

	A	Q	R	S	T	U	V
1	Subsystem:						
2	COMMAND MNEMONIC	ARG DATA RANGE LOW	ARG DATA RANGE HIGH	ARG MNEMONIC	ARG MNEMONIC VALUE	ARG MNEMONIC DEFAULT Y/N	ARG MNEMONIC MODE
3	ALICE_NOP						
4							
5	ALICE_ENTER_SAFE_STATE						
6	ALICE_ENTER_CHECKOUT_STATE						
7	ALICE_CONFIRM_CRITICAL						
8	ALICE_CONFIRM_CRITICAL	0x4100	0x4130	START_HISTOGRAM	0x4100		
9	ALICE_CONFIRM_CRITICAL			START_PIXELLIST			
10	ALICE_CONFIRM_CRITICAL			SET_PARAMETER			
11	ALICE_CONFIRM_CRITICAL			STORE_PARAMETERS			
12	ALICE_CONFIRM_CRITICAL			OPEN_DOOR			
13	ALICE_CONFIRM_CRITICAL			ACTIVATE			
14	ALICE_CONFIRM_CRITICAL						
15	ALICE_CONFIRM_CRITICAL						

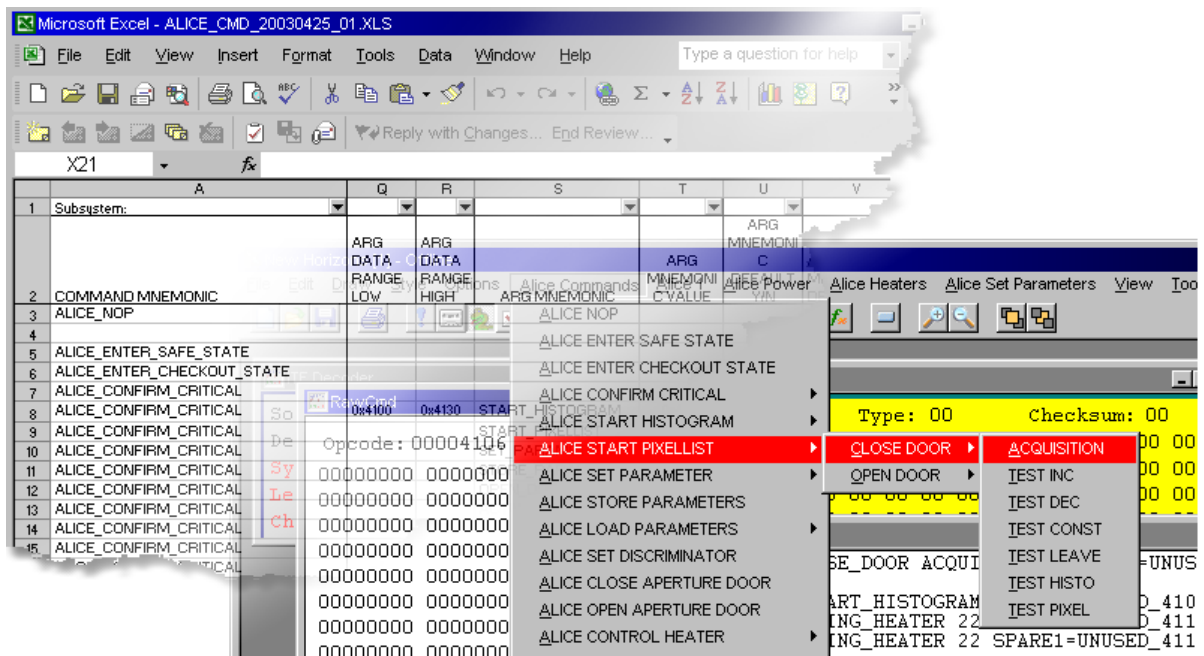
# GSEOS

## Excel Spreadsheet Conversion Tools

[www.gseos.com](http://www.gseos.com)

GSE Software, Inc. 1998 - 2016

# 1 GSEOS ExcelLoad Package



## 1.1 Introduction

The ExcelLoad package is a mechanism for converting command and telemetry databases from an Excel spreadsheet file conforming to the APL specification.

Loading a new CMD or TLM database does not impact any of the other GSEOS configurations. No customization is required to load a new CMD or TLM database. This allows for easy updating of new CMD or TLM definitions and flow down to the GSE configuration.

There are some exceptions to the 'zero impact' feature: Changes of command and telemetry point mnemonics may impair the GSEOS configuration if these names are used in downstream commands or decoders.

The conversion process is actually two different mechanisms, a telecommand spreadsheet (CMD) loader and a telemetry spreadsheet (TLM) loader. Both mechanisms are accessed in the same way by opening file with the .xls (Excel spreadsheet) extension.

### Quick Start

In order to import a CMD or TLM database you have to follow the steps below:

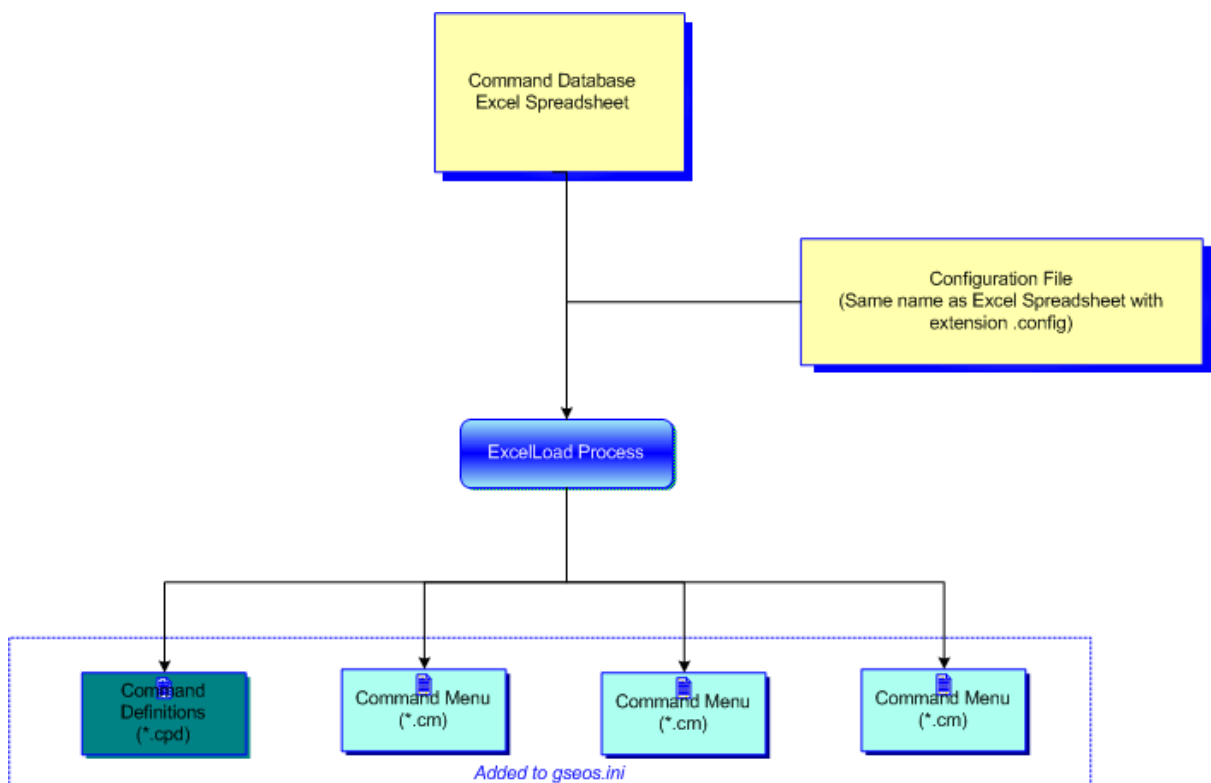
1. Open the Excel telecommand database spreadsheet in GSEOS. This will perform the database import and generate the appropriate configuration files.
2. Restart GSEOS. This will load the new files into GSEOS.

