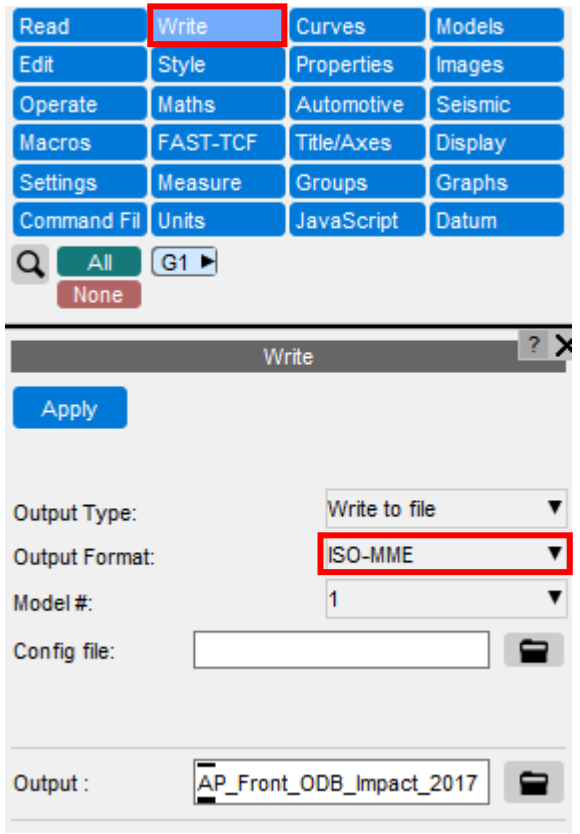
Oasys
T/HIS

Writing ISO-MME Files

Writing ISO-MME Files

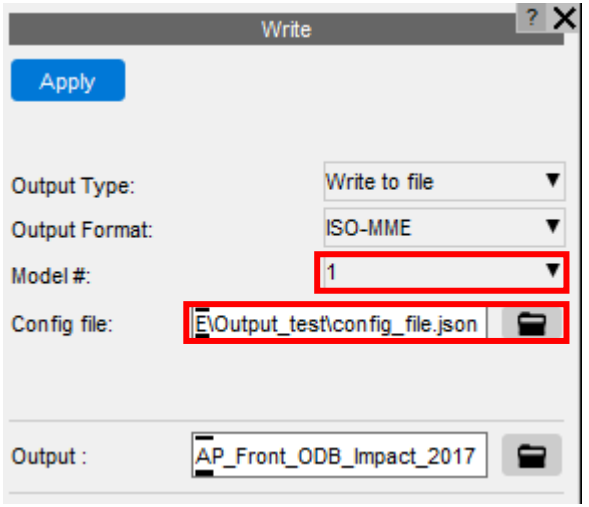
The ISO-MME format (ISO/TS 13499) is a data exchange format for crash analyses comprising a number of folders and files.

The option to write data in this format is included in T/HIS, but the method for doing so differs slightly to other write formats, so this tutorial shows how to use it.



Writing ISO-MME Files

To write ISO-MME files, instead of selecting curves that you want to write out, you have to select a model for T/HIS to extract the data from and provide a configuration file to specify what data should be written out.



The configuration file is needed because the naming conventions of the output files, set out in the ISO standard, follow specific rules which require extra data that isn't present in the curves alone and they also contain lines at the top of the files which describe the data in more detail.

