

Catch The Errors with the AIMMS Error Handler



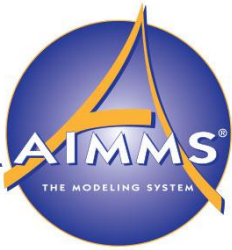
Haraldur Haraldsson

AIMMS Optimization Specialist



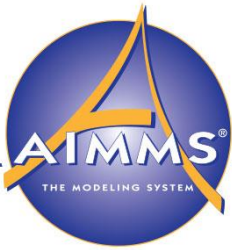
Objective of webinar

- Get you familiar with the error handler and how to use it effectively.
- Demonstrate how you can take control the behavior of AIMMS in case of an error using global and local error handlers.



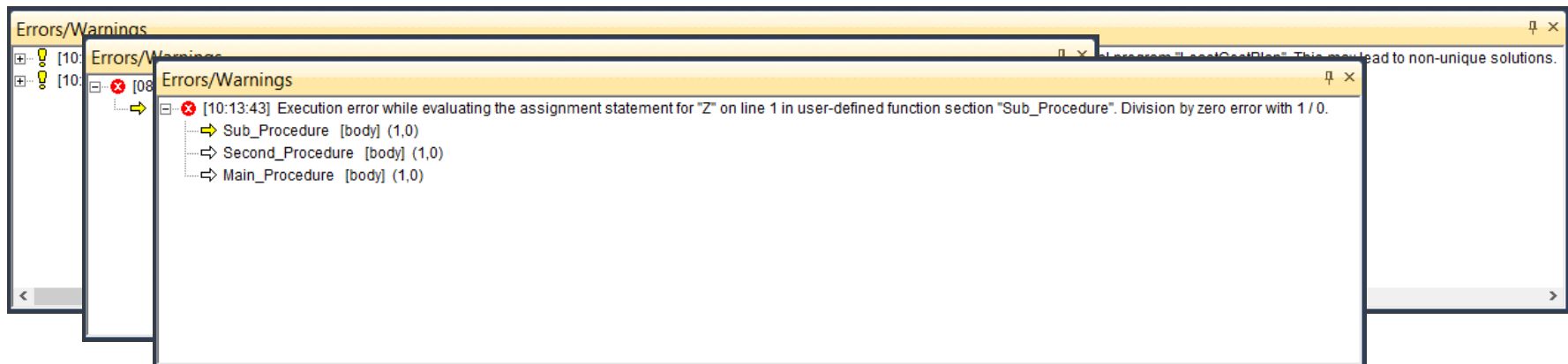
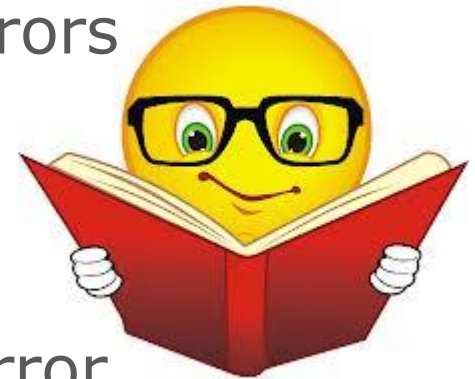
Today's Agenda

- Errors and Warnings
- Error Prevention
- Error Handler
- Global Error Handler
- Local Error Handler



Errors and Warnings

- Provide useful information about errors or warnings.
- Provide the location of the error or warning.
- In case of an execution error, the error message will track the execution stack.

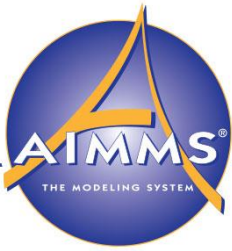




Error Prevention



- Allows you to check the model for preventable errors and stop the model execution if needed.
- Measures to check the input data before run.
- `FileExists (<filename>)`
 - `if not FileExists(ImportFile) then`
...
`endif;`
- `TestDataSource (<datasource>)`
 - `if not TestDataSource(ImportData) then`
...
`endif;`



Error Handler

- Allows you as a developer to take control over error messaging.
- You can modify the error message displayed.
- Give the end-user information on how to proceed.





Error Handler

- Improve end-user's experience.
- Report errors to the model developer.
- Guide to a fix.





Error Handler

Collection of identifiers that contain data of the error.