## 1.2 CMD Load

The CMD Loader generates two or more output files: The command definition file (\*.cpd) as specified in the Excel spreadsheet and one or more command menu (\*.cm) files containing command menus that list the commands defined.

Besides the Excel input file additional configuration information can be supplied in a [configuration file](#). The configuration must have the same file name as the Excel spreadsheet file with the extension .config instead of .xls.

The created files will be automatically added to your gseos.ini file. On the next restart these files will be loaded automatically. The command definitions are loaded and therefore all commands are available to screen definitions, command buttons, command menus, the STOL emulator etc..



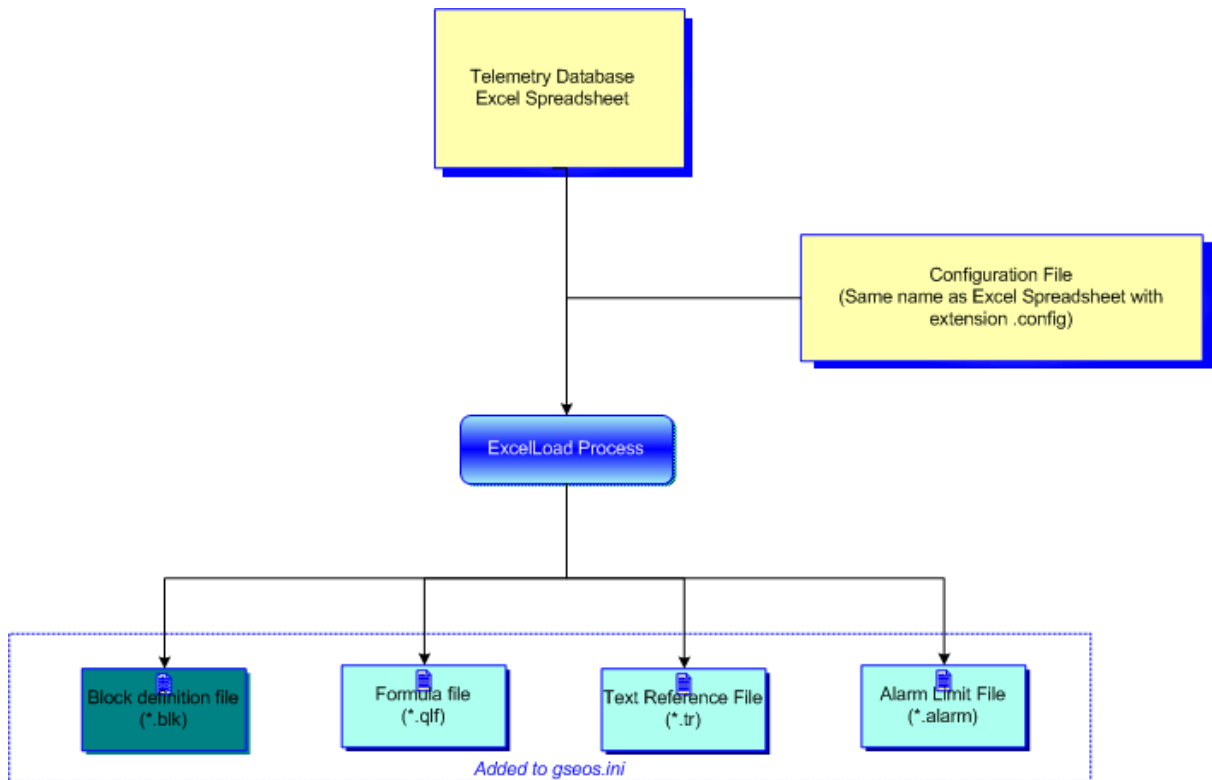
## 1.3 TLM Load

The TLM Loader will generate several output files: The block definition file (\*.blk), an optional formula file (\*.qlf), an optional text reference file (\*.tr) and an optional alarm limit file.

Besides the Excel input file additional configuration information can be supplied in a [TLM configuration file](#). The configuration must have the same file name as the Excel spreadsheet file with the extension .config instead of .xls.

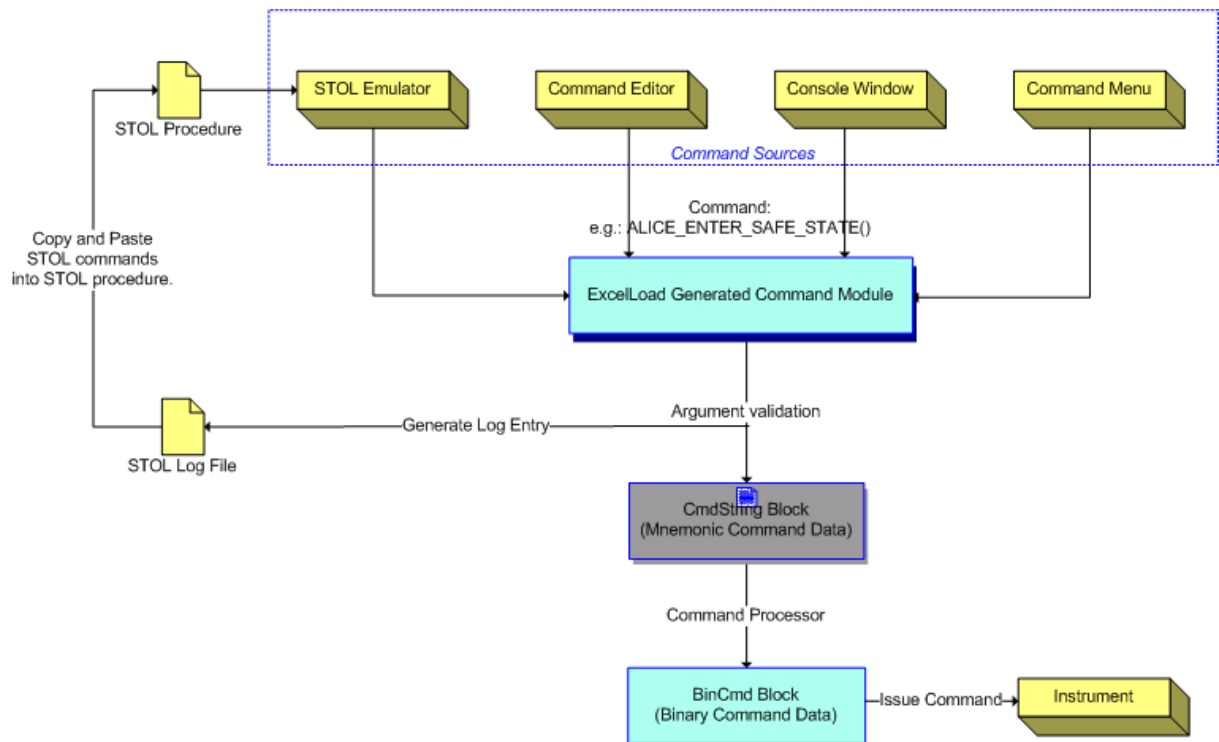
The created files will be automatically added to your gseos.ini file. On the next restart these files will be loaded automatically.