Writing ISO-MME Files

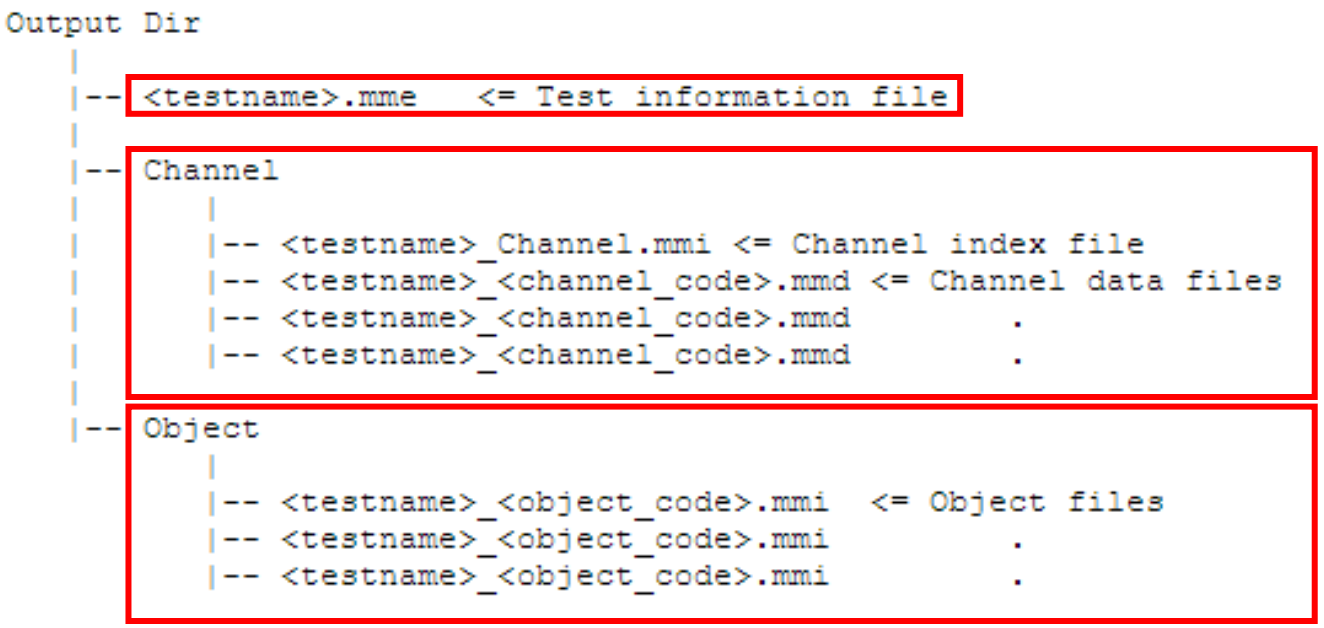
The ISO-MME configuration file is a JSON format file used to tell T/HIS what curves to generate, the data to extract, from which entities to extract it from, their locations in the model and whether any filtering is required.

This tutorial will show what needs to be written to this configuration file, but before that it will be helpful to show the structure of the directory where the various ISO-MME files get written out and the files it contains.

Overview

Writing ISO-MME Files

The ISO-MME format contains the following folders and files.



Details of each file are described in the following pages.

Writing ISO-MME Files

The test information file *<testname>.mme,* describes the test and the objects (vehicles, dummies, barriers) in it. As an example:

```
Data format edition number  :2.0
Timestamp                   :2020-11-10
Laboratory name             :Arup
Laboratory contact name     :NOVALUE
Laboratory contact phone    :NOVALUE
Laboratory contact fax      :NOVALUE
Laboratory contact email    :NOVALUE
Laboratory test ref number  :NOVALUE
Type of the test            :NOVALUE
Subtype of the test         :NOVALUE
Regulation                  :NOVALUE
Date of the test            :NOVALUE
Number of test objects      :3
#Begin of testobject
Type                        :D
Filename                    :my_test_D0.mmi
#End of testobject
#Begin of testobject
Type                        :1
Filename                    :my_test_1.mmi
#End of testobject
#Begin of testobject
Type                        :B
Filename                    :my_test_B.mmi
#End of testobject
```

Writing ISO-MME Files

The object files <*testname*>_<*object_code*>*.mmi,* describe each object (vehicles, dummies, barriers) in the test. As an example:

```
Name                              :H350
Velocity                          :NOVALUE
Mass                              :NOVALUE
Impact side                       :00
#Begin of biomechanical
Gender                            :male
Age                               :21
#End of biomechanical
```

Writing ISO-MME Files

The channel index file <*testname*>_*Channel.mmi,* lists the channel data files. As an example:

```
Number of channels          :3
Data origin                 :S
Data source                 :simulation
#Begin of channel
Extended channel code       :D0HEADMI0000000B
#End of channel
#Begin of channel
Extended channel code       :D0CHSTMI0000000C
#End of channel
#Begin of channel
Extended channel code       :D0PELVMI0000000C
#End of channel
```

Writing ISO-MME Files

The channel data files <*testname*>_<*channel_code*>*.mmd,* contain the time series data.

