherits the methods and data members of the Power_System_Data_Card class. It contains functionality to write all details which are specific to TACS (Transient Analysis of Control Systems) elements (not sources) and are not available in the Power_System_Data_Card class (see List of Methods).
Switch_Card	This class inherits the methods and data members of the Power_System_Data_Card class. It contains functionality to write all details which are specific to ATP Switch cards and are not available in the Power_System_Data_Card class (see List of Methods).

Class Name	Description
Source_Card	This class inherits the methods and data members of the Power_System_Data_Card class. It contains functionality to write all details which are specific to ATP Source cards and are not available in the Power_System_Data_Card class (see List of Methods).
Output_Var_Request_Card	This class inherits the methods and data members of the Power_System_Data_Card class. It contains functionality to write all details which are specific to ATP Output Variable Request cards and are not available in the Power_System_Data_Card class (see List of Methods).

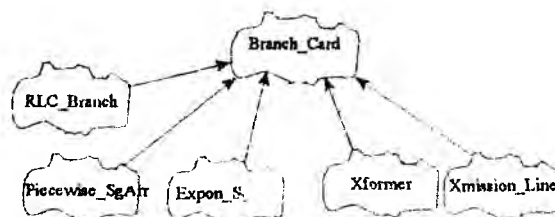


Figure 4- The Branch_Card Class Structure Diagram

Class Name	Description
RLC_Branch	This class inherits the methods and data members of the Branch_Card class. It contains functionality to write all details which are specific to RLC Element Branch cards (both Simple RLC and Distributed RLC) and are not available in the Branch_Card class (see List of Methods).
Piecewise_SgArr	This class inherits the methods and data members of the Branch_Card class and contains functionality to write all details which are specific to Piecewise-linear Model Surge Arrestor cards and are not available in the Branch_Card class (see List of Methods).

Class Name	Description
Exponential_SgArr	This class inherits the methods and data members of the Branch_Card class and contains functionality to write all details which are specific to Exponential Model Surge Arrestor cards and are not available in the Branch_Card class (see List of Methods).
Xformer	This class inherits the methods and data members of the Branch_Card class. It contains functionality to write all details which are specific to Transformer Branch cards and are not available in the Branch_Card class (see List of Methods).
Xmission_Line	This class inherits the methods and data members of the Branch_Card class. It contains functionality to write all details which are specific to Transmission Line Branch cards (including all methods which write details common to the Nominal-Pi model, Equivalent-Pi model, JMARTI model, and Modal Parameters model) and are not available in the Branch_Card class (see List of Methods).

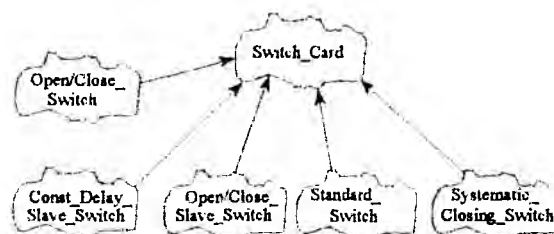


Figure 5 - The Switch_Card Class Heirarchy

Class Name	Description
Standard_Switch	This class inherits the methods and data members of the Switch_Card class. It contains functionality to write all details which are specific to Standard Switch Switch cards and are not available in the Switch_Card class (see List of Methods).

Class Name	Description
Open/Close_Switch	This class inherits the methods and data members of the Switch_Card class. It contains functionality to write all details which are specific to opening and closing switches Switch cards and are not available in the Switch_Card class, and are generic to both the opening and closing switches (see List of Methods).
Open/Close_Slave_Switch	This class inherits the methods and data members of the Switch_Card class. It contains functionality to write all details which are specific to the slave opening and slave closing switches Switch cards and are not available in the Switch_Card class, and are generic to both the slave opening and slave closing switcher (see List of Methods).
Systematic_Closing_Switch	This class inherits the methods and data members of the Switch_Card class. It contains functionality to write all details which are specific to Systematic Closing Switch (systematically increasing the time of switching to compare results) cards and are not available in the Switch_Card class (see List of Methods).

Class Name	Description
Const_Delay_Slave_Switch	This class inherits the methods and data members of the Switch_Card class. It contains functionality to write all details which are specific to Constant Delay Slave Switch cards and are not available in the Switch_Card class (see List of Methods).

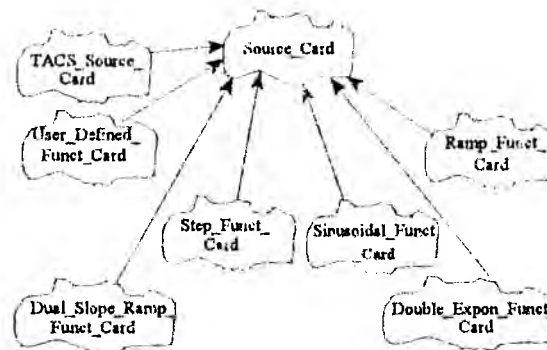


Figure 6- The Source_Card Class Structure
Heirarchy

Class Name	Description
TACS_Source_Card	This class inherits the methods and data members of the Source_Card class. It contains functionality to write all details which are specific to TACS (Transient Analysis of Control Systems) Source cards and are not available in the Source_Card class (see List of Methods).

Class Name	Description
User_Defined_Source_Card	This class inherits the methods and data members of the Source_Card class. It contains functionality to write all details which are specific to User Defined Source cards and are not available in the Source_Card class such as writing the v-t cards following a user defined source card (see List of Methods).
Dual_Slope_Ramp_Funct_Card	This class inherits the methods and data members of the Source_Card class. It contains functionality to write all details which are specific to Dual Slope Ramp Function Source cards and are not available in the Source_Card class (see List of Methods).
Step_Funct_Card	This class inherits the methods and data members of the Source_Card class. It contains functionality to write all details which are specific to Step Function Source cards and are not available in the Source_Card class (see List of Methods).
Sinusoidal_Funct_Card	This class inherits the methods and data members of the Source_Card class. It contains functionality to write all details which are specific to Sinusoidal Function Source cards and are not available in the Source_Card class (see List of Methods).
Ramp_Funct_Card	This class inherits the methods and data members of the Source_Card class. It contains functionality to write all details which are specific to Ramp Function Source cards and are not available in the Source_Card class (see List of Methods).
Double_Expon_Funct_Card	This class inherits the methods and data members of the Source_Card class. It contains functionality to write all details which are specific to Double Exponential Function Source cards and are not available in the Source_Card class (see List of Methods).

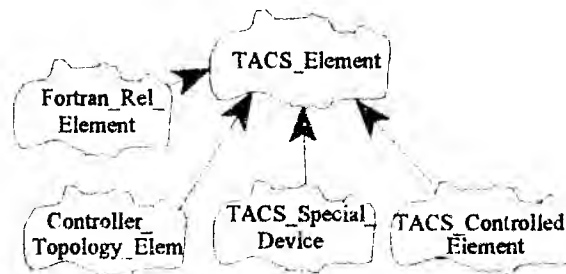
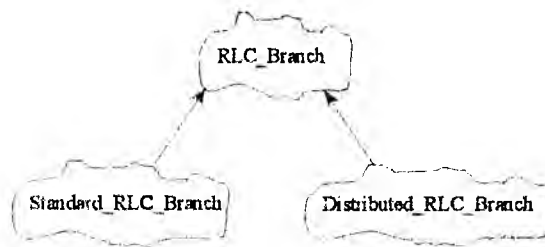
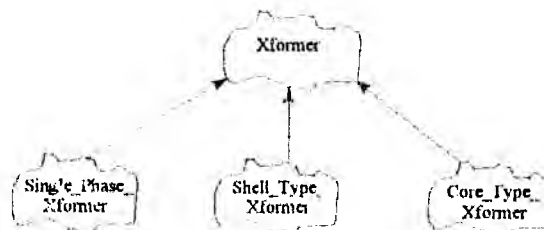


Figure 7 - The TACS_Element Class Heirarchy

Class Name	Description
Fortran_Rel_Element	This class inherits the methods and data members of the TACS_Element class. It contains functionality to write all details which are specific to Fortran Relationship TACS cards and are not available in the TACS_Element class (see List of Methods).
Controller_Topology_Elem	This class inherits the methods and data members of the TACS_Element class. It contains functionality to write all details which are specific to Controller Topology TACS cards and are not available in the TACS_Element class (see List of Methods).
TACS_Special_Device	This class inherits the methods and data members of the TACS_Element class. It contains functionality to write all details which are specific to TACS Special Devices cards and are not available in the TACS_Element class (see List of Methods).
TACS_Controlled_Element	This class inherits the methods and data members of the TACS_Element class. It contains functionality to write all details which are specific to TACS Controlled Element cards and are not available in the TACS_Element class (see List of Methods).

**Figure 8 - The RLC_Branch Class Heirarchy**

Class Name	Description
Simple_RLC_Branch	This class inherits the methods and data members of the RLC_Branch class and contains functionality to write all details which are specific to Simple RLC Branch cards and are not available in the RLC_Branch class (see List of Methods).
Distributed_RLC_Branch	This class inherits the methods and data members of the RLC_Branch class and contains functionality to write all details which are specific to Distributed RLC Branch cards and are not available in the RLC_Branch class (see List of Methods).

**Figure 9- The Xformer Class Structure Diagram**

Class Name	Description
Single_Phase_Xformer	This class inherits the methods and data members of the Xformer class and contains functionality to write all details which are specific to Single Phase Transformer cards and are not available in the Xformer class (see List of Methods).
Shell_Type_Xformer	This class inherits the methods and data members of the Xformer class and contains functionality to write all details which are specific to Shell-Type Transformer cards and are not available in the Xformer class (see List of Methods).
Core_Type_Xformer	This class inherits the methods and data members of the Xformer class and contains functionality to write all details which are specific to Core-Type Transformer cards and are not available in the Xformer class (see List of Methods).

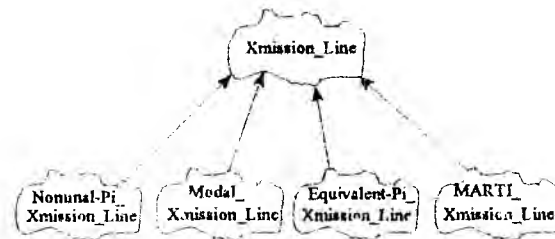


Figure 10- The Xmission_Line Class Structure Diagram

Class Name	Description
Nominal_Pi_Xmission_Line	This class inherits the methods and data members of the Xmission_Line class and contains functionality to write all details which are specific to Nominal-Pi Transmission line model cards and are not available in the Xmission_Line class (see List of Methods).
Modal_Xmission_Line	This class inherits the methods and data members of the Xmission_Line class and contains functionality to write all details which are specific to Modal Parameter Transmission line model card, and are not available in the Xmission_Line class (see List of Methods).
Equivalent_Pi_Xmission_Line	This class inherits the methods and data members of the Xmission_Line class and contains functionality to write all details which are specific to Equivalent-Pi Transmission line model cards and are not available in the Xmission_Line class (see List of Methods).
MARTI_Xmission_Line	This class inherits the methods and data members of the Xmission_Line class and contains functionality to write all details which are specific to JMARTI Transmission line model cards and are not available in the Xmission_Line class (see List of Methods).