The following diagram shows the TLM conversion process:



## 1.4 Command Flow

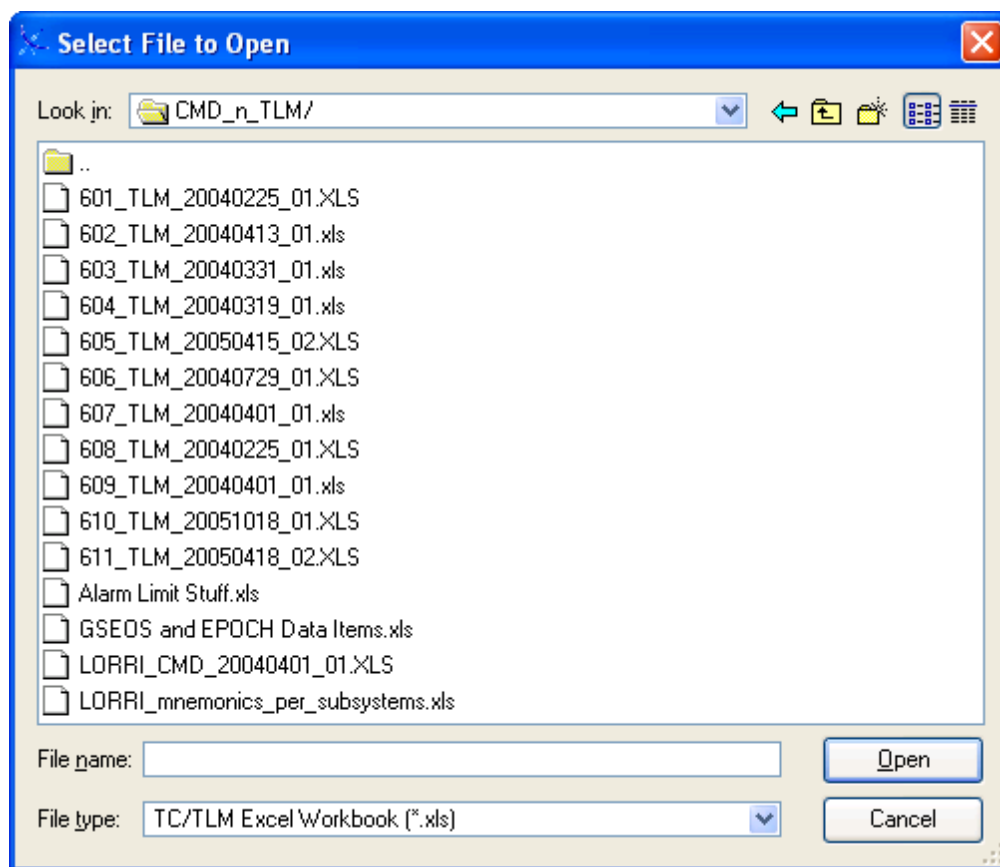
The picture below shows the logical command flow and the architectural design.



## 1.5 Usage

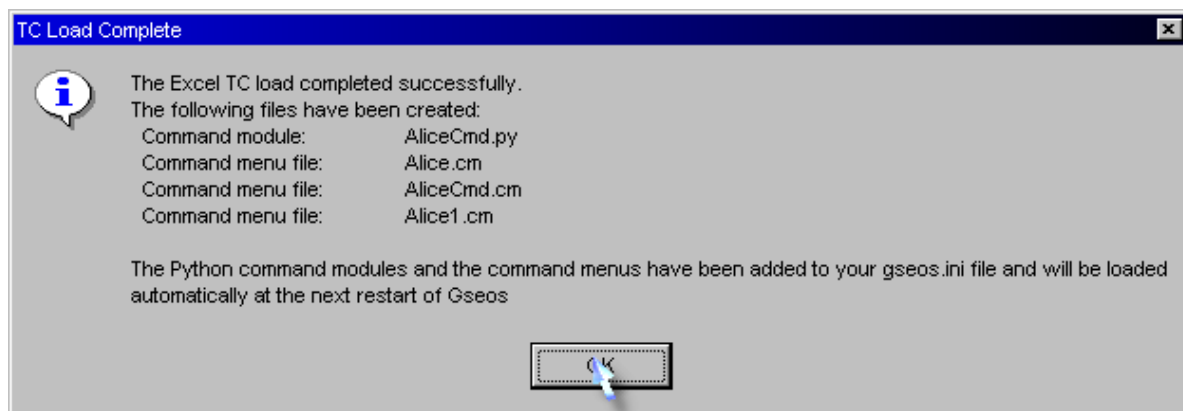
### Excel Spreadsheet Loader

To convert the Excel command or telemetry database into GSEOS configuration files select the File/Open menu from the GSEOS main menu. This opens a file open dialog as depicted below:



Select TC/TLM Excel Workbook (\*.xls) for the file type and select the command or TLM database you want to convert. You must have a [configuration file](#) with the same name as the Excel spreadsheet file with the extension .config in the same directory as the Excel file.

Once you select the Excel file and click open the conversion takes place. In case of errors a message box will list the errors. At the end of the conversion a message box indicates the files generated by the conversion:



This concludes the conversion. In order to load the new configuration files GSEOS must be

restarted.