As an example:

The channel data files contain the time series data:

```
Data structure              :Channel
Instrumentation standard    :NOVALUE
Name of the channel         :Accel x - Node 52503304 : (PelvisAccel_INJURY) (Reg 0.100E-03) (C 180)
Data source                 :simulation
Data status                 :ok
Cut off frequency           :NOVALUE
Channel amplitude class     :NOVALUE
Sampling interval           :0.0001
Bit resolution              :NOVALUE
Time of first sample        :0
Number of samples           :1500
Reference channel           :implicit
#Start of data
-6.09125e-05
-1785.28
-3315.55
    .
    .
#End of data
```

Writing ISO-MME Files

There are a number of options that can be set in the configuration file, but only a limited number of them are required.

The next page shows an example file using all the available options.

Writing ISO-MME Files

```
{
    "testName": "my_first_test",
    "timestamp": "2020-11-10",
    "laboratoryName": "Arup",
    "laboratoryContactPhone": "0123456789",
    "laboratoryContactFax": "9876543210",
    "laboratoryContactEmail": "abc@def.com",
    "laboratoryTestRef": "xyz",
    "typeOfTest": "FrontalImpact",
    "subtypeOfTest": "0 Degree Empty",
    "regulation": "EuroNCAP",
    "testDate": "2020-05-10",

    "vehicles": [
        {
            "name": "vehicle1",
            "velocity": 177,
            "mass": 100,
            "impactSide": "FR",

            "channels": [
                {
                    "mainLocation": "FOWE",
                    "fineLocation1": "01",

                    "source": "lsda",
                    "entityType": "node",
                    "id": "10175835",
                    "filter": "C60",
                    "component": "displacement x"
                },

                {
                    "mainLocation": "FOWE",
                    "fineLocation1": "02",

                    "source": "lsda",
                    "entityType": "node",
                    "id": "10180679",
                    "component": "displacement y"
                },
            ]
        }
    ],
```

```
"dummies": [
    {
        "name": "Driver",
        "gender": "male",
        "age": 21,
        "position": "3",

        "channels": [
            {
                "mainLocation": "PELV",
                "fineLocation1": "MI",
                "fineLocation2": "OU",
                "direction": "Y",

                "source": "lsda",
                "entityType": "beam",
                "id": "55501787",
                "filter": "C180",
                "component": "moment y"
            }
        ]
    }
],

"barriers": [
    {
        "name": "ODB",
        "barrierWidth": 1.5,
        "barrierHeight": 1.0,
        "yawAngle": 0.0,

        "channels": [
            {
                "mainLocation": "FBAR",
                "fineLocation1": "01",

                "source": "thf",
                "entityType": "node",
                "id": "92222096",
                "filter": "C600",
                "component": "displacement x"
            }
        ]
    }
],
```

```
    "mobileBarriers": [
        {
            "name": "MPDB",
            "barrierWidth": 1.5,
            "barrierHeight": 1.0,
            "yawAngle": 0.0,
        }
    ]
}
```

Writing ISO-MME Files

Descriptions for each option are described in the T/HIS manual.

The following shows the minimum set of options that need to be defined for T/HIS to be able to write the ISO-MME files.

Property	Description	Valid values
testName	Name of the test. This is used for the test directory name and in the filenames	Any value
mainLocation	Main location on the object. This is required for the channel code.	See the ISO Related Electronic Document B for valid values, e.g. “HEAD”, “CHST”
entityType	The entity type to extract data.	A FAST-TCF data extraction keyword, e.g. “node”, “beam”
id	The entity id to extract data from.	This can be the numerical id or a *DATABASE_HISTORY_ID name. In either case it should be specified as a string, e.g. "100", "my_node_id"
component	The component to read.	FAST-TCF data extraction component words, e.g. “displacement x”, “energy”

Example

Writing ISO-MME Files

So, to write out the ISO-MME files you will first need to create the configuration file. Currently this has to be done by hand in a text editor.

A simple example to write data for a node is shown below.