- `errh::PendingErrors`
- `errh::ErrorCodes`
- `errh::AllErrorCategories`
- `errh::AllErrorSeverities`



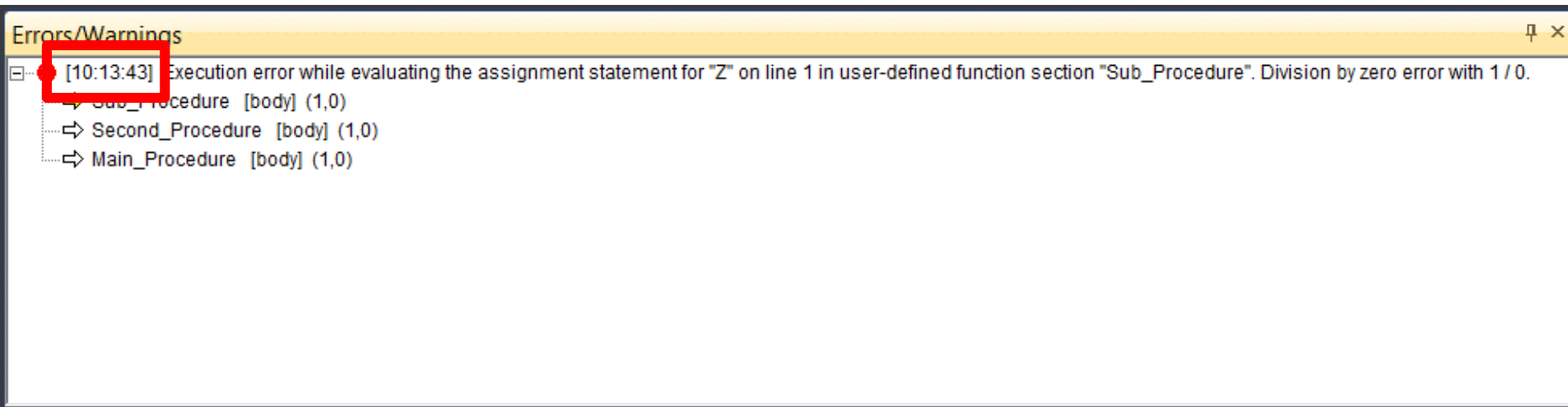
Error Handler

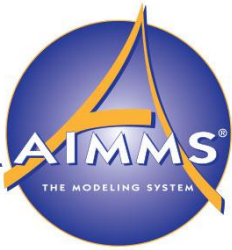
- AIMMS can call functions that display certain information from the error.
- Can be used to create a modified message to the end-user.



Error Handler

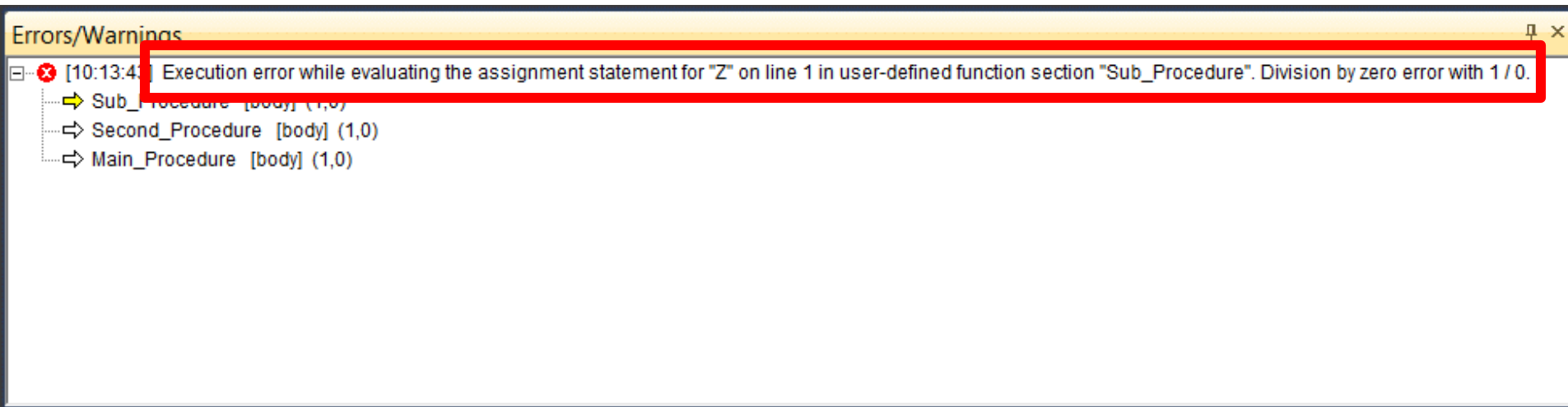
- `errh::CreationTime`
- Collects the time that the error occurred.