## 1.6 Excel Spreadsheet Spec

Column Name	Description	# Of Chars	Comments	Underlying ORACLE table	Input Syntax Checks (besides length check)
COMMAND MNEMONIC	Unique command mnemonic for a command, 1st 3 characters are reserved for the subsystem id, e.g. "CD_" for C&DH commands. Refer to the program specific Mnemonics Abbreviation List for standard abbreviations. No blanks or special characters except underscore allowed. Use uppercase.	35	Required for all rows.	Various	Must not contain any characters other than alphanumeric, underscore, or period.
ARG DATA WORD NUMBER	Number ordering each argument associated with a command including default data words. Note that arguments with arg_type of KEYWORD must have higher data word numbers than arguments with arg_type of POSITION. This line is blank for the first row of a command, and required for subsequent rows.	11	Required for all rows of a command associated with an argument.	CMD_LINE_ARGS	Must be numeric.
ALIAS MNEMONIC	Alternate mnemonic for command. Works the same as the command mnemonic. The alias mnemonic follows the same naming conventions as the mnemonic, as described above.	35	Not required. If it is there, cmd_load program takes it from the first row of the command. It is ignored if it is on subsequent rows.	COMMANDS	Must not contain any characters other than alphanumeric, underscore, or period.
CMD DESCRIPTION	Description of the command. Valid for the first row of a command.	63	Required for the first row associated with a command.	COMMANDS	Cannot contain commas (upsets PROC code that relies on comma separated files), square brackets (upsets command generator), or double quotes (not allowed through EPOCH ORACLE form).
CRITICAL CMD?	Causes "Are you sure" pop-up for operator	1	Defaults to N.	COMMANDS	Must be Y, N, or blank.
CMD UPLINK CODE	Command opcode.	10	Required for the first row of a command.	COMMANDS	Must be numeric.
CMD FORMAT NAME	Format from database and cmd_fmt_desc.dat	15	Required for the first row of a command.	COMMANDS	Must be a valid command format for this spacecraft.
ARG NUM BITS	Number of bits in the argument	11	Required for the first row of an argument or default data word.	CMD_LINE_ARGS	Must be numeric.
ARG DESCRIPTION	Description of the argument. Valid for the first row of each argument.	63	Required for the first row associated with an argument.	CMD_LINE_ARGS	Cannot contain commas (upsets PROC code that relies on comma separated files), square brackets (upsets command generator), or double quotes (not allowed through EPOCH ORACLE form).
ARG VALUE REQ	Does this argument have to be included in the command: Y or N. Should be Y if arg_type is position, but can be N if arg_type is keyword.	1	Defaults to Y for the first row of an argument.	CMD_LINE_ARGS	Must be Y, N, or blank.
ARG TYPE	Is argument recognized by position within the command (e.g. the second argument after the mnemonic) or is it identified by a keyword (e.g. F1=filename).	15	POSITION or KEYWORD. Defaults to POSITION. Valid for the first row of an argument.	CMD_LINE_ARGS	Must be POSITION or KEYWORD.
ARG KEYWORD	The keyword that is associated with an argument that has an ARG TYPE of KEYWORD.	35	Valid only if this row has an ARG TYPE of KEYWORD.	CMD_LINE_ARGS	Must not contain any characters other than alphanumeric, underscore, or period.
ARG VALUE SELECTED TYPE	Specifies whether argument may be numeric only, mnemonic only, either, or a file name.	15	Required for the first row of an argument: NUMERIC_ONLY, MNEMONIC_ONLY, EITHER, or FILE_NAME.	CMD_LINE_ARGS	Must be NUMERIC_ONLY, MNEMONIC_ONLY, EITHER, or FILE_NAME.
ARG DATA TYPE	data type of argument		Required for the first row of an argument: UNSIGNED, SIGNED, SIGNED_MAG (1's complement), FLOAT32_IEEE, or FLOAT64_IEEE, TIME32_GPS, RX2010, FLOAT32_RX2010, FLOAT48_RX2010, and a TBD STEREO_TIME. There exist other possible values that are rarely used at APL.	CMD_LINE_ARGS	Must be UNSIGNED, SIGNED, SIGNED_MAG, FLOAT32_IEEE, FLOAT64_IEEE, TIME32_GPS, RX2010, FLOAT32_RX2010, or FLOAT48_RX2010
ARG SCALE FACTOR	multiply input argument by a constant factor	24	Defaults to 1 for the first row of an argument.	CMD_LINE_ARGS	Must be numeric.
ARG OFFSET	add a constant to the input argument	24	Defaults to 0 for the first row of an argument.	CMD_LINE_ARGS	Must be numeric.
ARG DATA RANGE LOW	Lowest acceptable value for this argument	16	Optional.	CMD_ARG_DATA_RNGS	Must be numeric.
ARG DATA RANGE HIGH	Highest acceptable value for this argument	16	Optional.	CMD_ARG_DATA_RNGS	Must be numeric.
ARG MNEMONIC	Argument mnemonic	35	Optional. Valid only if this is a row associated with an argument.	CMD_ARG_MNEMONICS	Must not contain any characters other than alphanumeric, underscore, or period.
ARG MNEMONIC VALUE	Equivalent raw value corresponding to the specified argument mnemonic, e.g. argument mnemonic=ON, value=1	10	Required if this row includes an ARG MNEMONIC.	CMD_ARG_MNEMONICS	Must be numeric.
DEFAULT Y/N	If the user does not enter this argument, is this value the default?	1	Required if this row includes an ARG MNEMONIC. Only one arg mnemonic per command may be the default. This field defaults to N if not entered into the spreadsheet.	CMD_ARG_MNEMONICS	Must be Y, N, or blank.
ARG MNEMONIC DESCRIPTION	Description of this argument mnemonic/value. Often the same as the mnemonic, but sometimes a more complete description if the mnemonic is an abbreviation.	63	Required if this row includes an ARG MNEMONIC.	CMD_ARG_MNEMONICS	Must not contain commas or parentheses.
ARG FIXED DATA WORD VALUE	A fixed data word is an argument that has a constant value, and therefore need not be entered by the operator. This column contains the constant value of this argument.	35	Required if there is/are fixed data word(s) for this command.	CMD_OPER_DATA	Must be numeric.

