

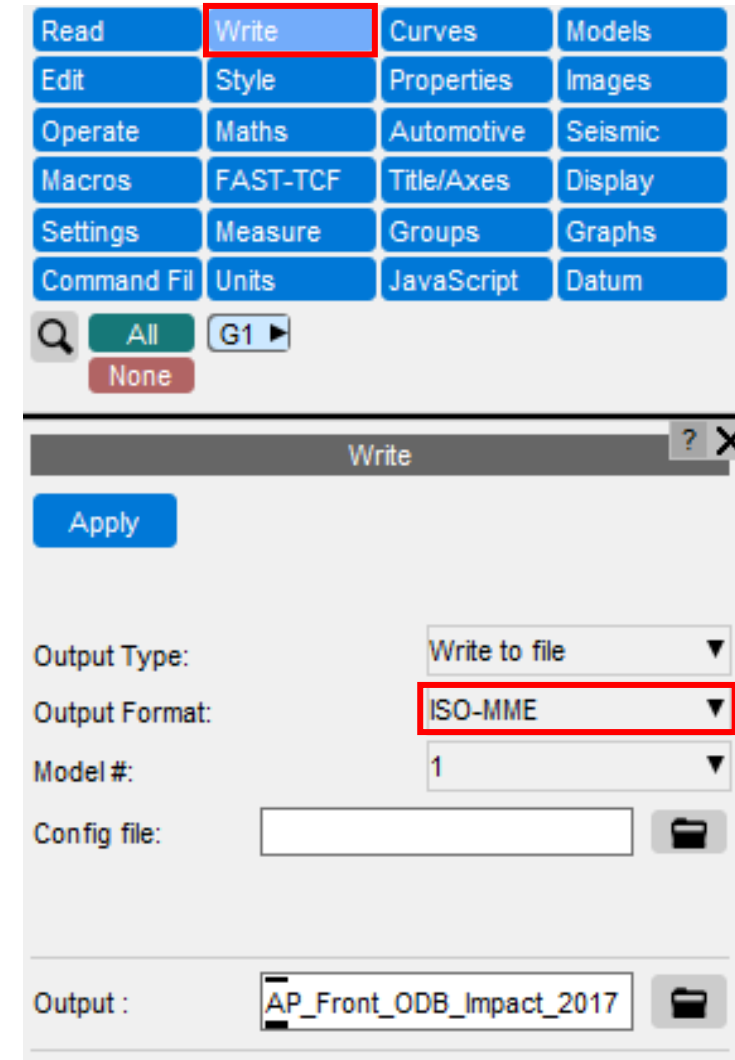
Writing ISO-MME Files



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.

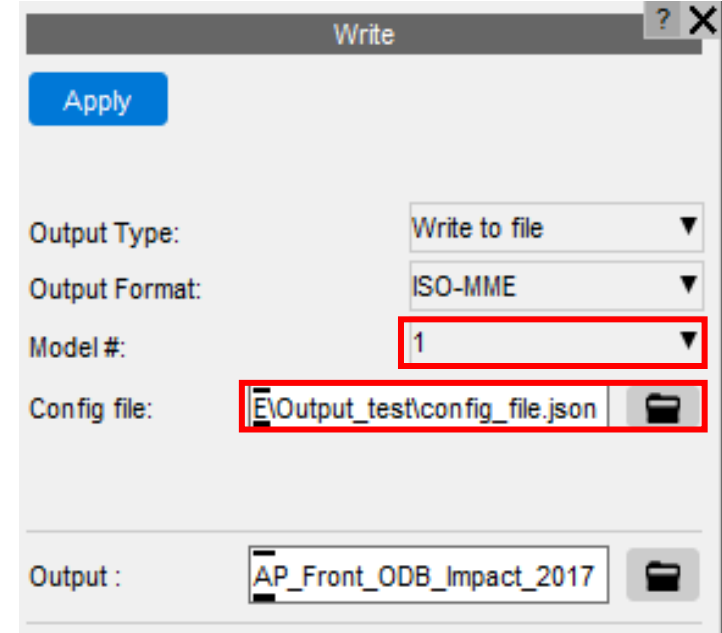


The screenshot displays the T/HIS software interface. At the top, a grid of blue buttons includes 'Read', 'Write' (highlighted with a red border), 'Curves', 'Models', 'Edit', 'Style', 'Properties', 'Images', 'Operate', 'Maths', 'Automotive', 'Seismic', 'Macros', 'FAST-TCF', 'Title/Axes', 'Display', 'Settings', 'Measure', 'Groups', 'Graphs', 'Command Fil', 'Units', 'JavaScript', and 'Datum'. Below this grid are search and filter buttons: a magnifying glass icon, 'All', 'G1', and 'None'. The 'Write' dialog box is open, showing an 'Apply' button at the top. The 'Output Type:' dropdown is set to 'Write to file'. The 'Output Format:' dropdown is set to 'ISO-MME' and is highlighted with a red border. The 'Model #' dropdown is set to '1'. The 'Config file:' field is empty with a folder icon to its right. The 'Output :' field contains the text 'AP_Front_ODB_Impact_2017' with a folder icon to its right.