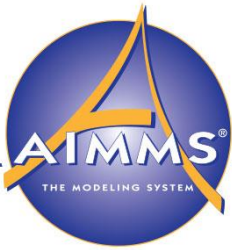




Error Handler

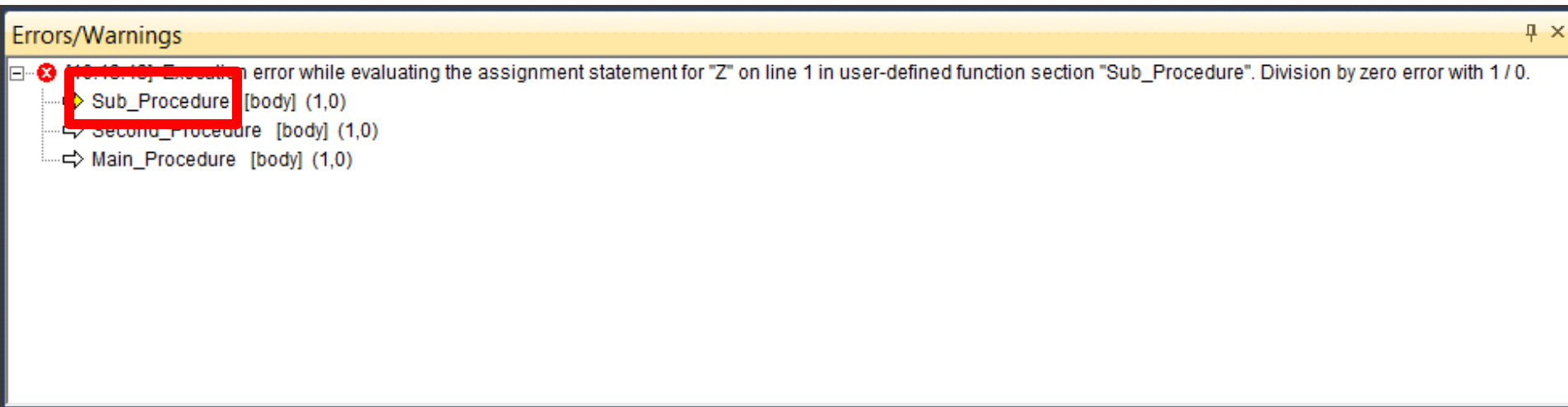
- `errh::Message`
- Collects the error message.





Error Handler

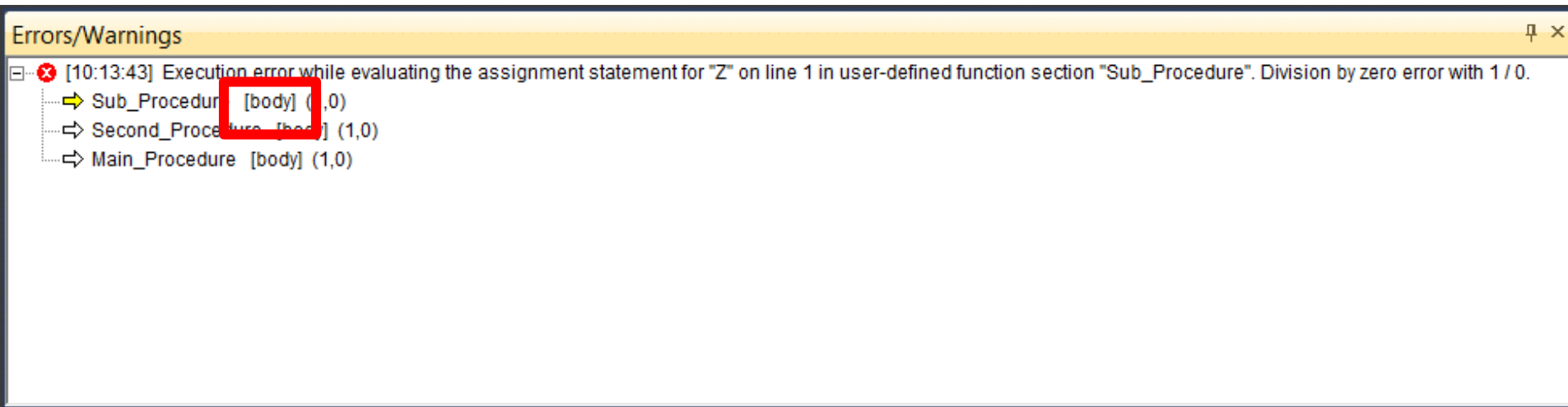
- `errh::Node`
- Collects the name of the identifier or node where the error occurred.