## 1.7 Command Menu Generation

Besides the command module the ExcelLoad process also generates one or more command menus. The command menu file is added to the gseos.ini file so it will be loaded

automatically on the next start. The command menu lists all commands that are defined in the command module and creates submenus for mnemonic parameters. A sample command menu file is listed below. You should never directly modify the generated command menus since these files get overwritten the next time the ExcelLoad process is run. If you want to customize the command menu(s) you can do so in the [configuration file](#).

```
Menu &Alice-TC
{
  Menuitem &01 - Nop, Common.igseAPI.CMD('ALICE_NOP')
  Menuitem &02 - Safe, Common.igseAPI.CMD('ALICE_ENTER_SAFE_STATE')
  Menuitem &03 - Checkout, Common.igseAPI.CMD('ALICE_ENTER_CHECKOUT_STATE')
  Popup &04 - Confirm Critical
  {
    Menuitem &Start Histogram, Common.igseAPI.CMD('ALICE_CONFIRM_CRITICAL',
('START_HISTOGRAM'))
    Menuitem &Start Pixel List, Common.igseAPI.CMD('ALICE_CONFIRM_CRITICAL',
('START_PIXELLIST'))
    Menuitem &Set Parameter, Common.igseAPI.CMD('ALICE_CONFIRM_CRITICAL',
('SET_PARAMETER'))
    Menuitem &Store Params, Common.igseAPI.CMD('ALICE_CONFIRM_CRITICAL',
('STORE_PARAMETERS'))
    Menuitem &Open ApDoor, Common.igseAPI.CMD('ALICE_CONFIRM_CRITICAL', ('OPEN_DOOR'))
    Menuitem &Set Hv On, Common.igseAPI.CMD('ALICE_CONFIRM_CRITICAL',
('ACTIVATE_HVPS'))
    Menuitem &Activate WPA/SMA, Common.igseAPI.CMD('ALICE_CONFIRM_CRITICAL',
('ACTIVATE_WPASMA'))
    Menuitem &Start Program, Common.igseAPI.CMD('ALICE_CONFIRM_CRITICAL',
('START_PROGRAM'))
    Menuitem &Request restart, Common.igseAPI.CMD('ALICE_CONFIRM_CRITICAL',
('REQUEST_RESTART'))
    Menuitem &Load Memory, Common.igseAPI.CMD('ALICE_CONFIRM_CRITICAL',
('LOAD_MEMORY'))
  }

  Popup &05 - Start Histogram (C)
  {
    Menuitem &Door open, Common.igseAPI.CMD('ALICE_START_HISTOGRAM', ('OPEN_DOOR',
'ACQUISITION'))
    Menuitem &Door closed, Common.igseAPI.CMD('ALICE_START_HISTOGRAM', ('CLOSE_DOOR',
'ACQUISITION'))
    Separator
    Menuitem &Test 1 (inc), Common.igseAPI.CMD('ALICE_START_HISTOGRAM', ('CLOSE_DOOR',
'TEST_INC'))
    Menuitem &Test 2 (dec), Common.igseAPI.CMD('ALICE_START_HISTOGRAM', ('CLOSE_DOOR',
'TEST_DEC'))
    Menuitem &Test 3 (con), Common.igseAPI.CMD('ALICE_START_HISTOGRAM', ('CLOSE_DOOR',
'TEST_CONST'))
    Menuitem &Test 4 (non), Common.igseAPI.CMD('ALICE_START_HISTOGRAM', ('CLOSE_DOOR',
'TEST_LEAVE'))
    Menuitem &Test 5 (his), Common.igseAPI.CMD('ALICE_START_HISTOGRAM', ('CLOSE_DOOR',
'TEST_HISTO'))
    Menuitem &Test 6 (pix), Common.igseAPI.CMD('ALICE_START_HISTOGRAM', ('CLOSE_DOOR',
'TEST_PIXEL'))
  }
}
```



## 1.8 TLM Module Generation

The TLM conversion generates several output file to configure GSEOS. The rest of this chapter details how the TLM database is mapped into the various output files.

The purpose of this package is to import a telemetry database specified in an Excel spreadsheet into several GSEOS configuration files.

The source spreadsheet consists of several worksheets. Each worksheet contains different parameters of the telemetry point specification.

### Block file generation

One of the output files is a block definition file. Since a spreadsheet contains a description of a CCSDS packet for a specific ApId only one block is generated.

The block file and the block name can be configured in a configuration file. If no file name or block name are specified the defaults are:

ApIdXXX (that is ApIdXXX.blk for the block file and ApIdXXX for the block name, where XXX is the hexadecimal representation of the ApId).

The "ApId"\_LOCA.CSV worksheet contains all the information needed for the block definition: Mnemonic, Start Byte, Start Bit, Data Size. These values will be directly translated into the Gseos Block and Item definitions. All items will be scalar items since the spreadsheet does not provision for array items. Currently only items up to 32 bit are supported.

A custom TLM Decoder will decode the generic TLM packet into the according block depending on the ApId. This decoder is implemented by the instrument teams as necessary.

GSEOS data items are not typed. A display format can be chosen when the data item is placed on a screen. The Type column is ignored for the block definition generation.

Block definitions can be customized. The [configuration file](#) allows you to specify additional items in the generated block.

### Conversion

The conversion column lets you specify a conversion from raw value to engineering value as well as display format.

The conversion for a telemetry point can be found on the "ApId"\_BASI.CSV sheet.

In GSEOS the display format is chosen when an item is placed on a screen. Therefore the settings: DEC, HEX, FLOAT, STRING are ignored. The settings LINEAR, STATE, and POLY are used to generate specific conversions that can be used when the item is displayed. LINEAR and POLY get converted into conversion functions with the same names. These conversion functions are specific for a data item and are defined in a .qlf formula file. The name of the formula file can be specified in the configuration file, if no file name is specified it defaults to the one specified for the block file with the extension .qlf. If a block file name is not specified the file name defaults to ApIdXXX.qlf.

The LINEAR calibration points are located on the "ApId"\_CALP.CSV sheet. For the LINEAR transformation the columns "Pair Num", "Cal Pair Raw (1-16)", and "Cal Pair Eng (1-16)" are used to generate the proper formula.

The POLY coefficients are on the "ApId"\_COEF.CSV sheet. For the POLY transformation the columns "C0", ... , "C7" are used to generate the proper formula.

When placing a data item on a screen you can choose the conversion function LINEAR or POLY. If you want to display the raw value choose 'No Conversion'.

You can also use these conversion functions from any Python script. They are located in the module GseosConversion. In order to invoke the linear conversion for ALICE\_HVPS\_SET\_VOLT you would specify the following code:

```
GseosConversion.fLinearConversion('ApId482.ALICE_HVPS_SET_VOLT')
```

The STATE conversion is performed with text reference files in GSEOS. A text reference file will be generated containing the according definitions. You then can place an item and select the style 'Casting' and select the appropriate text reference file. The name of the file generated can be configured in the configuration file and defaults to ApIdXXX.tr.

The STATE conversion uses the "State Low(M)", "State High(M)", and "State Msg(M)" columns to generate the text reference file.

### STATE conversion

The STATE message as defined on the STATE sheet can be accessed on display screens and from STOL. The ExcelLoad conversion will generate a text reference file (.tr) that contains all state messages including color codes. When you display an item and you choose the 'Text Reference' display option you can choose among the text references that were defined by the conversion. A text reference is named after the telemetry mnemonic it refers to. Instead of the Block.Item syntax used for telemetry mnemonics the period is replaced by an underscore. This makes it easy to associate the proper state messages to the telemetry item you want to display.