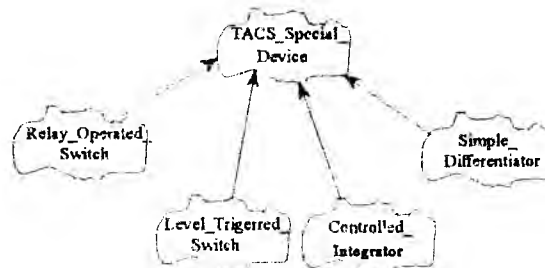


Figure 11 - The TACS_Special_Device Heirarchy

Class Name	Description
Relay_Operated_Switch	This class inherits the methods and data members of the TACS_Special_Device class and contains functionality to write all details which are specific to Relay Operated Switch cards and are not available in the TACS_Special_Device class (see List of Methods).
Level_Triggered_Switch	This class inherits the methods and data members of the TACS_Special_Device class and contains functionality to write all details which are specific to Level Triggered Switch cards and are not available in the TACS_Special_Device class (see List of Methods).
Controlled_Integrator	This class inherits the methods and data members of the TACS_Special_Device class and contains functionality to write all details which are specific to Controlled Integrator cards and are not available in the TACS_Special_Device class (see List of Methods).
Simple_Differentiator	This class inherits the methods and data members of the TACS_Special_Device class and contains functionality to write all details which are specific to Simple Differentiator cards and are not available in the TACS_Special_Device class (see List of Methods).

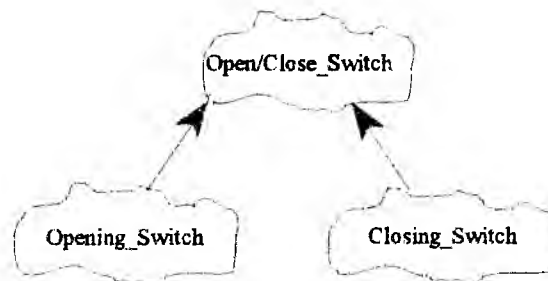


Figure 12- The Open/Close_Switch Class Structure Diagram

Class Name	Description
Opening_Switch	This class inherits the methods and data members of the Open/Close_Switch class and contains functionality to write all details which are specific to Opening Switch cards and are not available in the Open/Close_Switch class (see List of Methods).
Closing_Switch	This class inherits the methods and data members of the Open/Close_Switch class and contains functionality to write all details which are specific to Closing Switch cards and are not available in the Open/Close_Switch class (see List of Methods).

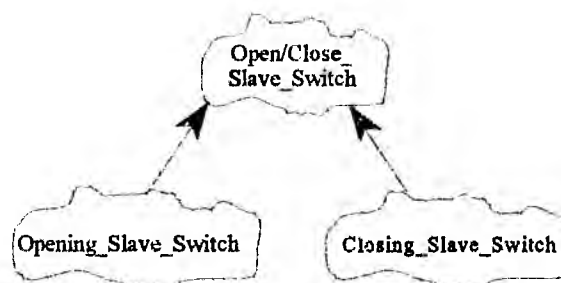


Figure 13- The Open/Close_Slave_Switch Class Structure Diagram

Class Name	Description
Opening_Slave_Switch	This class inherits the methods and data members of the Open/Close_Slave_Switch class and contains functionality to write all details which are specific to Slave Closing Switch cards and are not available in the Open/Close_Slave_Switch class (see List of Methods).
Closing_Slave_Switch	This class inherits the methods and data members of the Open/Close_Slave_Switch class and contains functionality to write all details which are specific to Slave Opening Switch cards and are not available in the Open/Close_Slave_Switch class (see List of Methods).

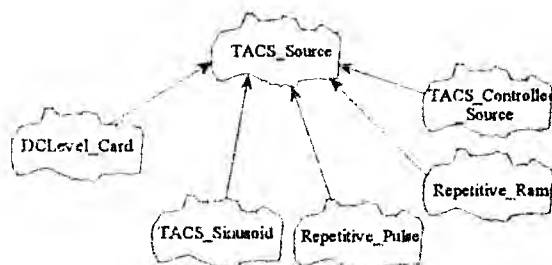


Figure 14 - The TACS_Source Class Hierarchy

Class Name	Description
DCLevel	This class inherits the methods and data members of the TACS_Source class and contains functionality to write all details which are specific to DC Level or Single Pulse cards and are not available in the TACS_Source class (see List of Methods).
TACS_Sinusoid	This class inherits the methods and data members of the TACS_Source class and contains functionality to write all details which are specific to TACS Sinusoidal Source cards and are not available in the TACS_Source class (see List of Methods).

Class Name	Description
Repetitive_Pulse	This class inherits the methods and data members of the TACS_Source class and contains functionality to write all details which are specific to Repetitive Pulse cards and are not available in the TACS_Source class (see List of Methods).
Repetitive_Ramp	This class inherits the methods and data members of the TACS_Source class and contains functionality to write all details which are specific to Repetitive Ramp cards and are not available in the TACS_Source class (see List of Methods).
TACS_Controlled_Source	This class inherits the methods and data members of the TACS_Source class and contains functionality to write all details which are specific to TACS Controlled Source cards and are not available in the TACS_Source class (see List of Methods).

1.2 The Data_Case Class

1.2.1 Class Description

The Data_Case class contains methods which create a data case (open a text file) and write the first data case lines such as BEGIN DATA CASE. This class contains the Data card classes of the entire simulation.

1.2.2 List of Methods

Method Name	Description
void Data_Case(char[30] *SimName)	Default Constructor - opens a text file, and writes the lines BEGIN NEW DATA CASE, and BLANK Card Ending Session at the relevant lines in the file. Also writes a comment card describing the simulation name.
void Data_Case(char[30] *SimName, char[20] *SimType)	Additional constructor - calls the standard constructor, and opens an EMTP file which is an extra file added to the standard simulation (such as LINE CONSTANTS).
Data_Case()	deletes all the variables passed to the Data_Case class.
void WriteFormattedOpen(char[12] filename)	This method writes the line '\$OPEN, UNIT = 7 FILE = dos_filename.PCH FORM=FORMATTED'
void WriteUnformattedOpen(char[12] filename)	This method writes the line '\$OPEN, UNIT = 4 FILE = dos_filename.P14 FORM=UNFORMATTED'
void AddLine()	Adds a line to the data case (initially starts with 1).

1.2.3 List of Data Members

Member Name	Description
char[30] SimulationName	The name of the simulation
char[20] SimulationType	The type of simulation being performed
int NumberOfLines	A counter for the number of lines in the data case.

1.2.4 Comments

See the class structure diagram for the Data_Case class (figure 2) for more details.

1.2.5 Revision History

First Revision: 01/02/1995

1.3 The Data_Card Class

1.3.1 Class Description

The Data_Card class contains generic methods to write information to the data file to the correct line, and the correct location on a line. These functions are used by every card in the data case.

1.3.2 List of Methods

Method Name	Description
void Data_Card()	Constructor
~Data_Card()	Destructor
void AddLine()	Calls the Data_Case::AddLine method to add a line to the simulation.
void WriteString (char[20] TheString, int CharPosition, int LineNumber)	Writes a string which is usually a maximum of 6 characters in length (ATP restriction), except in the case of ATP keywords such as SYSTEMATIC, in the correct position of the present line.
int GiveLineNumber()	Gives an outside class the position / line number of this line in the data case.

1.3.3 List of Data Members

Member Name	Description
WriteString: char[20] TheString	The string to be written in the data_card / line, which has been passed to the WriteString method.
WriteString: int CharacterPosition	The position of the character passed to the WriteString method.
int LineNumber	The line number of the data card used in both the GiveLineNumber, and the WriteString methods.

1.3.4 Comments

See the class structure diagram for the Data_Card class (figure 3) for more details.

1.3.5 Revision History

First Revision: 01/02/95

1.4 The Power_System_Data_Card Class

1.4.1 Class Description

This class inherits the methods and data members of the Data_Card class. It contains functionality to write all details which are specific to Power System Data cards and are not available in the Data_Card class. These methods include methods to write the left node name, right node name, left node2 name, and right node2 name common to almost all power system data cards.

1.4.2 List of Methods

Method Name	Description
void Power_System_Data_Card()	Constructor
~Power_System_Data_Card()	Destructor
void WriteLeftNodeName(char *Node1)	This method is used to write the left / top node names to the correct position in the data card.
void WriteRightNodeName(char *Node2)	This method is used to write the right / bottom node names to the correct position in the data card.
void WriteNode3Name(char *Node3)	This method is used to write the left / top node names of an element which shares all of the properties of the present element, if such an element exists, to the correct position in the data card.
void WriteNode4Name(char *Node4)	This method is used to write the right / bottom node names of an element which shares all of the properties of the present element, if such an element exists, to the correct position in the data card.
int IsSimilarElementUsed()	This method can be called from any other class to check whether or not the present card being written uses the properties of a similar element.

1.4.3 List of Data Members

Member Name	Description
char[6] Node1Name	The name of the left / top node name of the element.
char[6] Node2Name	The name of the right / bottom node name of the element.
char[6] Node3Name	The name of the left / top node name of the similar element.
char[6] Node4Name	The name of the right / bottom node name of the similar element.
int SimilarElementUsed	A boolean variable which is set to true if the Node3Name and Node4Name data members are not "" (ie. are not empty).

1.4.4 Comments

See the class structure diagram for the Power_System_Data_Card class (figure 4) for more details.

1.4.5 Revision History

First Revision: 01/02/95

1.5 The ATP_Control_Card Class

1.5.1 Class Description

The ATP_Control_Card class inherits the methods and data members of the Data_Card class. It contains functionality to write all details which are specific to ATP Control cards and are not available in the Data_Card class. This functionality includes writing each control card available (control cards are Fortran strings which make up an entire line / card such as '\$INPUT').

1.5.2 List of Methods

Method Name	Description
void ATP_Control_Card()	Constructor
~ATP_Control_Card()	Destructor
void WriteCommentCard (char[15] Comment)	This method is used to write whatever comment is needed.

1.5.3 List of Data Members

Member Name	Description
WriteCommentCard: char30] Comment	The variable which contains the actual comment to be written.

1.5.4 Comments

None

1.5.5 Revision History

First Revision: 01/02/95

1.6 The ATP_Setup_Card Class

1.6.1 Class Description

The ATP_Setup_Card class inherits the methods and data members of the Data_Card class. It contains functionality to write all details which are specific to ATP Control cards and are not available in the Data_Card class. This functionality includes writing all of the variables in each of the two miscellaneous cards available (The Floating Point Miscellaneous Card, and the Integer Miscellaneous Card), and special request cards.

1.6.2 List of Methods

Method Name	Description
void Setup_Card()	Constructor
~Setup_Card()	Destructor
void WriteBeginNewDataCase()	This method is called by the Data_Case class to write the line 'BEGIN NEW DATA CASE'
void WriteChangePrintoutFrequency()	This method is called by the Data_Case class to write the line 'CHANGE PRINTOUT FREQUENCY'
void WriteBlankCardString (char[15] Section)	This method is used to write the line 'BLANK Card ending section'.
void WritePrintoutFrequency (char[10] KCHG, char[10] MULT)	This method is used to write additional printout frequency cards if these are required.
void WriteFloatingPointMiscellaneous (char[10] Delta, char[10] Tmax, char[10] Xopt, char[10] Copt)	This method is used to write the entire Floating Point Miscellaneous card. It calls the WriteString method for each string.
void WriteIntegerMiscellaneous (char[10] Iout, char[10] Iplot, int Idoubl, int Kssout, int Maxout, int Ipun, int Memsav, int NENERG)	This method is used to write the entire Integer Miscellaneous card. It calls the WriteString method for each string.