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 or operation 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.

| ISO-MME Version 1.6 | ISO-MME Version 2.0 |
|---|---|
| <pre>Output Dir -- <Virtual Testing reference ID>.mme <= Test Data -- Channel -- <Virtual Testing reference ID>.chn <= Channel index file -- <Virtual Testing reference ID>.001 <= Channel data files -- <Virtual Testing reference ID>.002 . -- <Virtual Testing reference ID>.003 .</pre> | <pre>Output Dir -- <testname>.mme <= Test information file -- Channel -- <testname>_Channel.mmi <= Channel index file -- <testname>_<channel_code>.mmd <= Channel data files -- <testname>_<channel_code>.mmd . -- <testname>_<channel_code>.mmd . -- Object -- <testname>_<object_code>.mmi <= Object files -- <testname>_<object_code>.mmi . -- <testname>_<object_code>.mmi .</pre> |

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:

| ISO-MME Version 1.6 | ISO-MME Version 2.0 |
|--------------------------------------|-------------------------------------|
| Data format edition number :1.6 | Data format edition number :2.0 |
| Customer name :ARUP | Timestamp :2020-11-10 |
| Customer test ref. number :NOVALUE | Laboratory name :Arup |
| Customer project ref. number:9999 | Laboratory contact name :NOVALUE |
| Title :NOVALUE | Laboratory contact phone :NOVALUE |
| Timestamp :2022-09-29 | Laboratory contact fax :NOVALUE |
| 13:30:00 | Laboratory contact email :NOVALUE |
| Type of the test :NOVALUE | Laboratory test ref number :NOVALUE |
| Subtype of the test :NOVALUE | Type of the test :NOVALUE |
| Virtual Testing reference ID:NOVALUE | Subtype of the test :NOVALUE |
| Regulation :NOVALUE | Regulation :NOVALUE |
| Date of the test :2022-10-02 | Date of the test :NOVALUE |
| Name of test object 1 :NOVALUE | Number of test objects :2 |
| Ref. number of test object 1:NOVALUE | #Begin of testobject |
| Velocity test object 1 lon. :NOVALUE | Type :D |
| Velocity test object 1 lat. :NOVALUE | Filename :my_test_D0.mmi |
| Mass test object 1 :1230 | #End of testobject |
| Driver position object 1 :3 | #Begin of testobject |
| Impact side test object 1 :LE | Type :1 |
| Name of test object 2 :NOVALUE | Filename :my_test_1.mmi |
| Ref. number of test object 2:NOVALUE | #End of testobject |
| Velocity test object 2 lon. :NOVALUE | |
| Velocity test object 2 lat. :NOVALUE | |
| Mass test object 2 :1230 | |
| Driver position object 2 :2 | |
| Impact side test object 2 :LE | |
| Type of data source :Simulation | |