If you reference the state message from STOL you have to use the STATE() function. The STATE() function takes the telemetry mnemonic name as it's argument, e.g.:

```
STATE('ALICE_HK.SAFETY_ACTIVE')
```

This will return the status string that was defined for the range the telemetry value falls into. If the value does not match any range an empty string is returned. You can also get the STATE value from a Python script. You would use the GseosConversion module function fStateConversion. The following example shows how to get to the STATE value of the ALICE\_HVPS\_STATUS item:

```
GseosConversion.fStateConversion('ApId482.ALICE_HVPS_STATUS')
```

### Color codes

Two custom columns can be defined on the STAT sheet. They allow you to define the foreground and background color of the status message when displayed on a screen. The column headers must be to the left of the last regular EPOCH column (Spacecraft). The column names are 'GSEOS Foreground Color' and 'GSEOS Background Color'.

The following strings are valid color names (the names are case insensitive):

BLACK  
BLUE  
GREEN  
CYAN  
RED  
MAGENTA  
BROWN

LIGHTGRAY  
 DARKGRAY  
 LIGHTBLUE  
 LIGHTGREEN  
 LIGHTCYAN  
 LIGHTRED  
 LIGHTMAGENTA  
 YELLOW  
 WHITE

See below for a screenshot of a sample color definition. The colors are only used for GSEOS and are automatically formatted into the generated .tr file. If no colors are defined the foreground defaults to Black and the background to White.

	J	K	L
2	Spacecraft	GSEOS Foreground Color	GSEOS Background Color
3	ALL	LightRed	Black
4	ALL	Blue	Yellow
5	ALL	White	Yellow
6	ALL	Blue	Yellow
7	ALL	Blue	Yellow
8	ALL	Red	Yellow
9	ALL	Red	Yellow
10	ALL	Magenta	Black
11	ALL	Magenta	Black
12	ALL	White	White
13	ALL	White	White

### Alarm Limits

Alarm limits are implemented with GSEOS Alarm limit definition files (\*.alarm). The alarm limits that are automatically generated are named:

- TelemetryPoint Alarm Yellow Low
- TelemetryPoint Alarm Red Low
- TelemetryPoint Alarm Yellow High
- TelemetryPoint Alarm Red High

The Alarm limit file is generated as ApIdXXX.alarm by default and can be set with the [TLM configuration file](#). The following columns are used to create the alarm limit definitions: "Yellow Low Limit", "Yellow High Limit", "Red Low Limit", "Red High Limit".

### Limitations

The following entries are currently not supported and are ignored:

- Items larger than 32-bit.
- Scenarios
- Spacecraft
- Invert bit (IB)
- Reverse bit (RB)
- Any data type other than SIGNED, and UNSIGNED
- Context points

## 1.9 CMD Configuration File

The CMD configuration file allows to specify conversion options for the command ExcelLoad process. The configuration file must have the same file name as the Excel spreadsheet file but with a .config extension instead of .xls. The date and version part of the file name are optional for the configuration file, that is it will match all spreadsheets independent of the data and version number if you don't specify this for the configuration file.

The configuration file is a XML document. All elements must be embedded in the following root node:

```
<XML ID="GSEOS Conversion Configuration file">  
  
</XML>
```

The following paragraphs describe the data type definitions. A complete sample file can be found in the chapter: [Sample Configuration File](#).

### <ExcludeCommands>

This element is optional and there can be more than one <ExcludeCommands> element. The <ExcludeCommands> element contains <Command> elements with the attribute Mnemonic. The value of the Mnemonic attribute specifies the commands to be excluded from the command generation. You can specify any number of <Command> nodes within the <ExcludeCommands> element. The Mnemonic attribute values can use wildcards. A ? is a wildcard for any single character, the \* is a wildcard for any sequence of characters. These wildcards work identical to the DOS file system wildcards.

The following example excludes all commands that start with ALICE\_START and all commands that start with ALICE\_ and end in \_PARAMETER:

```
<ExcludeCommands>  
  <Command Mnemonic="ALICE_START*" />  
  <Command Mnemonic="ALICE_*_PARAMETER*" />  
</ExcludeCommands>
```

### <MenuConversion>

This element is optional and lets you configure the menu generation process. The <Strip> element allows you to strip off certain parts of the item mnemonic. The Prefix attribute strips off the specified prefix if the command mnemonic starts with that prefix. There can be multiple <Strip> elements.

The following example strips the LORRI\_ prefix from all menu entries (popup and menuitem) that start with LORRI\_:

```
<MenuConversion>
  <Strip Prefix = "LORRI_" />
</MenuConversion>
```

### **<EnterMacroMode>**

The <EnterMacroMode> element lists all the command mnemonics that put the system in macro mode. This element is optional and there can be more than one <EnterMacroMode> element. The <EnterMacroMode> element contains <Command> elements with the attribute Mnemonic. The value of the Mnemonic attribute specifies the commands that switch macro mode on. You can specify any number of <Command> nodes within the <EnterMacroMode> element. The Mnemonic attribute values can contain wildcards. Please refer to the <ExcludeCommands> element for the use of wildcards.

The following example tells the conversion routine to enter macro mode when the ALICE\_MAC\_DEF\_BEGIN command is encountered:

```
<EnterMacroMode>
  <Command Mnemonic="ALICE_MAC_DEF_BEGIN" />
</EnterMacroMode>
```

### **<LeaveMacroMode>**

The <LeaveMacroMode> element lists all the command mnemonics that make the system leave macro mode. This element is used in conjunction with the <EnterMacroMode> element to define all the commands that control the macro mode behavior. The <LeaveMacroMode> element is optional and there can be more than one <LeaveMacroMode> element. It contains <Command> elements with the attribute Mnemonic. The value of the Mnemonic attribute specifies the commands that turn macro mode off. You can specify any number of <Command> nodes within the <LeaveMacroMode> element. The Mnemonic attribute values can contain wildcards. Please refer to the <ExcludeCommands> element for the use of wildcards.

The following example tells the conversion routine to exit macro mode when the ALICE\_MAC\_DEF\_END command is encountered:

```
<LeaveMacroMode>
  <Command Mnemonic="ALICE_MAC_DEF_END" />
</LeaveMacroMode>
```

### **<CommandMenu>**

The <CommandMenu> element allows you to control the layout of your command menus. You can specify the grouping of commands into submenus or even individually place commands. You can take control of the placement of commands as well as add your own menu items, popups, and separators. Each <CommandMenu> entry creates a new top level menu. You can specify a separate command menu file with each <CommandMenu> element or you can place multiple menus into one and the same command menu file. If no FileName attribute is specified the menu will be appended to the default command menu file, the default command menu file has the same name as the <CommandModule> entry

with the extension '.cm'. The menu name is set with the Menu attribute of the <CommandMenu> element. See the example below:

```
<CommandMenu FileName="AliceCmd1.cm" Menu="Alice Commands">
  ...
  ...
  ...
</CommandMenu>
```

The following elements are allowed within a <CommandMenu> element: <Popup>, <MenuItem>, <Command>, <Separator>. These elements are explained below.