1.6.3 List of Data Members

Member Name	Description
WriteBeginNewDataCard: char[15] BeginString	The 'BEGIN DATA CASE' string
WriteBlankCardString: char[30] BlankString	The string used to write 'BLANK card ending'
WriteBlankCardString: char[20] Section	The string containing the name of the section to 'end' with a blank card
WriteChangePrintoutFrequency: char[24] ChangePntFreqString	The string used to write 'CHANGE PRINTOUT FREQUENCY'
WritePrintoutFrequency: char[10] KCHG	The integer specifying the time step number at which the kth frequency of output is to begin (used only on request of additional printout frequency cards in the Setup Simulation screen).
WritePrintoutFrequency: char[10] MULT	Specifying the new frequency of output (see PRINTOUT - see Integer Miscellaneous card) (used only on request of additional printout frequency cards in the Setup Simulation screen).
WriteFloatingPointMiscellaneous: char[10] Deltat	The simulation time step in floating point format, in seconds.
WriteFloatingPointMiscellaneous: char [10] Tmax	The required duration of simulation in floating point format, in seconds.
WriteFloatingPointMiscellaneous: char[10] Xopt	This is a flag for indicating the form of inductance input - whether inductance in linear branches is to be specified in mH (Xopt is set to 0.0), or Ω (if Xopt is set to the value of the frequency of the simulation).

Member Name	Description
WriteFloatingPointMiscellaneous: char[10] Copt	This is a flag for indicating the form of capacitance input - whether capacitance in linear branches is to be specified in μF (Copt is set to 0.0), or in $\mu mhos$ - capacitive susceptance (if Copt is set to the value of the frequency of the simulation).
WriteIntegerMiscellaneous: char[10] Iout	This is the frequency of output relative to the simulation time step (a 3 will request that the values of every 3rd time step be printed).
WriteIntegerMiscellaneous: char[10] Iplot	This is the frequency of saving simulation values for later plotting (not really relevant for ATP for Windows, but still available as an option).
WriteIntegerMiscellaneous: int Idoubl	This is a flag for controlling the output of a table showing network connectivity. A 0 will suppress the output whereas a 1 will produce it.
WriteIntegerMiscellaneous: int Kssout	This is a flag for controlling the printout of the steady-state phasor solution. The options are: 0 - no steady state printout 1 - print the complete steady state solution 2 - print switch flows and source injections, but not branch flows 3 - print branch flows requested in column 80 (variable P - see the Branch Card class) on the branch cards, switch flows and source injections.
WriteIntegerMiscellaneous: int Maxout	This is a flag for controlling the printout of extrema at the completion of the simulation. A 0 will suppress such output whereas a 1 will produce it.

Member Name	Description
WriteIntegerMiscellaneous: int Ipun	The variable IPUN is used to flag the presence of an extra, following card to manipulate the printout frequency. A -1 indicates an extra card while a 0 indicates none. (columns 47-48 -> right justify)
WriteIntegerMiscellaneous: int Memsav	The MEMSAV variable is a flag for controlling the dumping of EMTP memory onto disk at the end of the simulation for subsequent use with the START AGAIN request. Enter a 1 if such a memory saving is desired, and a 0 if it is not. (Column 56).
WriteIntegerMiscellaneous: int Nenerg	The variable NENERG is the number of iteration runs required for Monte Carlo analysis of the Slave opening switch. Usually 0, ie. usually ignored.

1.6.4 Comments

None

1.6.5 Revision History

First Revision: 01/02/95

1.7 The TACS_Element Class

1.7.1 Class Description

This class inherits the methods and data members of the Power_System_Data_Card class. It contains functionality to write all details which are specific to TACS (Transient Analysis of Control Systems) elements (not sources) and are not available in the Power_System_Data_Card class (see List of Methods). This functionality includes writing the comments required for the use of all TACS elements.

1.7.2 List Of Methods

Method Name	Description
void TACS_Element ()	Constructor
~TACS_Element ()	Destructor
Write TACSInitStrings ()	This method is used to write the TACS strings required to indicate the beginning of a TACS Element definition.

1.7.3 List of Data Members

Member Name	Description
char[15] TACSStandAloneString	This is a variable which contains the 'TACS STAND ALONE' string.
char[10] TACSHybridString	This is a variable which contains the 'TACS HYBRID' string.

1.7.4 Comments

See the class structure diagram for the TACS_Element class (figure

1.7.5 Revision History

First Revision: 01/02/95

1.8 The Branch_Card Class

1.8.1 Class Description

This class inherits the methods and data members of the Power_System_Data_Card class. It contains functionality to write all details which are specific to ATP Branch cards and are not available in the Power_System_Data_Card class (see List of Methods). This functionality includes writing the variables

1.8.2 List of Methods

Method Name	Description
void Branch_Card()	Constructor
~Branch_Card()	Destructor
void DeleteElement()	This method is called to delete a branch card and all of it's properties.
void WriteItype(int Itype)	This method is used to write the Itype variable which every branch card needs to identify it.
void WriteP(int P)	This method is used to write the variable P (Column 80) to the data card. The variable P is a flag for specifying what outputs are required from this branch. See User Manual for what each number stands for.

1.8.3 List of Data Members

Member Name	Description
WriteItype: int Itype	This member is describes the type of element this particular card describes. Every type of element has a two digit Itype code which identifies it.
int P	See above

1.8.4 Comments

See the class structure diagram for the Branch_Card class (figure 4)

1.8.5 Revision History

First Revision: 01/02/95

1.9 The Switch_Card Class

1.9.1 Class Description

This class inherits the methods and data members of the Power_System_Data_Card class. It contains functionality to write all details which are specific to ATP Switch cards and are not available in the Power_System_Data_Card class (see List of Methods). This functionality includes writing the Itype or description code of the switch card element.

1.9.2 List of Methods

Method Name	Description
void Switch_Card()	Constructor
~Switch_Card()	Destructor
void DeleteElement()	This method is used to delete a source and all of its properties.
void WriteIType()	This method is used to write the ITYPE (00) in the switch card since this is always the same value for switch card.
void WriteStatisticsMiscellaneousCard (int ISW, int ITest, int IDist, int Imax, int Kssout, int Nseed)	This method is called by the Standard_Switch class, Open/Close_Switch class, and the Open/Close_Slave_Switch class to set a number of parameters including the listing of switch operating times, controlling the global offset times for random switches, controlling the type of distribution for the random switch operating times, controlling the listing of deterministic extrema for each energisation etc.

Method Name	Description
void WriteSystematicMiscellaneousCard (int ISW, int Itest, int Imax, int Kssout)	This method is called by the Systematic Closing Switch class as well as the Const Delay Slave Switch class in order to enter parameters relating to the systematic closing of those switches. These parameters include a flag for controlling the listing of switch operating times, a flag for controlling whether the systematic switch times are to be median or minimum times, a flag for controlling the listing of deterministic extrema, and a flag for controlling the printout of additional time step data for each energisation.

1.9.3 List of Data Members

Member Name	Description
WriteIType: int IType	This member describes the type of element this particular card describes. Every type of element has a two digit IType code which identifies it.
WriteStatisticsMiscellaneousCard: int IDist	This is a flag for controlling the type of distribution for the random switch operating times. The options are Gaussian (0), and uniform (1).
WriteStatisticsMiscellaneousCard: int Nseed	This is a flag for controlling the seed of the random number generator. The options are no control - non repeatable (0), and same seed - repeatable (1).
int ISW	This is a flag for controlling the listing of switch operating times. The options are: no listing of switch operating times (0), and listing of switch operating times (1).
int Itest	This is a flag for controlling the global offset times for random switches. The options are: 0 -> include global offset time for all switches 1 -> no global offset time 2 -> include global offset time, but only for closing switches 3 -> include global offset time, but only for opening switches.

Member Name	Description
int Imax	This is a flag for controlling the listing of deterministic extrema for each energisation. The options are no listing (0), and listing (1).
int Kssout	This is a flag for controlling the printout of additional time step data for each energisation. The options are no additional listing (-1 or blank), and additional listing (0).

1.9.4 Comments

See the class structure diagram for the Switch_Card class (figure 5)

1.9.5 Revision History

First Revision: 01/02/95

1.10 The Source_Card Class

1.10.1 Class Description

This class inherits the methods and data members of the Power_System_Data_Card class. It contains functionality to write all details which are specific to ATP Switch cards and are not available in the Power_System_Data_Card class (see List of Methods). This functionality includes writing the Itype description variable, the IV variable (which determines whether the source is a voltage or current source), and the start and stop times.

1.10.2 List of Methods

Method Name	Description
void Source_Card()	Constructor
~Source_Card()	Destructor
void DeleteElement()	This method is used to delete a source and all of its properties.

Method Name	Description
void WriteSourceCard (int Itype, int IV, char[10] TStart, char[10] TStop, char[10] AMPL)	This method is used to write the ITYPE variable in the source card, the flag which controls whether the source is a voltage or current source, the start and stop times of the source, and the amplitude of the source if relevant.

1.10.3 List of Data Members

Member Name	Description
WriteSourceCard: int IType	This member describes the type of element this particular card describes. Every type of element has a two digit Itype code which identifies it.
WriteSourceCard: int IV	This member determines whether the function is a voltage source, or a current source. A 0 will produce a voltage source, whereas a -1 will produce a current source.
WriteSourceCard: char[10] AMPL	This member is the amplitude of the function in floating point format.
WriteSourceCard: char[10] TStart	This member is the starting time of the function in floating point format.
WriteSourceCard: char[10] TStop	This member is the stopping time of the function in floating point format.

1.10.4 Comments

See the class structure diagram for the Source_Card class (figure 6)

1.10.5 Revision History

First Revision: 01/02/95

1.11 The Output_Variable_Request_Card Class

1.11.1 Class Description

This class inherits the methods and data members of the Power_System_Data_Card class. It contains functionality to write all details which are specific to ATP Switch cards and are not available in the Power_System_Data_Card class (see List of Methods). This functionality includes writing the variable names for all of the variables whose output is required.

1.11.2 List of Methods

Method Name	Description
void Output_Variable_Request_Card()	Constructor
~Output_Variable_Request_Card()	Destructor
void WriteVariableName (char[6] VariableName)	This method writes each variable in the Output variable request card ie. each variable for which an output is requested.
void WritePlotAllVariables()	This method writes a 1 in column 2 of the Output Variable Request card indicating that the voltage of all nodes is required.

1.11.3 List of Data Members

Member Name	Description
WriteVariableName: char[6] VariableName	This is the variable name of the node for which output is required.

1.11.4 Comments

None

1.11.5 Revision History

First Revision: 01/02/95

1.12 The RLC_Branch Class

1.12.1 Class Description

This class inherits the methods and data members of the RLC_Branch class and contains functionality to write all details which are specific to Simple RLC Branch cards and are not available in the RLC_Branch class (see List of Methods). This functionality includes writing the lumped or distributed resistance, capacitance, and inductance.

1.12.2 List of Methods

Method Name	Description
void RLC_Branch ()	Constructor
~RLC_Branch ()	Destructor
void WriteRLC (char[10] Resistance, char[10] Inductance, char[10] Capacitance)	This method is used to write the variables R, L, and C which make up the lumped or distributed resistance, inductance, and capacitance.

