

1.48.2 List of Methods

Method Name	Description
void Opening_Slave_Switch ()	Constructor
~ Opening_Slave_Switch ()	Destructor
void WriteOpeningSlaveSwitchCard (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.

1.48.3 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 card.

1.48.4 Comments

None

1.48.5 Revision History

First Revision: 01/02/95

1.49 The Closing_Slave_Switch Class**1.49.1 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 the Slave Closing Switch cards and are not available in the Open/Close_Slave_Switch class (see List of Methods). This functionality includes writing the strings which are unique to a slave closing switch, to the slave closing switch card.

1.49.2 List of Methods

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

1.49.3 List of Data Members

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

1.49.4 Comments

None

1.49.5 Revision History

First Revision: 01/02/95

1.50 The DC_Level Class**1.50.1 Class Description**

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). This functionality involves creating an object of type TACS_Source using the constructor, since all of the functionality needed for the DC Level or Single Pulse is included in the TACS_Source Class.

1.50.2 List of Methods

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

1.50.3 List of Data Members

None

1.50.4 Comments

None

1.50.5 Revision

First Revision: 01/4/95

1.51 The TACS_Sinusoid Class**1.51.1 Class Description**

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). This functionality includes writing the frequency of the source as well as the initial phase angle of the source.

1.51.2 List of Methods

Method Name	Description
void TACS_Sinusoid ()	Constructor
~TACS_Sinusoid ()	Destructor
void WriteTACSSinusoidCard (char[10] B, char[10] C)	This method is used to write the frequency of the source, and the initial phase angle of the source.

1.51.3 List of Data Members

Member Name	Description
WriteTACSSinusoidCard: char[10] B	The frequency of the source in floating point format, in Hz.
WriteTACSSinusoidCard: char[10] C	The initial phase angle of the source in floating point format, in degrees.

1.51.4 Comments

None

1.51.5 Revision History

First Revision: 01/02/95

1.52 The Repetitive_Pulse Class**1.52.1 Class Description**

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).

1.52.2 List of Methods

Method Name	Description
void Repetitive_Pulse ()	Constructor
~Repetitive_Pulse ()	Destructor
void WriteRepetitivePulseCard (char[10] B, char[10] C)	This method is used to write the period of the pulses and the pulse width of the pulses.

1.52.3 List of Data Members

Member Name	Description
WriteRepetitivePulseCard: char[10] B	The period of the pulses in floating point format, in seconds.
WriteTACSSinusoidCard: char[10] C	The pulse width in floating point format, in seconds.

1.52.4 Comments

None

1.52.5 Revision History

First Revision: 01/02/95

1.53 The Repetitive_Ramp Class**1.53.1 Class Description**

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).

1.53.2 List of Methods

Method Name	Description
void Repetitive_Ramp ()	Constructor
~Repetitive_Ramp ()	Destructor
void WriteRepetitiveRampCard (char[10] B)	This method is used to write the period of the repetitive ramp.

1.53.3 List of Data Members

Member Name	Description
WriteRepetitiveRampCard: char[10] B	The period of the repetitive ramp in floating point format, in seconds.

1.53.4 Comments

None

1.53.5 Revision History

First Revision: 01/02/95

1.54 The TACS_Controlled_Source Class**1.54.1 Class Description**

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).

1.54.2 List of Methods

Method Name	Description
void TACS_Controlled_Source ()	Constructor
~TACS_Controlled_Source ()	Destructor
void WriteTACSCntrlSourceCard (char[10] Name, int IV)	This method is used to write the parameters of the TACS controlled source which are not written by the TACS_Source class. These parameters include the name of the TACS parameter and the flag to control whether the source is a current or voltage source.

1.54.3 List of Data Members

Member Name	Description
WriteTACSCntrlSourceCard: char[10] Name	The name of the TACS parameter (same name as node source is connected to.
WriteTACSCntrlSourceCard: int IV	This is a flag to control whether the source is a voltage source or a current source. A 0 indicates that it is a voltage source, while a -1 indicates that it is a current source.

1.54.4 Comments

None

1.54.5 Revision History

First Revision: 01/02/95

1.55 List of Files

Every class in this section has its own class (.cpp) file named after the class, and a corresponding header (.h) file.

Author: Orbach Tamir.

Name of thesis: Strategies for designing useful interfaces for software in high voltage applications and a case study of these principles applied to the alternative transients program (ATP).

PUBLISHER:

University of the Witwatersrand, Johannesburg

©2015

LEGALNOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page) for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.