### <Command>

The <Command> element specifies the commands to be imported into this menu. The Mnemonic attribute specifies the command mnemonic you want to import. You can use the wildcards ? and \* to match multiple commands. However, if a single command from the Excel spreadsheet matches multiple commands it will only be imported the first time it matches. This allows you to specify a catch all at the end of your menu definition:

```
...
...
...
<Command Mnemonic="*" />
</CommandMenu>
```

The above will list all the commands that have not been matched so far. As a matter of fact, if you don't specify a <CommandMenu> at all the system will use the following default:

```
<CommandMenu Menu="CommandModule">
  <Command Mnemonic="*" />

</CommandMenu>
```

<Command> elements can only be placed within <CommandMenu> or <Popup> elements. They can't contain any elements and are therefore the leaf nodes of the <CommandMenu> element.

### <Popup>

The <Popup> element allows you to create your own nesting levels. <Popup> elements can contain the same elements as the <CommandMenu> element. The <Popup> element has one mandatory attribute: Name. Name specifies the name of the Popup.

```
<Popup Name="Heater Commands">
  ...
  ...
  ...
</Popup>
```

### <MenuItem>

The <MenuItem> element lets you insert your own commands besides the ones found in the Excel spreadsheet definition file. The <MenuItem> has two attributes: Name, which specifies the menu name and Command, which specifies the command string.

<MenuItem> elements can only be placed within <CommandMenu> or <Popup> elements.

```
<MenuItem Name="Heater On" Command="ALICE_HTR_ON()" />
```

### <Separator>

The <Separator> element inserts a separator in the menu. <Separator> elements are valid anywhere within a <CommandMenu> or <Popup> element.

```
<Separator/>
```

## 1.9.1 Sample Configuration File

```
<XML ID="GSEOS Conversion Configuration file">

  <!-- ===== -->
  <!-- <ExcludeCommands> -->
  <!-- Any commands listed in the ExcludeCommands section will not be -->
  <!-- converted. You can use the wildcard characters ? and * to match -->
  <!-- multiple commands. -->
  <!-- ===== -->
  <ExcludeCommands>
    <Command Mnemonic="ALICE_START*" />
    <Command Mnemonic="ALICE_*_PARAMETER*" />
  </ExcludeCommands>

  <ExcludeCommands>
    <Command Mnemonic="ALICE_*ACTIVATE*" />
    <Command Mnemonic="ALIC?_NOP" />
  </ExcludeCommands>

  <!-- ===== -->
  <!-- <MenuConversion> -->
  <!-- This element lets you configure the menu generation process. -->
  <!-- The <Strip> element allows you to strip off certain parts of the -->
  <!-- item mnemonic. The Prefix attribute strips off the specified prefix -->
  <!-- if the command mnemonic starts with that prefix. -->
  <!-- -->
  <!-- ===== -->
  <MenuConversion>
    <Strip Prefix = "LORRI_" />
  </MenuConversion>

  <!-- ===== -->
  <!-- <EnterMacroMode> -->
  <!-- Any commands listed in the EnterMacroMode element will cause the -->
  <!-- system to enter macro mode. You can use the wildcard characters ? -->
  <!-- and * to match multiple commands. -->
  <!-- ===== -->
  <EnterMacroMode>
    <Command Mnemonic="ALICE_MAC_DEF_BEGIN" />
```

```

</EnterMacroMode>

<!-- ===== -->
<!-- <LeaveMacroMode> -->
<!-- Any commands listed in the EnterMacroMode element will cause the -->
<!-- system to leave macro mode. You can use the wildcard characters ? -->
<!-- and * to match multiple commands. -->
<!-- ===== -->
<LeaveMacroMode>
  <Command Mnemonic="ALICE_MAC_DEF_END" />
</LeaveMacroMode>

<!-- ===== -->
<!-- <CommandMenu> -->
<!-- A CommandMenu allows you to customize the command menu generation -->
<!-- process. You can take control of the placement of commands as well -->
<!-- as add your own Menuitems, Popups, and Separators. -->
<!-- with the same name as the CommandModule plus a .cm extension. -->
<!-- Each CommandMenu entry creates a new top level Menu. You can specify -->
<!-- a separate file. If no file name is specified the menu will be -->
<!-- appended to the default command menu file. The FileName attribute -->
<!-- of the CommandMenu sets the file name. You have to specify the menu -->
<!-- name with the Menu attribute. -->
<!-- e.g.: <CommandMenu FileName="AliceCmd1.cm" Menu="Alice Commands"> -->
<!-- -->
<!-- Within a <CommandMenu> element you can place <Popup>, <Menuitem>, -->
<!-- <Command>, and <Separator> elements. These elements are explained -->
<!-- below. -->
<!-- -->
<!-- <Command> -->
<!-- The <Command> element specifies the commands to be imported into -->
<!-- this menu. You can use the wildcards ? and * to match multiple -->
<!-- commands. However, if a single command from the Excel spreadsheet -->
<!-- matches multiple commands it will only be imported the first time -->
<!-- it matches. -->
<!-- This allows you to specify a catch all at the end of your menu -->
<!-- definition: -->
<!-- -->
<!--   <Command Mnemonic="*" /> -->
<!-- </CommandMenu> -->
<!-- -->
<!-- This will list all the commands that have not been matched so far. -->
<!-- As a matter of fact, if you don't specify a CommandMenu at all the -->
<!-- system will use the following default: -->
<!-- -->
<!--   <CommandMenu Menu="CommandModule"> -->
<!--     <Command Mnemonic="*" /> -->
<!--   </CommandMenu> -->
<!-- -->
<!-- Command elements can only be placed within CommandMenu or Popup -->
<!-- elements. -->
<!-- -->
<!-- <Popup> -->
<!-- The Popup element allows you to create your own nesting levels. -->
<!-- Popup elements can contain the same elements as the CommandMenu -->
<!-- element. The popup element has one mandatory attribute: Name. -->
<!-- Name specifies the name of the Popup. -->
<!-- e.g.: <Popup Name="Heater Commands"> -->
<!-- -->
<!-- <Menuitem> -->
<!-- The Menuitem element lets you insert your own commands besides the -->