1.12.3 List of Data Members

Member Name	Description
WriteRLC: char[10] Resistance	The resistance of the RLC branch.
WriteRLC: char[10] Inductance	The inductance of the RLC branch.
WriteRLC: char[10] Capacitance	The capacitance of the RLC branch.

1.12.4 Comments

See the class structure diagram for the RLC_Branch class (figure 7)

1.12.5 Revision History

First Revision: 01/02/95

1.13 The Piecewise_SgArr Class

1.13.1 Class Description

This class inherits the methods and data members of the Branch_Card class and contains functionality to write all details which are specific to Piecewise-linear Model Surge Arrestor cards and are not available in the Branch_Card class (see List of Methods).

1.13.2 List of Methods

Method Name	Description
void Piecewise_SgArr ()	Constructor
~Piecewise_SgArr ()	Destructor
void WriteSgArrDetails (char[10] Vflash, char[10] Tdly, char[10] JUMP, char[10] VSeal)	This method is used to write the variables Vflash, Tdly, Jump, and Vseal, which make up the parameters of the piecewise linear modelled surge arrestor which are not written in the Branch_Card class.
void WriteVCharacteristics (char[10] V, char[10] I)	This method is used to write the set of cards specifying the piecewise-linear v-I characteristic.

1.13.3 List of Data Members

Member Name	Description
WriteSgArrDetails: char[10] VFlash	The branch voltage at which breakdown of the spark gap occurs in floating point format, in volts.
WriteSgArrDetails: char[10] Tdly	The earliest time after flashover that the arc can extinguish at a current zero, in floating point format, in seconds.

Member Name	Description
WriteSgArrDetails: char[10] Jump	The specific segment of the piecewise-linear v-I characteristic at which conduction begins after flashover in floating point format.
WriteSgArrDetails: char[10] VSeal	the threshold value of branch voltage for arc extinction, in floating point format, in volts. If no threshold value exists then leave blank.
WriteVCharacteristics: char[10] V	The kth inflection point on the piecewise-linear characteristic in floating point format, in volts.
WriteVCharacteristics: char[10] I	The kth inflection point on the piecewise-linear characteristic in floating point format, in amps.

1.13.4 Comments

None

1.13.5 Revision History

First Revision: 01/02/95

1.14 The Exponential_SgArr Class

1.14.1 Class Description

This class inherits the methods and data members of the Branch_Card class and contains functionality to write all details which are specific to Exponential Model Surge Arrestor cards and are not available in the Branch_Card class (see List of Methods). This functionality includes writing the ZnO identifier IDEN, the Vref and Vflash variables and the exponential segment cards defining the exponential function (see User Reference Manual explanation of exponential modelled surge arrestor).

1.14.2 List of Methods

Method Name	Description
void Exponential_SgArr ()	Constructor

Method Name	Description
~Exponential_SgArr ()	Destructor
void WriteSgArrDetails (char[25] VRef, char[25] VFlash, char[25] MULT, char[25] EXPON, char[25] VMIN)	This method is used to write the variables Iden (5555 in the case of the exponential surge arrester), VRef, VFlash, MULT, EXPON, and VMIN, which make up the parameters of the exponential modelled surge arrester which are not written in the Branch_Card class as well as the parameters of the exponential function described by the equation for the exponential surge arrester (See User Reference Manual - Chapter 7 - Element Details - Exponential Model Surge Arrester).

1.14.3 List of Data Members

Member Name	Description
WriteSgArrDetails: char[3] Iden	The string variable which contains the iden string '5555' (the ZnO identifier).
WriteSgArrDetails: char[25] VRef	The value of Vref in the defining function, in floating point format, in volts. The defining function is: $i = MULT \left[\frac{V}{VREF} \right]^{EXPON}$

Member Name	Description
WriteSgArrDetails: char[25] VFlash	The normalized (with respect to Vref) voltage at which breakdown in the spark gap occurs, in floating point format. For gapless surge arrestor, enter any negative number such as -1.0)
WriteSgArrDetails: char[25] MULT	The value of MULT in the defining equation, in floating point format (See VRef).
WriteSgArrDetails: char[25] EXPON	The value of EXPON in the defining equation, in floating point format (See Vref).
WriteSgArrDetails char[25] VMIN	The normalized (with respect to VRef) voltage above which the above values of MULT and EXPON are applicable (until the next value of VMIN if any).
WriteSgArrDetails char[3] SegmentTerminationString	The string '9999' needed to terminate the exponential surge arrestor cards.

1.14.4 Comments

None

1.14.5 Revision History

First Revision: 01/02/95

1.15 The Xformer Class

1.15.1 Class Description

This class inherits the methods and data members of the Branch_Card class. It contains functionality to write all details which are specific to Transformer Branch cards and are not available in the Branch_Card class (see List of Methods). This functionality includes writing the entire parameter list or data card collection of a single phase transformer, but not the preceding data card for a three phase transformer.

1.15.2 List of Methods

Method Name	Description
void Xformer ()	Constructor
~Xformer ()	Destructor
void WriteTransformerCard (char[10] Ref, char[10] Istdy, char[10] ψ Stdy, char[10] BUSTOP, char[10] RMAG)	This method is used to write the first in the collection of data cards relating to a transformer. This card includes details of the current and flux components to be used in the steady state initialisation as well as the core loss, and a unique name for the transformer.
void WriteCurrFluxCard (char[16] Curr, char[16] Flux)	This method is used to write the flux cards which must follow the transformer card (previous method) describing the piecewise-linear flux versus magnetising current characteristic. The curve must be monotonically increasing. If no flux cards are inserted then the magnetising current is assumed to be zero. This method also write the termination card for these flux cards.
void WriteWindingCard (char[10] Rk, char[10] Lk, char[10] Nk)	This method is used to write the winding cards which follow the previous two cards described. Each winding must have a number which is written to the ITYPE variable (recommended to start at 01). The winding resistance, leakage inductance, and number of turns are also written in this card.

1.15 The Xformer Class

1.15.1 Class Description

This class inherits the methods and data members of the Branch_Card class. It contains functionality to write all details which are specific to Transformer Branch cards and are not available in the Branch_Card class (see List of Methods). This functionality includes writing the entire parameter list or data card collection of a single phase transformer, but not the preceding data card for a three phase transformer.

1.15.2 List of Methods

Method Name	Description
void Xformer ()	Constructor
~Xformer ()	Destructor
void WriteTransformerCard (char[10] Ref, char[10] Istdy, char[10] ψ Stdy, char[10] BUSTOP, char[10] RMAG)	This method is used to write the first in the collection of data cards relating to a transformer. This card includes details of the current and flux components to be used in the steady state initialisation as well as the core loss, and a unique name for the transformer.
void WriteCurrFluxCard (char[16] Curr, char[16] Flux)	This method is used to write the flux cards which must follow the transformer card (previous method) describing the piecewise-linear flux versus magnetising current characteristic. The curve must be monotonically increasing. If no flux cards are inserted then the magnetising current is assumed to be zero. This method also write the termination card for these flux cards.
void WriteWindingCard (char[10] Rk, char[10] Lk, char[10] Nk)	This method is used to write the winding cards which follow the previous two cards described. Each winding must have a number which is written to the ITYPE variable (recommended to start at 01). The winding resistance, leakage inductance, and number of turns are also written in this card.

1.15.3 List of Data Members

Member Name	Description
WriteTransformerCard: char[10] Ref	The reference name of an identical transformer (name entered in the BUSTOP field of that transformer's first card), whose parameters are to be duplicated for this transformer (if such a transformer exists - else leave blank).
WriteTransformerCard: char[10] Istdy	The current component of the point on the linear characteristic of the flux versus magnetising current characteristic to be used in the steady state initialisation in floating point format, in amperes.
WriteTransformerCard: char[10] ψ Stdy	flux component of the point on the linear characteristic of the flux versus magnetising current characteristic to be used in the steady state initialisation in floating point format, in webers.
WriteTransformerCard: char[10] BUSTOP	A unique name given to this transformer which can then be entered in the Ref fields of another transformer to indicate that they have similar parameters.
WriteTransformerCard: char[10] RMAG	The Core Loss expressed as a linear resistance in parallel with the magnetisation inductance in floating point format, in Ω . The default value is ∞ .
WriteCurrFluxCard: char[16] Curr	The current associated with the respective breakpoint of the flux versus magnetising current characteristics in floating point format, in amperes.
WriteCurrFluxCard: char[16] Flux	The flux associated with the respective breakpoint of the flux versus magnetisation current characteristic in floating point format, in webers.
WriteWindingCard: char[10] Rk	The winding resistance in floating point format, in Ω .
WriteWindingCard: char[10] Lk	The winding leakage inductance in floating point format, in mH ($XOPT = 0.0$) or Ω (if $XOPT =$ frequency at which the inductive reactance is to be specified). Only L1 can be zero.

Member Name	Description
WriteWindingCard: char[10] Nk	The number of turns of the winding specified in floating point format. Since only ratios are important, the rated voltage for each winding can be specified instead of the number of turns.

1.15.4 Comments

None

1.15.5 Revision History

First Revision: 01/02/95

1.16 The Xmission_Line Class

1.16.1 Class Description

This class inherits the methods and data members of the Branch_Card class. It contains functionality to write all details which are specific to Transmission Line Branch cards (including all methods which write details common to the Nominal-Pi model, Equivalent-Pi model, and Modal Parameters model) and are not available in the Branch_Card class (see List of Methods). This functionality includes running the Line Constant add on program to ATP (The JMARTI add on is run from the MARTI_Xmission_Line Class).

1.16.2 List of Methods

Method Name	Description
void Xmission_Line ()	Constructor
~Xmission_Line ()	Destructor
void WriteInitLineConstants ()	This method is used to write the initial commands which invoke Line Constants (add on program) including the strings 'LINE CONSTANTS', and invoking the 'BEGIN NEW DATA CASE' string in the Data_Case class.

Method Name	Description
void WriteLineConstantsBranch (char[6] Node1, char[6] Node 2, char[6] Node3, char[6] Node4, char[6] Node 5, char[6] Node6, char[6] Node7, char[6] Node 8, char[6] Node9, char[6] Node10, char[6] Node 11, char[6] Node12, int MetricInclude, int FrequencyInclude)	This method writes the Branch card to the Line Constants file. This card involves naming the left and right nodes of all of the overhead transmission line conductors grouped in pairs corresponding to each phase conductor. This method also writes the 'METRIC' string to convert the units used to units/km if required, and the 'FREQUENCY' string if this has been requested.
void WriteConductorCard (int lphase, char[10] Skin, char[10] Resis, char[10] Diam, char[10] Horiz, char[10] Vtower, char[10] Vmid, char[10] Separ, char[10] Alpha, char[10] Nbund)	This method writes the parameters of the conductor cards. It is called for each of the three conductors in the transmission line, and includes details such as the phase number of each conductor, the skin ratio, the DC resistance, the outside diameter of the conductor, etc.
void WriteFrequencyCard (char[10] RHO, char[10] Freq, char[5] ICPR, char[5] IZPR, int ICAP, char[10] Dist, int IPIPR, int ISEG, int Mutual, int Idec, int Ipnt, int Ipun, int Modal, int Tr)	This method write the Frequency card to the Line Constants file. This includes the ground resistivity, the frequency at which the LINE CONSTANTS program should be run etc.