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
```

These files are only required in ISO-MME Version 2.0



Writing ISO-MME Files

The channel index file *<testname>_Channel.mmi* or *<testname>.chn*, lists the channel data files. As an example:

| ISO-MME Version 1.6 (.chn file) | | ISO-MME Version 2.0 (.mmi file) | |
|---------------------------------|-------------------|---------------------------------|-------------------|
| Instrumentation standard | : Not applicable | Number of channels | :3 |
| Number of channels | :7 | Data origin | :S |
| Name of channel 001 | :13HEAD0000WSACX0 | Data source | : simulation |
| Name of channel 002 | :13HEAD0000WSACY0 | #Begin of channel | |
| Name of channel 003 | :13HEAD0000WSACZ0 | Extended channel code | :D0HEADMI0000000B |
| Name of channel 004 | :13HEAD0000WSDCX0 | #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* or *<testname>.001*, contain the time series data. As an example:

| ISO-MME Version 1.6 (.001 file) | | ISO-MME Version 2.0 (.mmd file) | |
|---------------------------------|-------------------|---------------------------------|---|
| Test object number | :1 | Data structure | :Channel |
| Name of the channel | : | Instrumentation standard | :NOVALUE |
| Laboratory channel code | :NOVALUE | Name of the channel | :Accel x - Node 52503304 : (PelvisAccel_INJURY) (Reg 0.100E-03) (C 180) |
| Customer channel code | :NOVALUE | Data source | :simulation |
| Channel code | :13HEAD0000WSACX0 | Data status | :ok |
| Unit | :m / (s * s) | Cut off frequency | :NOVALUE |
| Reference system | :NOVALUE | Channel amplitude class | :NOVALUE |
| Transducer type | :NOVALUE | Sampling interval | :0.0001 |
| Pre-filter type | :NOVALUE | Bit resolution | :NOVALUE |
| Cut off frequency | :NOVALUE | Time of first sample | :0 |
| Channel amplitude class | :NOVALUE | Number of samples | :1500 |
| Sampling interval | :0.0001 | Reference channel | :implicit |
| Bit resolution | :NOVALUE | #Start of data | |
| Time of first sample | :NOVALUE | -6.09125e-05 | |
| Number of samples | :1999 | -1785.28 | |
| -0.42144 | | -3315.55 | |
| -0.00030 | | . | |
| -0.00028 | | . | |
| | | #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": "Far side",
  "timestamp": "12/18/2023, 4:24:25 PM",
  "formatVersion": "1.6",
  "laboratoryName": "Oasys LS-DYNA Environment",
  "customerName": "Euro NCAP",
  "customerTestRefNumber": "001",
  "customerProjectRefNumber": "1234",
  "virtualTestingReferenceId": "FS_Pole_75_X-ref_z-ref_50M_Sim_1",
  "typeOfTest": "SideImpact",
  "subtypeOfTest": "Far Side + VTC",
  "regulation": "EuroNCAP",
  "testDate": "12/18/2023",
  "unitsSystemModel": "U2",
  "unitsSystemDisplay": "U1",
  "descriptors": [
    {
      "description": "Type of data source",
      "value": "Simulation"
    },
    {
      "description": "Dummy simulation Model Specification",
      "value": "WSID 50 M v3.4.1"
    }
  ],
  "dummies": [
    {
      "testObject": "1",
      "position": "1",
      "vehicleTestObject": "1",
      "name": "PDB WSID 50M 4.0",
      "gender": "male"
    }
  ],
  "vehicles": [
    {
      "testObject": "1",
      "position": "0",
      "driverPosition": "1",
      "name": "Lighting McQueen",
      "refNumber": "NO VALUE",
      "mass": "NO VALUE",
      "impactSide": "RI",
      "descriptors": [
        {
          "description": "Vehicle Model",
          "value": "95"
        }
      ]
    }
  ]
},
```

```
"channels": [
  {
    "testObject": "1",
    "position": "1",
    "mainLocation": "ABRI",
    "fineLocation1": "RI",
    "fineLocation2": "01",
    "fineLocation3": "WS",
    "physicalDimension": "AN",
    "direction": "#",
    "entityType": "spring",
    "component": "rotation",
    "id": 10324,
    "write": false
  },
  {
    "testObject": "1",
    "position": "1",
    "mainLocation": "ABRI",
    "fineLocation1": "RI",
    "fineLocation2": "02",
    "fineLocation3": "WS",
    "physicalDimension": "AN",
    "direction": "#",
    "entityType": "spring",
    "component": "rotation",
    "id": 10325,
    "write": false
  },
  {
    "testObject": "0",
    "position": "0",
    "mainLocation": "EKIN",
    "fineLocation1": "SU",
    "fineLocation2": "00",
    "fineLocation3": "00",
    "direction": "0",
    "physicalDimension": "EN",
    "entityType": "whole",
    "filter": "C",
    "component": "kinetic",
    "write": true
  },
  {
    "testObject": "0",
    "position": "0",
    "mainLocation": "EHOU",
    "fineLocation1": "SU",
    "fineLocation2": "00",
    "fineLocation3": "00",
    "direction": "0",
    "physicalDimension": "EN",
    "entityType": "whole",
    "filter": "A",
    "component": "hourglass",
    "write": true
  }
],
```

```
"descriptors": [
  {
    "description": "Energy Type",
    "value": "Hourglass Energy"
  }
],
"operations": [
  {
    "operation": "sub",
    "input": [
      "11ABRIRI02WSAN#0",
      1.570796
    ],
    "output": "11ABRIRI02WSANZ0",
    "write": false
  },
  {
    "operation": "c180",
    "input": [
      "11ABRIRI02WSANZ0"
    ],
    "output": "11ABRIRI02WSANZC",
    "write": true,
    "descriptors": [
      {
        "description": "Abdomen Compression valid Value",
        "value": "46"
      },
      {
        "description": "Abdomen Compression Invalid Value",
        "value": "65"
      }
    ]
  }
]
}
```