Error Handler

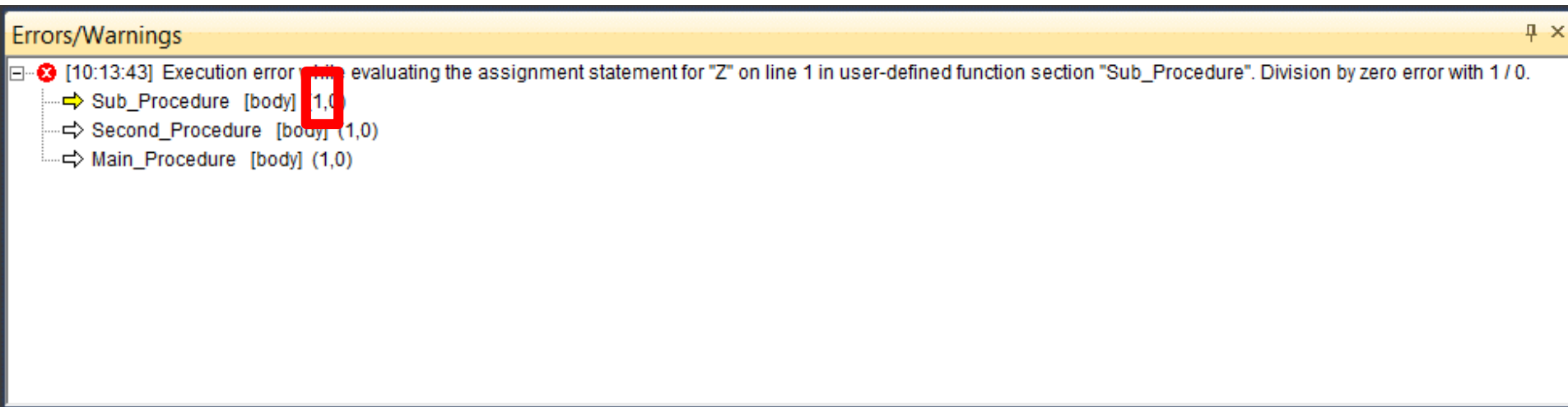
- `errh::Attribute`
- Collects the location in identifier or node where the error occurred.

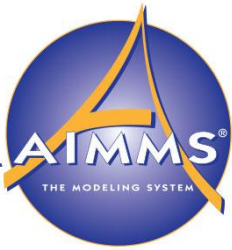




Error Handler

- `errh::Line`
- Collects the line number where the error occurred.





Error Handler

- `errh::Column`
- Collects the column number where the error occurred.

Errors/Warnings

[10:13:43] Execution error while evaluating the assignment statement for "Z" on line 1 in user-defined function section "Sub_Procedure". Division by zero error with 1 / 0.

- Sub_Procedure [body] (1,0)
- Second_Procedure [body] (1,0)
- Main_Procedure [body] (1,0)



Error Handler

- `errh::CreationTime`
- `errh::Message`
- `errh::Node`
- `errh::Attribute`
- `errh::Line`
- `errh::Column`



Error Handler

- `errh::CreationTime`
- `errh::Message`
- `errh::Node`
- `errh::Attribute`
- `errh::Line`
- `errh::Column`
- `errh::Category`
- `errh::Code`
- `errh::Filename`
- `errh::InsideCategory`
- `errh::IsMarkedAsHandled`
- `errh::Multiplicity`
- `errh::NumberOfLocations`
- `errh::Severity`



Error Handler

Two important functions

- `errh::MarkAsHandled`

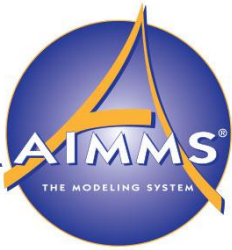
Marks the error as handled. Will not appear in the errors/warnings dialog.