1.16.3 List of Data Members

Member Name	Description
WriteLineConstantsBranch: char[6] Node1, Node2, Node3 Node4, Node5, Node6 Node7, Node8, Node9, Node10, Node11, Node12	These are the left and right node names of the conductors on the overhead transmission line. Usually only the first 6 names are used.
WriteLineConstantsBranch: int MetricInclude	This is a binary flag which is True if the 'METRIC' keyword is needed (to override the english units default - feet, miles etc).

Member Name	Description
WriteLineConstantsBranch: int WriteFrequency	This is a binary flag which is True if the 'FREQUENCY' keyword is needed (requests a frequency analysis of the line parameters - if this is True, the associated frequency cards must be included immediately following the Conductor Cards.
WriteConductorCard: int Iphase	The Phase Number in Integer format of the conductor
WriteConductorCard: char[10] Skin	The skin ratio: ratio of the thickness of a tubular conductor to it's outer diameter in real number format - use 0.5 for solid conductor.
WriteConductorCard: char[10] Resis	The DC resistance of conductor in real number format, in Ω/km .
WriteConductorCard: char[10] Diam	The outside diameter of tubular conductor in real number format, in cm.
WriteConductorCard: char[10] Horiz	The horizontal separation of the centre of the conductor from some arbitrary reference, in real number format, in metres.
WriteConductorCard: char[10] Vtower	The height of the conductor at the tower, in real number format, in metres.
WriteConductorCard: char[10] Vmid	The height of the conductor at midspan, in real number format, in metres.
WriteConductorCard: char[10] Separ	The separation between adjacent subconductors in real number format, in cm.
WriteConductorCard: char[10] Alpha	The angular position of first subconductor (any subconductor) from horizontal, in real number format, in degrees. Positive angles are measured counter-clockwise.
WriteConductorCard: char[10] Nbund	The number of subconductors in the bundle in integer format.
WriteFrequencyCard: char[10] RHO	The ground resistivity in real number format, in Ωm
WriteFrequencyCard: char[10] Freq	The frequency of the LINE CONSTANTS calculation in real number format, in Hz. For Logarithmic frequency sweeping this is the starting frequency.

Member Name	Description
WriteFrequencyCard: char[5] ICPR	This is a flag for requesting respective distributed shunt capacitance or susceptance matrices as output (also see ICAP) It is a char variable because the value of this variable determines which column to put a 1 in. The columns are as follows: 30 -> $[C]^{-1}$ or $[\omega C]^{-1}$ 31 -> $[C_E]^{-1}$ or $[\omega C_E]^{-1}$ 32 -> $[C_S]^{-1}$ or $[\omega C_S]^{-1}$ 33 -> $[C]$ or $[\omega C]$ 34 -> $[C_E]$ or $[\omega C_E]$ 32 -> $[C_S]$ or $[\omega C_S]$ The subscripts signify: none -> unreduced system - each physical conductor has a row or column matrix, as does each ground wire. E -> equivalent phase conductor parameters (after elimination of ground wires and the bundling of phase conductors). S -> symmetrical component parameters of the equivalent conductors.

Member Name	Description
WriteFrequencyCard: char[5] IZPR	This is a flag for requesting respective distributed series impedance matrices as output. It is a 'char' type variable because the value of this variable determines which column to put a 1 in. The columns are as follows: 37 -> $[Z]$ 38 -> $[Z_t]$ 39 -> $[Z_s]$ 40 -> $[Z]^{-1}$ 41 -> $[Z_t]^{-1}$ 42 -> $[Z_s]^{-1}$ The subscripts are as before.
WriteFrequencyCard: int IC	This is a flag for whether $[C]$ or $[\omega C]$ is required (for the ICPR request). A 1 indicates $[C]$, while a 0 indicates $[\omega C]$.
WriteFrequencyCard: char[10] Dist	The length of the transmission line in real number format, in km.
WriteFrequencyCard: int IPIPR	This is a flag for requesting EQUIVALENT-PI model matrices as output (either in admittance form $[Y]$ or in impedance form $[Z]$) - a 1 is placed in the column indicated by IPIPR. The options are: 54 -> $[Y]$ 55 -> $[Y_s]$ 56 -> $[Z]$ 57 -> $[Z_s]$

Member Name	Description
WriteFrequencyCard: int ISEG	This is a flag for specifying continuity of ground wires. The options are 0 for continuous ground wires (normal situation), and 1 for segmented ground wires (cut at short intervals).
WriteFrequencyCard: int Mutual	This is a flag for indicating the presence of a communication circuit parallel to the power circuit. A 0 indicates no communication circuit while a 1 indicates the presence.
WriteFrequencyCard: int Idec	This is left blank for the normal single frequency line. For logarithmic frequency sweeping, Idec is the number of decades to be swept, in integer format.
WriteFrequencyCard: int Ipnt	This is left blank for the normal single frequency line. For logarithmic frequency sweeping, Ipnt is the number of points per decade for which line constants are to be calculated, in integer format.
WriteFrequencyCard: int Ipuw	This is a flag to indicate whether or not the Nominal-Pi model is required. A 1 (True) indicates that it is, and will write the string '44' in columns 66-68.
WriteFrequencyCard: int Modal	This is a flag to indicate whether the transmission line is continuously transposed or untransposed. A 0 indicates that the line is continuously transposed while a 1 indicated that it is untransposed.

Member Name	Description
WriteFrequencyCard: int Tr	This is a flag for requesting either the full complex, or real part only, transformation matrix for the modal analysis of untransposed lines: 0 -> real part only - used generally -9 -> full complex transformation matrix -1 -> user-defined transformation matrix supplied immediately following the final frequency card.

1.16.4 Comments

See the Class Structure diagram for the Xmission_Line (figure 10).

1.16.5 Revision History

First Revision: 01/02/95

1.17 The Fortran_Rel_Element Class

1.17.1 Class Description

This class inherits the methods and data members of the TACS_Element class. It contains functionality to write all details which are specific to Fortran Relationship TACS cards and are not available in the TACS_Element class (see List of Methods). This functionality includes writing the Fortran relationship to the card.

1.17.2 List of Methods

Method Name	Description
void Fortran_Rel_Element ()	Constructor.
~Fortran_Rel_Element ()	Destructor.
void WriteFortranStatement (char[100] FortranStatement)	This method is used to write the Fortran Relationship statement relating to the Fortran relationship element to the data case.

1.17.3 List of Data Members

Member Name	Description
WriteFortranStatement: char[100] FortranStatement	This is a variable which contains the Fortran relationship string.

1.17.4 Comments

None.

1.17.5 Revision History

First Revision. 01/02/95

1.18 The Controller_Topology_Elem Class

1.18.1 Class Description

This class inherits the methods and data members of the TACS_Element class. It contains functionality to write all details which are specific to Controller Topology TACS cards and are not available in the TACS_Element class (see List of Methods). This functionality includes writing the three data cards required to create a Controller Topology element including writing the input names, numerators, and denominators.

1.18.2 List Of Methods

Method Name	Description
void Controller_Topology_Elem ()	Constructor.
~Controller_Topology_Elem	Destructor
void WriteControllerCard (int N, char[6] Output, char[6] Input1, char[6] Input2, char[6] Input3, char[6] Input4, char[6] Input5, char[6] K, char[6] FixedLo, char[6] FixedHi)	This method is used to write the first controller card which includes the input names, the order of the highest power of the denominator, and the output name.
void WriteNumeratorsCoeffs (char[10] N1, char[10] N2 char[10] N3, char[10] N4 char[10] N5, char[10] N6 char[10] N7)	This method is used to write the numerators of the controller topology element to the data case.
void WriteDenominatorsCoeffs (char[10] D1, char[10] D2 char[10] D3, char[10] D4 char[10] D5, char[10] D6 char[10] D7)	This method is used to write the denominators of the controller topology element to the data case.

1.18.3 List of Data Members

Member Name	Description
WriteControllerCard: int N	This variable is the order of the highest power of s, the denominator - the order of the numerator must be less than or equal to the order of the denominator
WriteControllerCard: char[6] Output	The name of the output parameter.
WriteControllerCard: char[6] Input1, char[6] Input2, char[6] Input3, char[6] Input4, char[6] Input5	The names of the 5 input parameters including their sign (-/+).
WriteControllerCard: char[6] K,	The gain in floating point format.
WriteControllerCard: char[6] FixedLo	The fixed low limit of the output parameter in floating point format.
WriteControllerCard: char[6] FixedHi	The fixed high limit of the output parameter in floating point format.
WriteNumeratorsCoeffs char[10] N1, char[10] N2 char[10] N3, char[10] N4 char[10] N5, char[10] N6 char[10] N7	The numerator coefficients in the transfer function in floating point format.
WriteDenominatorsCoeffs: char[10] D1, char[10] D2 char[10] D3, char[10] D4 char[10] D5, char[10] D6 char[10] D7	The denominator coefficients in the transfer function in floating point format.

1.18.4 Comments

None

1.18.5 Revision History

First Revision. 01/02/95

1.19 The TACS_Special_Device Class

1.19.1 Class Description

This class inherits the methods and data members of the TACS_Element class. It contains functionality to write all details which are specific to TACS Special Devices cards and are not available in the TACS_Element class (see List of Methods). This functionality includes writing the output name, and the input names.

1.19.2 List Of Methods

Method Name	Description
void TACS_Special_Device ()	Constructor.
~TACS_Special_Device	Destructor
void WriteSpecialDeviceCard (char[6] Output, char[6] Input1, char[6] Input2, char[6] Input3, char[6] Input4, char[6] Input5)	This method is used to write the input names, and the output name of the special device.

1.19.3 List of Data Members

Member Name	Description
WriteSpecialDeviceCard: char[6] Output	The name of the output parameter.
WriteSpecialDeviceCard: char[6] Input1, char[6] Input2, char[6] Input3, char[6] Input4, char[6] Input5	The names of the 5 input parameters including their sign (-/+).

1.19.4 Comments

None

1.19.5 Revision History

First Revision: 01/02/95

1.20 The TACS_Controlled_Element Class

1.20.1 Class Description

This class inherits the methods and data members of the TACS_Element class. It contains functionality to write all details which are specific to TACS Special Devices cards and are not available in the TACS_Element class (see List of Methods). This functionality includes writing the TACS string in for the TACS controlled element.

1.20.2 List Of Methods

Method Name	Description
void TACS_Controlled_Element ()	Constructor.
~TACS_Controlled_Element	Destructor
void WriteTACSString ()	This method is used to write the 'TACS' string to the TACS controlled component.

1.20.3 List of Data Members

Member Name	Description
char[3] TACSString	The 'TACS' string

1.20.4 Comments

None

1.20.5 Revision History

First Revision: 01/02/95

1.21 The Standard_Switch Class

1.21.1 Class Description

This class inherits the methods and data members of the Switch_Card class. It contains functionality to write all details which are specific to Standard Switch cards and are not available in the Switch_Card class (see List of Methods). This functionality includes writing all of the details of the standard switch other than the Itype, which is written by the Switch_Card class, and the Node1 and Node2 names, which are written by the Data_Card class.

1.21.2 List of Methods

Method Name	Description
void Standard_Switch ()	Constructor
~Standard_Switch ()	Destructor
void WriteStandardSwitchCard (char[10] TCclose, char[10] TOpen, char[10] IE)	This method writes the main Standard Switch card to the data case. This includes the start time, stop time, and current threshold of the switch.