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 test. | Any value |
| thisVersion | T/HIS Build Version | Any Numerical value. For Eg:- “21.0” |
| thisBuildNumber | T/HIS Build Number | Any numerical value. For Eg:- “6100” |

| Property | Description | Valid values |
|------------|--------------------------------------|--|
| testObject | testObject classification. | See the 'Test Object' section in the ISO Related Electronic Document B for valid values, e.g. "1", "2", “D” etc. |
| position | The dummy’s position in the vehicle. | See the ‘Position’ section in the ISO Related Electronic Document B for valid values, e.g. “1”, “2”. |

| Property | Description | Valid values |
|--------------|--|--|
| testObject | testObject classification. It is the first character in the ISO-MME channel code. | See the 'Test Object' section in the ISO Related Electronic Document B for valid values, e.g. "1", "2", “D” etc. |
| position | The dummy’s position in the vehicle. It is the second character in the ISO-MME channel code. | See the 'Position' section in the ISO Related Electronic Document B for valid values, e.g. "1", "2". |
| 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". |



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",
  "thisBuildNumber": "6500",
  "thisVersion": "21.0",

  "dummies": [
    {
      "testObject": "1",
      "position": "1"
    }
  ],
  "channels": [
    {
      "testObject": "1",
      "position": "1",
      "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.



| | | | |
|-------------|----------|------------|---------|
| Read | Write | Curves | Models |
| Edit | Style | Properties | Images |
| Operate | Maths | Automotive | Seismic |
| Macros | FAST-TCF | Title/Axes | Display |
| Settings | Measure | Groups | Graphs |
| Command Fil | Units | JavaScript | Datum |

<< Undock

Read Data

?

X

| | | | |
|-----------|------------|---------|-------------|
| LS-DYNA | Groups | Keyword | T/HIS Curve |
| Bulk Data | Keyboard | CSV | Screen |
| ISO | LS-PREPOST | DIAdem | NASTRAN |
| CURVOUT | Equation | | |

Open :

Single Model

Filename :

T/ACCORD_64KPH_ODB_026.thf

☐ Extract curves to match model :

0

☐ Overwrite existing curves

☒ Copy curve styles ☐ Use default styles

☐ Set styles

Colour

Width

Style

Symbol

Copy ▶

Copy ▶

Copy ▶

Copy ▶

Model Unit System :

Undefined

Apply

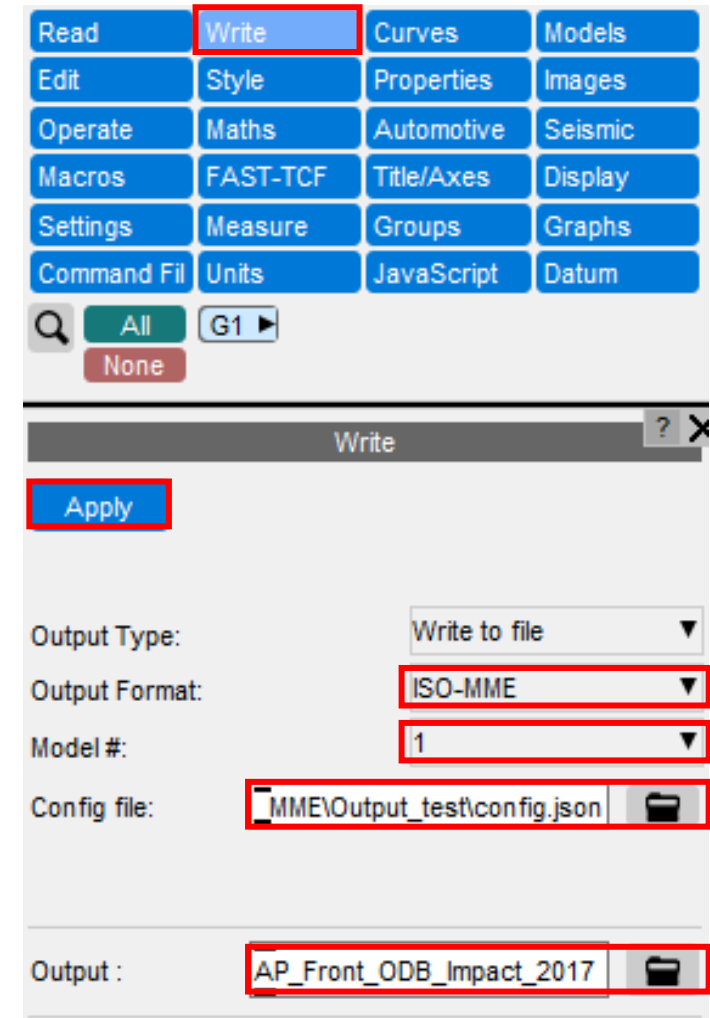


Writing ISO-MME Files

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 us

Global / UK

T: +44 121 213 3399

E: dyna.support@arup.com

India

T: +91 40 69019723 / 98

E: india.support@arup.com

China

T: +86 21 3118 8875

E: china.support@arup.com

USA

T: +1 415 940 0959

E: us.support@arup.com

Subscribe to
our newsletter:



Follow us on:



@Oasys LS-DYNA
Environment



@Oasys LS-DYNA
Environment



@Oasys



@Oasys

www.oasys-software.com/dyna/