- `errh::Adapt`

Allows you to change specific data for an error

- Severity
- Message
- Category
- Code

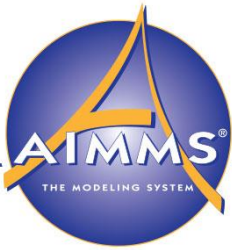




Global Error Handler

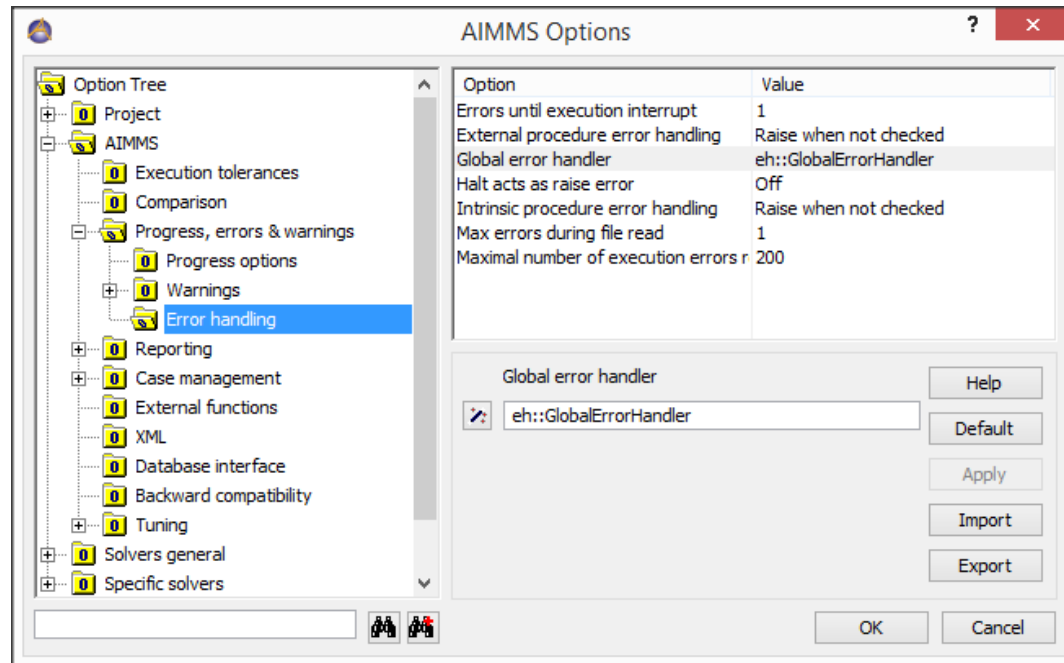
- The Global Error Handler treats runtime errors anywhere in your model.
- At the end of a finished execution run the Global error handler is called if there are any unhandled warning messages.
- If an error is raised the execution run is halted.





Global Error Handler

- Activated in the 'Project Options' dialog

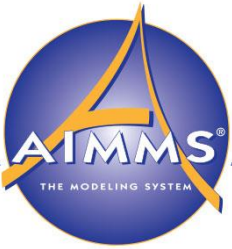


- Refer to a procedure that handles the error



Global Error Handler

- The procedure needs to have an element parameter with range `errh::AllPendingErrors` as input argument.
- The procedure contains statements that controls how the error is handled and how the error message is displayed to the end-user.



Global Error Handler

```
ErrorString := errh::Message(err);  
  
ErrorString += "/n contact your software development team.";  
  
...  
  
errh::MarkAsHandled(err);  
  
DialogMessage( ErrorString, "Error" );
```



Global Error Handler

- Example



Local Error Handler

- Priority over the Global Error Handler
- Created within a `BLOCK` statement
- Include `ONERROR err DO`
- Use it to handle hard to check errors



Local Error Handler

BLOCK

```
read from file ImportFile;
```

```
ONERROR err DO
```

```
if errh::Filename( err ) then
```

```
    ErrorString := FormatString( "Cannot read data  
                                from the file %s", ImportFile );
```

```
    DialogMessage( ErrorString, "Error" );
```

```
    errh::MarkAsHandled( err );
```

```
endif;
```

```
ENDBLOCK;
```



Local Error Handler

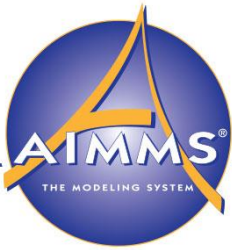
- Place it in the model where you suspect a runtime error might occur.
- Guide the end-user to the cause of the error to fix it.
- Will be forwarded to the Global Error Handler if not marked as handled.





Local Error Handler

- Example



Review

- Errors/Warnings dialog window
- To prevent preventable errors
- The benefits of an Error Handler
- Activation and setup of a Global Error Handler
- Address hard to check errors with a Local Error Handler





More on error handling

- AIMMS Language Reference
 - Chapter 8.4 – *Raising and handling warnings and errors*
- AIMMS Function Reference
 - Chapter 24 – *Error Handling Functions*
 - Chapter 36 – *Error Handling Related Identifiers*
- AIMMS Functional Examples
 - Error Handling
- Blogposts
 - Catching errors and warnings in AIMMS – Guido Diepen
 - Under control: managing errors and warnings within AIMMS – Peter Nieuwesteeg



Announcement of next webinar

The next webinar in this series, titled “Letting Your Application do the Modeling; Model Edit Functions”, will be presented by Chris Kuip, Senior AIMMS Developer.

Join us – January 28, 2015:

- 10 AM CET and
- 8AM PDT/11AM EDT



Q & A



You can always email us at support@aimms.com with questions or suggestions.