1.21.3 List of Data Members

Member Name	Description
WriteStandardSwitchCard: char[10] TCclose	The switch closing time in floating point format, in seconds. Entering -1.0 requests that the switch be closed during the initial establishment of steady state conditions.
WriteStandardSwitchCard: char[10] TOpen	The start of attempted switch opening in floating point format, in seconds. Entering 0.0 or leaving blank requests the switch to be open at the start of the simulation (must have been previously closed by setting Tclose to -1.0).
WriteStandardSwitchCard: char[10] IE	The current threshold corresponding to the switch actually opening, in floating point format, in amperes.

1.21.4 Comments

None

1.21.5 Revision History

First Revision: 01/02/95

1.22 The Open/Close_Switch Class

1.22.1 Class Description

This class inherits the methods and data members of the Switch_Card class. It contains functionality to write all details which are specific to opening and closing switches Switch cards and are not available in the Switch_Card class, and are generic to both the opening and closing switches (see List of Methods). This functionality includes writing the average closing time, and the standard deviation of the jitter in the switch closing time.

1.22.2 List of Methods

Method Name	Description
void Open/Close_Switch ()	Constructor
~Open/Close_Switch ()	Destructor
void WriteOpen/CloseSwitchCard (char[10] TAV, char[10] TSD)	This method writes the main Opening or closing Switch card to the data case. This includes the average closing time of the switch, and the standard deviation of the jitter in the switch closing time.

1.22.3 List of Data Members

Member Name	Description
WriteOpen/CloseSwitchCard: char[10] TAV	The average closing time of the switch in floating point format, in seconds.
WriteOpen/CloseSwitchCard: char[10] TSD	The standard deviation of the jitter in the switch closing time in floating point format, in seconds.

1.22.4 Comments

None

1.22.5 Revision History

First Revision: 01/02/95

1.23 The Open/Close_Slave_Switch Class

1.23.1 Class Description

This class inherits the methods and data members of the Switch_Card class. It contains functionality to write all details which are specific to the slave opening and slave closing switches. Switch cards and are not available in the Switch_Card class, and are generic to both the slave opening and slave closing switches (see List of Methods). This functionality includes writing the average delay in closing of the switch with respect to the master switch, and the standard deviation of the delay, as well as writing the node names of the master switch.

1.23.2 List of Methods

Method Name	Description
void Open/Close_Slave_Switch ()	Constructor
~Open/Close_Slave_Switch ()	Destructor
void WriteOpen/CloseSlaveSwitchCard (char[10] TDELAY(AV), char[10] TDELAY(SD), CHAR[6] Node5, char[6] Node6)	This method writes the main opening or closing slave switch card to the data case. This includes the average delay in closing time of this slave switch, and the standard deviation of the delay in closing of this slave switch with respect to the master switch. The node names of the master switch are also written by this method

1.23.3 List of Data Members

Member Name	Description
WriteOpen/CloseSlaveSwitchCard: char[10] TDELAY(AV)	The average delay in the closing of this slave switch with respect to the master switch in floating point format, in seconds.
WriteOpen/CloseSlaveSwitchCard: char[10] TDELAY(SD)	The standard deviation of the delay in closing time of this slave switch with respect to the master switch in floating point format, in seconds.
WriteOpen/CloseSlaveSwitchCard: char[6] Node5	The name of the left / top node of the master switch.
WriteOpen/CloseSlaveSwitchCard: char[6] Node6	The name of the right / bottom node of the master switch.

1.23.4 Comments

None

1.23.5 Revision History

First Revision: 01/02/95

1.24 The Systematic_Closing_Switch Class**1.24.1 Class Description**

This class inherits the methods and data members of the Switch_Card class. It contains functionality to write all details which are specific to Systematic Closing Switch (systematically increasing the time of switching to compare results) cards and are not available in the Switch_Card class (see List of Methods). This functionality includes writing the entire list of parameters of a switch whose closing time is systematically increased, except for the ltype which is written by the Switch_Card class, the node names, which are written by the Data_Card class, and the output type which is written by the Power_Systems_Data_Card class.

1.24.2 List of Methods

Method Name	Description
void Systematic_Closing_Switch()	Constructor
~Systematic_Closing_Switch ()	Destructor
void WriteSystemClosingSwitchCard! (char[10] T, char[10] DELT, CHAR[10] NSTEP)	This method write the parameters of the systematically increasing closing time switch. These parameters include the mean closing time or the initial closing time of the switch, the time increment for the systematic increase of closing time, the number of time increments, and the 'SYSTEMATIC' string required for this card.

1.24.3 List of Data Members

Member Name	Description
WriteSystemClosingSwitchCard: char[10] T	The mean closing time (ITEST = 0) or the initial closing time (ITEST = 1) in floating point format, in seconds - ITEST can be found on the SYSTEMATIC MISCELLANEOUS DATA CARD.
WriteSystemClosingSwitchCard: char[10] DELT	The time increment for the systematic increase of closing time in floating point format, in seconds.
WriteSystemClosingSwitchCard: char[6] NSTEF	The number of time increments in floating point format.
WriteSystemClosingSwitchCard: CHAR[10] SystematicString	This is a variable which contains the string 'SYSTEMATIC' which must be written to the systematic closing switch data card.

1.24.4 Comments

None

1.24.5 Revision History

First Revision: 01/02/95

1.25 The Const_Delay_Slave_Switch Class

1.25.1 Class Description

This class inherits the methods and data members of the Switch_Card class. It contains functionality to write all details which are specific to Constant Delay Slave Switch cards and are not available in the Switch_Card class (see List of Methods). This functionality includes writing the entire list of parameters for the constant delay slaved switch with the exception of those parameters which are written in classes which are inherited by the Const_Delay_Slave_Switch class.

1.25.2 List of Methods

Method Name	Description
void Const_Delay_Slave_Switch()	Constructor
~Const_Delay_Slave_Switch ()	Destructor
void WriteConstDelaySwitchCard (char[10] Tconst, char[6] Node5, CHAR[6] Node6)	This method write the parameters of the constant delay slave switch. These parameters include the node names of the master switch, and the constant delay in closing the slave switch.

1.25.3 List of Data Members

Member Name	Description
WriteConstDelaySwitchCard: char[10] Tconst	The constant delay in closing of this slave switch with respect to the master switch in floating point format, in seconds.
WriteConstDelaySwitchCard: char[6] Node5	The name of the left / top node of the master switch in this case.
WriteConstDelaySwitchCard: char[6] Node6	The name of the right / bottom node of the master switch in this case.
WriteConstDelaySwitchCard: char[10] SystematicString	This is a variable which contains the string 'SYSTEMATIC' which is needed in the card for the constant delay slave switch.

1.25.4 Comments

None

1.25.5 Revision History

First Revision: 01/02/95

1.26 The TACS_Source Class**1.26.1 Class Description**

This class inherits the methods and data members of the Source_Card class. It contains functionality to write all details which are specific to TACS (Transient Analysis of Control Systems) Source cards and are not available in the Source_Card class (see List of Methods). This functionality includes writing the output name, the amplitude, the start time and the stop time to the data card.

1.26.2 List of Methods

Method Name	Description
void TACS_Source ()	Constructor
~TACS_Source ()	Destructor
void WriteTACSSourceCard (char[10] Output, char[10] A, char[10] Tstart, char[10] Tstop)	This method write the parameters of the TACS source to the card. These parameters include the name of the output from this TACS source, the amplitude of the source, the delay before the source is activated, and the stop time of the source.

1.26.3 List of Data Members

Member Name	Description
WriteTACSSourceCard: char[10] Output	The name assigned to the output of this source.
WriteTACSSourceCard: char[10] A	The amplitude of this source in floating point format, in volts or amperes.
WriteTACSSourceCard: char[10] Tstart	The delay before the source is activated in floating point format, in seconds. The source value will be zero during this delay
WriteTACSSourceCard: char[10] Tstop	The time at which the source is terminated - source is returned to zero, in floating point format, in seconds. If this is left blank the source will continue indefinitely.

1.26.4 Comments

None

1.26.5 Revision History

First Revision: 01/02/95

1.27 The User_Defined_Source Class

1.27.1 Class Description

This class inherits the methods and data members of the Source_Card class. It contains functionality to write all details which are specific to User Defined Source cards and are not available in the Source_Card class such as writing the v-t cards following a user defined source card (see list of Methods). This functionality includes writing the cards which specify the value of the user defined source at a particular simulation time as well as the termination card for the User Defined Source.

1.27.2 List of Methods

Method Name	Description
void User_Defined_Source ()	Constructor.
~User_Defined_Source ()	Destructor.
void WriteUserDefSourceCard (char[10] USRCE1, char[10] USRCE2, char[10] USRCE3)	This method writes the values of the user-defined sources (at the particular time step corresponding to the number of the element USRCE).

1.27.3 List of Data Members

Member Name	Description
WriteUserDefSourceCard: char[10] USRCE1, char[10] USRCE2, char[10] USRCE3	The values of the user-defined sources (at the particular simulation time step) in floating point format, in volts or amps
WriteUserDefSourceCard: char[3] TerminationString	The string '9999' needed to indicate that the user defined source cards have all been placed in the data case.

1.27.4 Comments

None

1.27.5 Revision History

First Revision: 01/02/95

1.28 The Sinusoidal_Funct_Source Class**1.28.1 Class Description**

This class inherits the methods and data members of the Source_Card class. It contains functionality to write all details which are specific to Sinusoidal Function Source cards and are not available in the Source_Card class (see List of Methods). This functionality includes writing frequency of the sinusoid, the initial condition of the sinusoid and the flag which determines whether this initial condition is in seconds or degrees.

1.28.2 List of Methods

Method Name	Description
void Sinusoidal_Funct ()	Constructor
~Sinusoidal_Funct ()	Destructor
void WriteSinusoidalFuncCard (char[10] Freq, char[10] Time-O, int A1)	This method writes all of the parameters of the sinusoidal function which are not written by the method in the Source_Card class. These parameters include the frequency of the sinusoid, the initial condition of the sinusoid and the flag for determining whether the initial condition is in seconds or degrees.

1.28.3 List of Data Members

Member Name	Description
WriteSinusoidalFuncCard: char[10] Freq	The frequency of the sinusoid in floating point format, in Hz.
WriteSinusoidalFuncCard: char[10] Time-O	The initial condition of the sinusoid in floating point format in either seconds or degrees depending on A1
WriteSinusoidalFuncCard: int A1	This is a flag which determines if the initial condition described by the Time-O data member is in degrees or seconds. A 0 indicates that the initial condition is in degrees while a 1 indicates that the initial condition is in seconds.

1.28.4 Comments

None

1.28.5 Revision History

First Revision: 01/02/95

1.29 The Step_Funct_Source Class

1.29.1 Class Description

This class inherits the methods and data members of the Source_Card class. It contains functionality to write all details which are specific to Step Function Source cards and are not available in the Source_Card class (see List of Methods). This functionality involves creating an object of type Step_Funct_Source using the constructor, since all of the functionality needed for the Step Function is included in the Source_Card class.