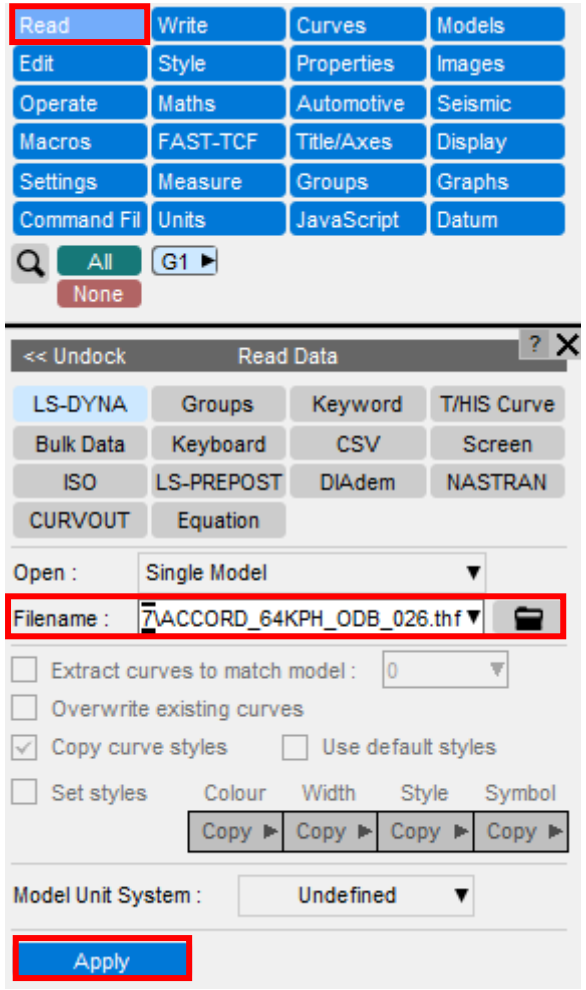
Copy and edit it, setting the “id” to a valid id in your model. You may also want to set the “mainLocation” value to something other than “PELV” depending on where the node is in the dummy.

```
{
    "testName": "my_first_test",

    "dummies": [
        {
            "channels": [
                {
                    "mainLocation": "PELV",

                    "entityType": "node",
                    "id": "55501787",
                    "component": "displacement x"
                }
            ]
        }
    ]
}
```

Save the file as ***config.json***

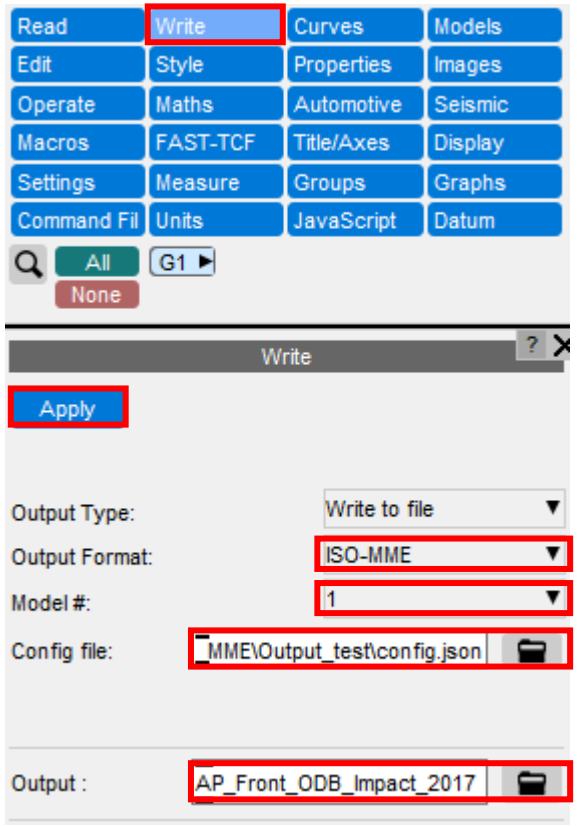
Writing ISO-MME Files

In T/HIS read in the model you want to write ISO-MME files for.



Writing ISO-MME Files

In the *Write* menu:



1. Set the output format to 'ISO-MME'
2. Select the model that you read in
3. Select the config file
4. Select a directory to output the files to
5. Press Apply

T/HIS will then generate the curves defined in the configuration file and write the ISO-MME files to the output directory.

Contact Information

ARUP

For more information please contact the following:

UK Contact:	China Contact:	India Contact:	USA West Contact:
T: +44 121 213 3399	T: +86 21 3118 8875	T: +91 40 69019723 / 98	T: +1 415 940 0959

or your local Oasys distributor