```



```

<!-- ones found in the Excel spreadsheet definition file. -->
<!-- The MenuItem has two attributes: Name, which specifies the menu -->
<!-- name and Command, which is the command string. -->
<!-- e.g.: <MenuItem Name="Heater On" Command="ALICE_HTR_ON()" /> -->
<!-- -->
<!-- MenuItem elements can only be placed within CommandMenu or Popup -->
<!-- elements. -->
<!-- -->
<!-- <Separator> -->
<!-- The Separator element inserts a separator in the menu. Separator -->
<!-- elements are valid anywhere within a CommandMenu or Popup element. -->
<!-- e.g.: <Separator/> -->
<!-- ===== -->

<CommandMenu Menu="Alice 1" FileName="Alice.cm">
  <Popup Name="Heater Commands">
    <MenuItem Name="Heater On" Command="ALICE_HEATER_ON()" />
    <Separator/>
    <MenuItem Name="Heater Off" Command="ALICE_HEATER_OFF()" />

    <Popup Name="More Heater Commands">
      <MenuItem Name="Heater On" Command="ALICE_HEATER_ON()" />
      <MenuItem Name="Heater Off" Command="ALICE_HEATER_OFF()" />
    </Popup>
  </Popup>

  <Popup Name="Alice Commands...">
    <Command Mnemonic="ALICE_C*" />
    <Popup Name="Check nesting...">
      <Popup Name="The rest...">
        <Command Mnemonic="*" />
      </Popup>
    </Popup>
  </Popup>

  <MenuItem Name="No Operation" Command="ALICE_NOP()" />
</CommandMenu>

<CommandMenu Menu="Alice Power">
</CommandMenu>

<CommandMenu Menu="Alice Heaters" FileName="Alice1.cm">
</CommandMenu>

<CommandMenu Menu="Alice Set Parameters" FileName="Alice1.cm">
</CommandMenu>

</XML>

```

## 1.10 TLM Configuration File

The CMD configuration file allows to specify conversion options for the TLM ExcelLoad process. The configuration file must have the same file name as the Excel spreadsheet file but with a .config extension instead of .xls. The date and version part of the file name are

optional for the configuration file, that is it will match all spreadsheets independent of the data and version number if you don't specify this for the configuration file.

The configuration file is a XML document. All elements must be embedded in the following root node:

```
<XML ID="GSEOS Conversion Configuration file">

</XML>
```

The following paragraphs describe the data type definitions. A complete sample file can be found here: [TLM Sample Configuration File](#).

### **<ExcludeTLM>**

This element is optional and there can be more than one <ExcludeTLM> element. The <ExcludeTLM> element contains <Item> elements with the attribute Mnemonic. The value of the Mnemonic attribute specifies the TLM mnemonic to be excluded from the TLM generation. You can specify any number of <Item> nodes within the <ExcludeTLM> element. The Mnemonic attribute values can use wildcards. A ? is a wildcard for any single character, the \* is a wildcard for any sequence of characters. These wildcards work identical to the DOS file system wildcards.

The following example excludes all items that start with ALICE\_HV\_ and all commands that start with ALICE\_ and end in \_PARAMETER:

```
<ExcludeTLM>
  <Command Mnemonic="ALICE_HV*" />
  <Command Mnemonic="ALICE_*_PARAMETER*" />
</ExcludeTLM>
```

### **<BlockConversion>**

This element lets you configure the block generation process. The <Strip> element allows you to strip off certain parts of the item mnemonic. The Prefix attribute strips off the specified prefix if the mnemonic starts with that prefix. If all your mnemonics start with ALICE\_ you can strip off the prefix ALICE\_ and the generated block will contain the names without this prefix. Use this feature with caution since the STOL command might not understand mnemonics modified in this way!

```
<BlockConversion>
  <Strip Prefix="ALICE_" />
</BlockConversion>
```

### **<Names>**

The Names element lets you specify various names you want to use instead of the default ones. The default file names for most modules is ApIdXXX.ext where XXX is the hexadecimal ApId and ext is the appropriate extension (.blk, .qlf, .tr).

The following conversion names can be set:

### **<BlockFileName>**

This element allows you to specify the block file name of the block definition file that is

generated. The default is ApIdXXX.blk for the block file name, where XXX is the ApId in hexadecimal format.

#### <BlockName>

This element allows you to specify the block name of the block that is defined in the block definition file. The default is ApIdXXX, where XXX is the ApId in hexadecimal format.

#### <FormulaFileName>

This element allows you to specify the formula file name of the block definition file that is generated. The default is the same as specified for <BlockFileName>, except with the extension .qlf. If <BlockFileName> is not specified it defaults to ApIdXXX.qlf.

```
<Names>
  <BlockFile    Name="MDS_StatusFile" />
  <Block        Name="MDS_Status" />
  <FormulaFile  Name="MDS_Formulas" />
</Names>
```

### 1.10.1 TLM Sample Configuration File

```
<XML ID="GSEOS Conversion Configuration file">

  <!-- ===== -->
  <!-- <ExcludeTLM> -->
  <!-- Any TLM items listed in the ExcludeTLM section will not be -->
  <!-- converted. You can use the wildcard characters ? and * to match -->
  <!-- multiple TLM items. -->
  <!-- ===== -->
  <ExcludeTLM>
    <Item Mnemonic="XALICE_START*" />
    <Item Mnemonic="XALICE_*_PARAMETER*" />
  </ExcludeTLM>

  <ExcludeTLM>
    <Item Mnemonic="XALICE_*ACTIVATE*" />
    <Item Mnemonic="XALIC?_NOP" />
  </ExcludeTLM>

  <!-- ===== -->
  <!-- <Names> -->
  <!-- -->
  <!-- The Names element lets you specify various names you want to -->
  <!-- use instead of the default ones. -->
  <!-- -->
  <!-- <BlockFileName> -->
  <!-- This element allows you to specify the block file name of the block -->
  <!-- definition file that is generated. -->
  <!-- The default is ApId_XXX for the block name, where XXX is the ApId -->
  <!-- in hexadecimal format. -->
  <!-- -->
  <!-- <BlockName> -->
  <!-- This element allows you to specify the block name that is generated. -->
  <!-- The default is ApId_XXX for the block name, where XXX is the ApId -->
```

```

<!-- in hexadecimal format. -->
<!-- -->
<!-- <FormulaFileName> -->
<!-- This element allows you to specify the formula file name of the -->
<!-- formula file that is generated. -->
<!-- The default is the same as specified for <BlockFileName>, except -->
<!-- with the extension .qlf. -->
<!-- If <BlockFileName> is not specified it defaults to ApId_XXX.qlf. -->
<!-- ===== -->
<Names>
  <BlockFile Name="MDS_StatusFile" />
  <Block Name="MDS_Status" />
  <FormulaFile Name="MDS_Formulas" />
</Names>

<!-- ===== -->
<!-- <BlockConversion> -->
<!-- This element lets you configure the block generation process. -->
<!-- The <Strip> element allows you to strip off certain parts of the -->
<!-- item mnemonic. The Prefix attribute strips off the specified prefix -->
<!-- if the mnemonic starts with that prefix. -->
<!-- ===== -->
<BlockConversion>
  <Strip Prefix="ALICE_" />
</BlockConversion>

</XML>

```