1.29.2 List of Methods

Method Name	Description
void Step_Funct_Source ()	Constructor
~Step_Funct_Source ()	Destructor

1.29.3 List of Data Members

None

1.29.4 Comments

None

1.29.5 Revision History

First Revision: 01/02/95

1.30 The Ramp_Funct_Source Class

1.30.1 Class Description

This class inherits the methods and data members of the Source_Card class. It contains functionality to write all details which are specific to Ramp Function Source cards and are not available in the Source_Card class (see List of Methods). This functionality includes writing the time at which the ramp reaches it's final value.

1.30.2 List of Methods

Method Name	Description
void Ramp_Funct_Source ()	Constructor
~Ramp_Funct_Source ()	Destructor
void WriteRampFuncCard (char[10] Time-0)	This method writes the parameters of the ramp function which are not already written by the Source_Card class. These parameters include the time at which the ramp reaches it's final value.

1.30.3 List of Data Members

Member Name	Description
WriteRampFuncCard: char[10] Time-0	This is the time at which the ramp reaches it's final value (amplitude) in floating point format, in seconds.

1.30.4 Comments

None

1.30.5 Revision History

First Revision: 01/02/95

1.31 The Dual_Ramp_Funct_Source Class**1.31.1 Class Description**

This class inherits the methods and data members of the Source_Card class. It contains functionality to write all details which are specific to Dual Ramp Function Source cards and are not available in the Source_Card class (see List of Methods). This functionality includes writing the time at which the ramp reaches it's peak value, the tail value of the dual ramp, and the time at which the ramp has dropped to the amplitude of the tail value.

1.31.2 List of Methods

Method Name	Description
void Dual_Ramp_Funct_Source()	Constructor
~Dual_Ramp_Funct_Source()	Destructor
void WriteDualRampFunctCard (char[10] Time-0, char[10] Tailval, char[10] Time-1)	This method writes the parameters of the ramp function which are not already written by the Source_Card class. These parameters include the time at which the ramp reaches it's peak value, the tail value of the dual ramp function, and the time at which the ramp reaches this tail value.

1.31.3 List of Data Members

Member Name	Description
WriteDualRampFunctCard: char[10] Time-0	This is the time at which the ramp reaches it's final value (amplitude) in floating point format, in seconds.
WriteDualRampFunctCard: char[10] Tailval	The particular value of the function tail in floating point format, in volts or amps.
WriteDualRampFunctCard: char[10] Time-1	The time at which the ramp has dropped to amplitude Tailval in floating point format, in seconds.

1.31.4 Comments

None

1.31.5 Revision History

First Revision: 01/02/95

1.32 The Double_Expon_Funct_Source Class

1.32.1 Class Description

This class inherits the methods and data members of the Source_Card class. It contains functionality to write all details which are specific to Double Exponential Function Source cards and are not available in the Source_Card class (see List of Methods).

1.32.2 List of Methods

Method Name	Description
void Double_Expon_Funct_Source ()	Constructor
~Double_Expon_Funct_Source()	Destructor
void WriteDoubleExponCard (char[10] A, char[10] B, char[10] C)	This method is used to write the parameters of the double exponential function which are not written by the Source_Card class. These parameters include the three defining equation parameters A, B, and C.

1.32.3 List of Data Members

Member Name	Description
WriteDoubleExponFunct: char[10] A	The variable A in the defining equation of the double exponential function: $f(t) = A [e^{Bt} - e^{Ct}].$
WriteDoubleExponFunct: char[10] B	The variable B in the above equation.
WriteDoubleExponFunct: char[10] C	The variable C in the above equation.

1.32.4 Comments

None

1.32.5 Revision History

First Revision: 01/02/95

1.33 The Simple_RLC Class

1.33.1 Class Description

This class inherits the methods and data members of the RLC_Branch class and contains functionality to write all details which are specific to Simple RLC Branch cards and are not available in the RLC_Branch class (see List of Methods). This functionality includes creating an object of type RLC_Branch since the entire functionality for the simple RLC branch is included in the RLC_Branch class.

1.33.2 List of Methods

Member Name	Description
void Simple_RLC	Constructor
~Simple_RLC	Destructor

1.33.3 List of Data Members

None

1.33.4 Comments

None

1.33.5 Revision History

First Revision: 01/02/95

1.34 The Distributed_RLC Class

1.34.1 Class Description

This class inherits the methods and data members of the RLC_Branch class and contains functionality to write all details which are specific to Distributed RLC Branch cards and are not available in the RLC_Branch class (see List of Methods). This functionality includes writing the distance of the line, the format of the inductance and capacitance inputs, and the flag for controlling whether or not the line is transposed.

1.34.2 List of Methods

Method Name	Description
void Simple_RLC	Constructor
~Simple_RLC	Destructor
void WriteDistributedRLCCard (char[10] Dist, int lline, int lpose)	This method writes the parameters of the distributed RLC branch which are not included in the RLC_Branch class. These parameters include the distance of the branch, a flag for determining the format of the distributed capacitance and inductance written in the RLC_Branch class, and a flag for controlling whether or not the line is transposed.

1.34.3 List of Data Members

Member Name	Description
WriteDistributedRLCCard: char[10] Dist	The line length in floating point format, in km. Must be entered as a negative number to signal an untransposed line.

Member Name	Description
WriteDistributedRLCCard: int ILine	This is a flag for determining the format of the capacitance and inductance in the distributed RLC branch. These values are written to the card in the RLC_Branch class. The options are as follows: For the inductance: 0 -> the modal distributed inductance in floating point format in mH/km (if XOPT is set to 0.0) or Ω/km (if XOPT is set to the frequency). 1, 2 -> the modal surge impedance in floating point format, in Ω. capacitance: 0 -> the modal distributed capacitance in floating point format, in μF/km (if COPT = 0.0) or μmho/km (if COPT = frequency). 1 -> modal propagation velocity in floating point format, in km/sec. 2 -> modal travel time in floating point format, in seconds (must exceed simulation time step Δt).
WriteDistributedRLCCard: int IPose	This is a flag for indicating if the line is transposed or untransposed. A 0 indicates that the line is transposed (will write a 0 in column 56) while a 1 indicates that the line is untransposed (will write an N in column 56).

1.34.4 Comments

None

1.34.5 Revision History

First Revision: 01/02/95

1.35 The Single_Phase_Xformer Class

1.35.1 Class Description

This class inherits the methods and data members of the Xformer class and contains functionality to write all details which are specific to Single Phase Transformer cards and are not available in the Xformer class (see List of Methods). This functionality includes creating an object of type Xformer since the entire functionality for the Single Phase Transformer is included in the Xformer class.

1.35.2 List of Methods

Member Name	Description
void Single_Phase_Xformer	Constructor
~Single_Phase_Xformer	Destructor

1.35.3 List of Data Members

None

1.35.4 Comments

None

1.35.5 Revision History

First Revision: 01/02/95

1.36 The Shell_Type_Xformer Class

1.36.1 Class Description

This class inherits the methods and data members of the Xformer class and contains functionality to write all details which are specific to Shell-Type Transformer cards and are not available in the Xformer class (see List of Methods). This functionality includes writing the transformer name, and the reluctance of the zero-sequence flux path.

1.36.2 List of Methods

Member Name	Description
void Shell_Type_Xform	Constructor
~Shell_Type_Xformer	Destructor
void WriteShellTypeCard (char[10] Ref, char[10] Ro)	This method is used to write the details of the three phase transformer that aren't written in the Xformer class. These parameters include the reference name for the transformer and the reluctance of the zero-sequence path. This method also calls the Xformer class WriteWindingCard method for every phase of the transformer separately.

1.36.3 List of Data Members

Data Members	Description
WriteShellTypeCard: char[10] Ref	A unique name for this transformer. Use Unknown!
WriteShellTypeCard: char[10] Ro	The reluctance of the zero-sequence flux path in floating point format.

1.36.4 Comments

None

1.36.5 Revision History

First Revision: 01/02/95

1.37 The Core_Type_Xformer Class**1.37.1 Class Description**

This class inherits the methods and data members of the Xformer class and contains functionality to write all details which are specific to Core-Type Transformer cards and are not available in the

Xformer class (see List of Methods). This functionality includes writing the transformer name, and the reluctance of the zero-sequence flux path.

1.37.2 List of Methods

Member Name	Description
void Core_Type_Xformer	Constructor
~Core_Type_Xformer	Destructor
void WriteCoreTypeCard (char[10] Ref, char[10] Ro)	This method is used to write the details of the three phase transformer that aren't written in the Xformer class. These parameters include the reference name for the transformer and the reluctance of the zero-sequence path. This method also calls the Xformer class WriteWindingCard method for every phase of the transformer separately.

1.37.3 List of Data Members

Data Members	Description
WriteCoreTypeCard: char[10] Ref	A unique name for this transformer. Use Unknown!
WriteCoreTypeCard: char[10] Ro	The reluctance of the zero-sequence flux path in floating point format.

1.37.4 Comments

None

1.37.5 Revision History

First Revision: 01/02/95

1.38 The Nominal_Pi_Xmission_Line Class

1.38.1 Class Description

This class inherits the methods and data members of the Xmission_Line class and contains functionality to write all details which are specific to Nominal-Pi Transmission line model cards and are not available in the Xmission_Line class (see List of Methods).

1.38.2 List of Methods

Method Name	Description
void Nominal_Pi_Xmission_Line ()	Constructor
~Nominal_Pi_Xmission_Line ()	Destructor
void WriteNominalPiFrequencyCard ()	This method is used to call the Xmission_Line class WriteFrequencyCard method, and gives the Ipun variable as 1 so that the data case knows that the Nominal-Pi model is required. This method has the same inputs as the Xmission_Line WriteFrequencyCard method.

1.38.3 List of Data Members

See the Xmission_Line class data members for the WriteFrequencyCard method.

1.38.4 Comments

None

1.38.5 Revision History

First Revision: 01/02/95

1.39 The Equivalent_Pi_Xmission_Line Class

1.39.1 Class Description

This class inherits the methods and data members of the Xmission_Line class and contains functionality to write all details which are specific to Equivalent-Pi Transmission line model cards and are not available in the Xmission_Line class (see List of Methods).

1.39.2 List of Methods

Member Name	Description
void Equivalent_Pi_Xmission_Line ()	Constructor
~Equivalent_Pi_Xmission_Line ()	Destructor
void WriteEquivalentPiFrequencyCard ()	This method is used to call the Xmission_Line class WriteFrequencyCard method. It has the same inputs as the Xmission_Line WriteFrequencyCard method, and gives the lpipr variable as either 54, 55, 56, or 57 so that the data case knows that the Nominal-Pi model is required.

1.39.3 List of Data Members

See the Xmission_Line class data members for the WriteFrequencyCard method.

1.39.4 Comments

None

1.39.5 Revision History

First Revision: 01/02/95

1.40 The Modal_Parameter_Xmission_Line Class

1.40.1 Class Description

This class inherits the methods and data members of the Xmission_Line class and contains functionality to write all details which are specific to Modal Parameter Transmission line model cards and are not available in the Xmission_Line class (see List of Methods).

1.40.2 List of Methods

Method Name	Description
void Modal_Parameter_Xmission_Line ()	Constructor
~Modal_Parameter_Xmission_Line ()	Destructor
void WriteModalParameterFrequency Card ()	This method is used to call the Xmission_Line class WriteFrequencyCard method. It has the same inputs as the Xmission_Line WriteFrequencyCard method, and gives the modal variable as 1 so that the line is untransposed, and the model is immediately taken as the Modal parameters model.

1.40.3 List of Data Members

See the Xmission_Line class data members for the WriteFrequencyCard method.

1.40.4 Comments

None

1.40.5 Revision History

First Revision: 01/02/95

1.41 The MARTI_Xmission_Line Class

1.41.1 Class Description

This class inherits the methods and data members of the Xmission_Line class and contains functionality to write all details which are specific to JMARTI Transmission line model cards and are not available in the Xmission_Line class (see List of Methods). This functionality includes methods which call a separate Frequency card three times at different frequencies, as well as sets up the JMARTI support program.

1.41.2 List of Methods

Method Name	Description
void MARTI_Xmission_Line ()	Constructor
~MARTI_Xmission_Line ()	Destructor
void SetupMARTIModel ()	This method writes the strings 'JMARTI SETUP' and 'ERASE' to the card in order to indicate the beginning of a JMARTI model. This method then calls the required Xmission_Line class methods in the required order.
void WriteSeparateFrequencyCard (char[10] RHO, char[10] Freq1, char[10] Freq2, char[10] Freq3, int FCAR, char[10] Dist, int ISEG, int Mutual, int Ipnt, int Ipun, int Modal, int Tr)	This method is used instead of the Xmission_Line WriteFrequencyCard method. It is called three times for the MARTI model, each time using a different frequency for the Freq data member.

1.41.3 List of Data Members

Member Name	Description
WriteSeparateFrequencyCard: char[10] RHO	The ground resistivity in real number format, in Ωm

Member Name	Description
WriteSeparateFrequencyCard: char[10] Freq1	The frequency at which the diagonalizing transformation is to be performed if the line is untransposed, in real number format, in Hz.
WriteSeparateFrequencyCard: char[10] Freq2	The frequency of the steady state solution, in real number format, in Hz.
WriteSeparateFrequencyCard: char[10] Freq3	The lowest frequency of the frequency domain description of the transmission line in real number format, in Hz.
WriteSeparateFrequencyCard: char[10] Dist	The length of the transmission line in real number format, in km.
WriteSeparateFrequencyCard: int ISEG	This is a flag for specifying continuity of ground wires. The options are 0 for continuous ground wires (normal situation), and 1 for segmented ground wires (cut at short intervals).
WriteSeparateFrequencyCard: int Mutual	This is a flag for indicating the presence of a communication circuit parallel to the power circuit. A 0 indicates no communication circuit while a 1 indicates the presence.
WriteSeparateFrequencyCard: int Modal	This is a flag to indicate whether the transmission line is continuously transposed or untransposed. A 0 indicates that the line is continuously transposed while a 1 indicates that it is untransposed.
WriteSeparateFrequencyCard: int Tr	This is a flag for requesting either the full complex, or real part only, transformation matrix for the modal analysis of untransposed lines: 0 -> real part only - used generally -9 -> full complex transformation matrix -1 -> user-defined transformation matrix supplied immediately following the final frequency card.

1.41.4 Comments

None

1.41.5 Revision History

First Revision: 01/02/95

1.42 The Relay_Operated_Switch Class

1.42.1 Class Description

This class inherits the methods and data members of the TACS_Special_Device class and contains functionality to write all details which are specific to Relay Operated Switch cards and are not available in the TACS_Special_Device class (see List of Methods). This functionality includes writing the gain, the fixed threshold, the driving signal and the named threshold to the data card for this switch.

1.42.2 List of Methods

Method Name	Description
void Relay_Operated_Switch ()	Constructor
~Relay_Operated_Switch ()	Destructor
void WriteRelayOpSwitchCard (char[6] A, char[6] B, char[6] C, char[6] D, char[6] E)	This method writes the parameters A, B, C, D, and E to the Relay operated switch card. These elements represent the gain, fixed threshold, named threshold, driving signal, and a flag which determines the operating mode of the switch.

1.42.3 List of Data Members

Member Name	Description
WriteRelayOpSwitchCard: char[6] A	This is the gain in floating point format. The default is 1.0 (see User Reference Manual - The Relay-Operated Switch for details on the block diagram of the relay-operated switch).
WriteRelayOpSwitchCard: char[6] B	This is the value of the FIXED THRESHOLD of the switch in floating point format.

Member Name	Description
WriteRelayOpSwitchCard: char[6] C	This is a flag which determines the operating mode of the switch. The options are: Switch normally open (0) Switch normally closed (1) Switch normally open, but closed during the initial establishment of steady state conditions (-2).
WriteRelayOpSwitchCard: char[6] D	The name of the TACS parameter that will supply the value of the NAMED THRESHOLD level (optional).
WriteRelayOpSwitchCard: char[6] E	The name of the TACS parameter that will supply the value of the DRIVING SIGNAL level.

1.42.4 Comments

None

1.42.5 Revision History

First Revision: 01/02/95

1.43 The Level_Triggerred_Switch Class

1.43.1 Class Description

This class inherits the methods and data members of the TACS_Special_Device class and contains functionality to write all details which are specific to Level Triggerred Switch cards and are not available in the TACS_Special_Device class (see List of Methods). This functionality includes writing the gain, the fixed threshold, the driving signal and the named threshold to the data card for this switch.

1.43.2 List of Methods

Method Name	Description
void Level_Trigerred_Switch ()	Constructor
~Level_Trigerred_Switch ()	Destructor
void WriteLevelTrigerredSwitchCard {char[6] A, char[6] B, char[6] C, char[6] D, char[6] E}	This method writes the parameters A, B, C, D, and E to the Level Trigerred switch card. These elements represent the gain, fixed threshold, named threshold, driving signal, and a flag which determines the operating mode of the switch.

1.43.3 List of Data Members

Member Name	Description
WriteLevelTrigerredSwitchCard: char[6] A	This is the gain in floating point format. The default is 1.0 (see User Reference Manual - The Level-Trigerred Switch for details on the block diagram of the level-triggerred switch).
WriteLevelTrigerredSwitchCard: char[6] B	This is the value of the FIXED THRESHOLD of the switch in floating point format.
WriteLevelTrigerredSwitchCard: char[6] C	This is a flag which determines the operating mode of the switch. The options are: Switch normally open (0) Switch normally closed (1) Switch normally open, but closed during the initial establishment of steady state conditions (-2) Switch normally closed, and closed during the initial establishment of steady state conditions (-1).
WriteLevelTrigerredSwitchCard: char[6] D	The name of the TACS parameter that will supply the value of the NAMED THRESHOLD level (optional).
WriteLevelTrigerredSwitchCard: char[6] E	The name of the TACS parameter that will supply the value of the DRIVING SIGNAL level.

1.43.4 Comments

None

1.43.5 Revision History

First Revision: 01/02/95

1.44 The Controlled_Integrator Class

1.44.1 Class Description

This class inherits the methods and data members of the TACS_Special_Device class and contains functionality to write all details which are specific to Controlled Integrator cards and are not available in the TACS_Special_Device class (see List of Methods). This functionality includes writing the gain, the value of the denominators in the equation for the integrator, the name that supplies the CONTROL value in the equation, and the name that supplies the RESET parameter.

1.44.2 List of Methods

Method Name	Description
void Controlled_Integrator ()	Constructor
~Controlled_Integrator ()	Destructor
void WriteControlledIntegratorCard (char[6] A, char[6] B, char[6] C, char[6] D, char[5] E)	This method writes the parameters A, B, C, D, and E to the Controlled Integrator card. These elements represent the gain, the value of the denominators D_0 and D_1 , in the equation for the integrator, the name that supplies the CONTROL value in the equation, and the name that supplies the RESET parameter.

1.44.3 List of Data Members

Member Name	Description
WriteControlledIntegratorCard: char[6] A	This is the gain in floating point format. The default is 1.0 (see User Reference Manual - The Controlled Integrator for details on the block diagram of the controlled integrator).
WriteControlledIntegratorCard: char[6] B	The value of D_0 in floating point format.
WriteControlledIntegratorCard: char[6] C	The value of D_1 in floating point format.
WriteControlledIntegratorCard: char[6] D	The name of the TACS parameter that will supply the value of the CONTROL value (or leave blank).
WriteControlledIntegratorCard: char[6] E	The name of the TACS parameter that will supply the value of the RESET value.

1.44.4 Comments

None

1.44.5 Revision History

First Revision: 01/02/95

1.45 The Simple_Differentiator Class

1.45.1 Class Description

This class inherits the methods and data members of the TACS_Special_Device class and contains functionality to write all details which are specific to Simple Differentiator cards and are not available in the TACS_Special_Device class (see List of Methods). This functionality includes writing the gain.

1.45.2 List of Methods

Method Name	Description
void Simple_Differentiator ()	Constructor
~Simple_Differentiator ()	Destructor
void WriteSimpleDifferentiatorCard (char[6] A)	This method writes the parameter A. This element represent the gain.

1.45.3 List of Data Members

Member Name	Description
WriteControlledIntegratorCard: char[6] A	This is the gain in floating point format. The default is 1.0 (see User Reference Manual - The Simple Differentiator for details on the block diagram of the simple differentiator).

1.45.4 Comments

None

1.45.5 Revision History

First Revision: 01/02/95

1.46 The Opening_Switch Class**1.46.1 Class Description**

This class inherits the methods and data members of the Open/Close_Switch class and contains functionality to write all details which are specific to Opening Switch cards and are not available in the Open/Close_Switch class (see List of Methods). This functionality includes writing the current threshold to the data case as well as writing the strings which are unique to an opening switch to the opening switch card.

1.46.2 List of Methods

Method Name	Description
void Opening_Switch ()	Constructor
~Opening_Switch ()	Destructor
void WriteOpeningSwitchCard (char[10] IE)	This method writes the main opening switch card to the data case. This includes the current threshold, as well as the '3333.STATISTICS' string.

1.46.3 List of Data Members

Member Name	Description
WriteOpeningSwitchCard: char[10] IE	The current threshold corresponding to the switch actually opening in floating point format, in seconds.
WriteOpeningSwitchCard: char[15] StatisticsString	This is a variable which contains the string '3333.STATISTICS' which is needed in the opening switch c

1.46.4 Comments

None

1.46.5 Revision History

First Revision: 01/02/95

1.47 The Closing_Switch Class

1.47.1 Class Description

This class inherits the methods and data members of the Open/Close_Switch class and contains functionality to write all details which are specific to Closing Switch cards and are not available in the Open/Close_Switch class (see List of Methods). This functionality includes writing the strings which are unique to a closing switch, to the closing switch card.

1.47.2 List of Methods

Method Name	Description
void Closing_Switch ()	Constructor
~Closing_Switch ()	Destructor
void WriteClosingSwitchCard ()	This method writes the main closing switch card to the data case. This includes the 'STATISTICS' string.

1.47.3 List of Data Members

Member Name	Description
WriteClosingSwitchCard: char[10] StatisticsString	This is a variable which contains the string 'STATISTICS' which is needed in the opening switch card.

1.47.4 Comments

None

1.47.5 Revision History

First Revision: 01/02/95

1.48 The Opening_Slave_Switch Class

1.48.1 Class Description

This class inherits the methods and data members of the Open/Close_Slave_Switch class and contains functionality to write all details which are specific to Slave Closing Switch cards and are not available in the Open/Close_Slave_Switch class (see List of Methods). This functionality includes writing the current threshold of the switch as well as writing the strings necessary to indicate a slave opening switch.