

# Adding variables into an output report filename

You can use the following variables and include them in an output report filename.

- Date variable
- Template variable
- Random number
- Project attribute

## Date variable

A date variable is a variable for the current date and time. (See section **\$date** in the previous chapter). The date variable consists of:

- A date variable with the default format.
- A date variable with a customized date format.

### Date variables with default format

Using a date variable without an attribute causes it to format the current date and time into a default format.

The syntax: **\$(date)**

For example:

```
C:\OutputReport\index_$(date).html
```

The output report filename would be *"index\_2011-12-21 11.55.44.html"*.

### Date variable with custom format

Using a date variable with an attribute causes it to format the current date and time into a specific format.

The syntax: **\$(date.get('format'))** or **\$(date.get("format"))**

For example:

```
C:\OutputReport\index_$(date.get('yyyyMMdd-hhmmss')).html
```

The output filename would be *"index\_20111221-115544.html"*.

## Template variable

A template variable is a variable for a template that is created in Report Wizard. It allows you to add all template variables into an output report filename (see [Including Variables in a Template](#)).

### Including template variables in an output report filename

When you include template variables created in an output report filename, the output filename which consists of the variables will be converted into value. This section uses the following template variables as examples, see figure below.

Parent	parent value
Child	child value

### Adding a value of a top-level variable

You can retrieve a top-level variable in an output report filename by using the following syntax.

The syntax: **\$(Parent)**

For example:

```
C:\OutputReport\index_$(Parent).html
```

The report output filename would be *"index\_parent value.html"*.

## Adding a value of a child variable

You can retrieve any of the following child variable values in an output report filename.

1. Getting a child variable at an index position:  
**The syntax: `${Parent.get(0)}`**
2. Getting a child variable by child names:  
**The syntax: `${Parent.get('Child')}` or `${Parent.get("Child")}`**
3. Getting a child value by referencing its name (in this case "Child"):  
**The syntax: `${Parent.Child}`**

For example:

```
C:\OutputReport\index_${Parent.get('Child')}.html
```

The output report filename would be "*index\_child value.html*".

## Random number

Report Wizard supports only pseudorandom numbers in an output report filename. The random number consists of:

- A random number with the default distributed int value.
- A random number with a distributed int value between 0 (inclusive) and specified value (exclusive).

### Random number with the default distributed integer value

You can use Report Wizard to generate pseudorandom numbers with the default distributed integer value.

**The syntax: `${random}`**

For example:

```
C:\OutputReport\index_${random}.html
```

The output report filename would be "*index\_21547.html*".

### Random number with a distributed integer value between 0 (inclusive) and specified value (exclusive)

You can use Report Wizard to generate pseudorandom numbers with the default distributed integer value between 0 and a specific value.

**The syntax: `${random.nextInt(n)}`**

The Parameters:

"n" - the bound on the random number to be returned. The "n" value must be a positive integer.

For example:

```
C:\OutputReport\index_${random.nextInt(1000)}.html
```

The output report filename would be "*index\_457.html*".

## Project attribute

You can add a project attribute or information about a project into an output report filename.

The project attribute consists of:

- Project name
- Project version

### Project name

You can retrieve the name of a project by using the following syntax.

**The syntax: `${project.name}`**

For example:

```
C:\OutputReport\index_${project.name}.html
```

The output report filename would be *"index\_myProject.html"*.

## Project version

You can retrieve a project version by using the following syntax.

**The syntax: \${project.version}**

For example:

```
C:\OutputReport\index_${project.version}.html
```

The output report filename would be *"index\_1234.html"*.

## Exception flow

You may accidentally specify an invalid output report filename such as:

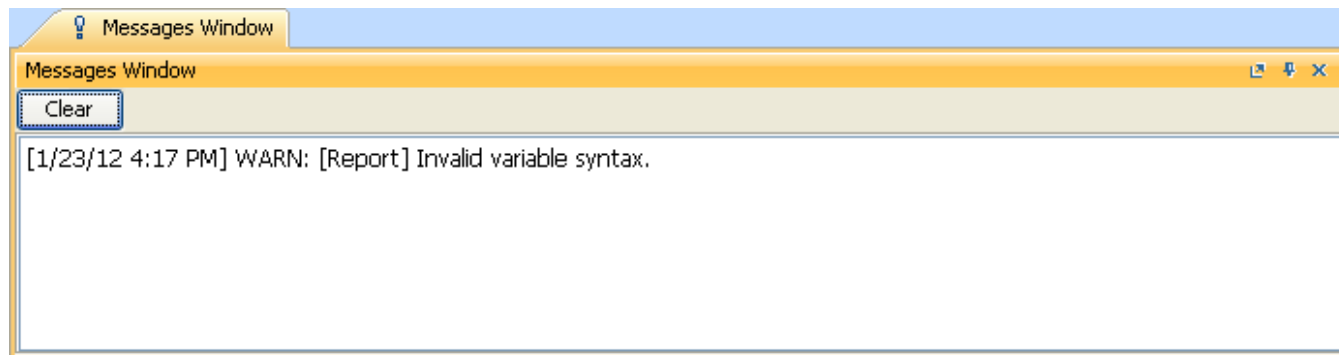
- An invalid syntax variable.
- A variable that does not exist in a template variable.
- An invalid parameter.

If this happens, Report Wizard will use this variable as a filename and a warning message will open in a log file (*\\modeling tool\\version\\md.log*).

For example:

```
C:\OutputReport\index_${date.get('yyyyMMdd-hhmmss')}.html
```

A single quote after "yyyyMMdd-hhmmss" was missing. Therefore, the output filename would be "index\_\${date.get('yyyyMMdd-hhmmss')}.html" and a warning message would open, see figure below.

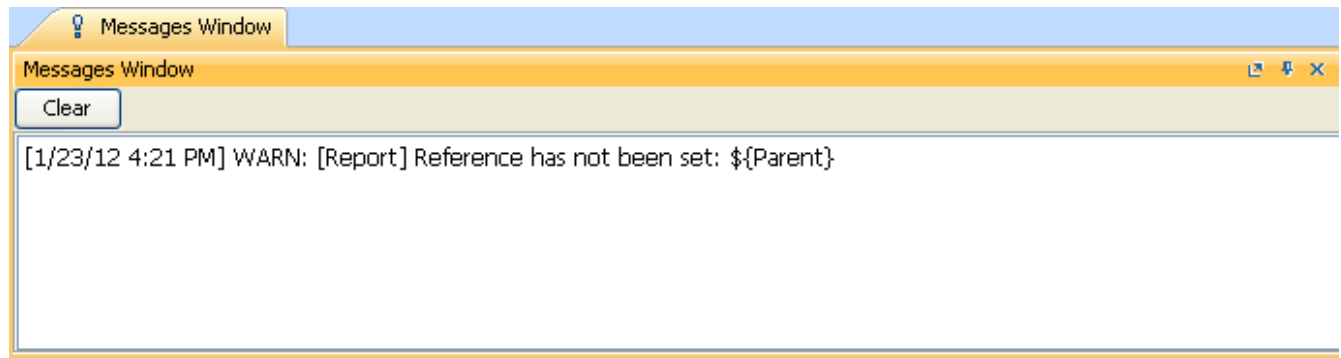


A Warning Message of an Invalid Variable Syntax.

For example:

```
C:\OutputReport\index_${Parent}.html
```

"\${Parent}" was not in the report template variable. Therefore, the output filename would be "index\_\${Parent}.html" and a warning message would open, see figure below.

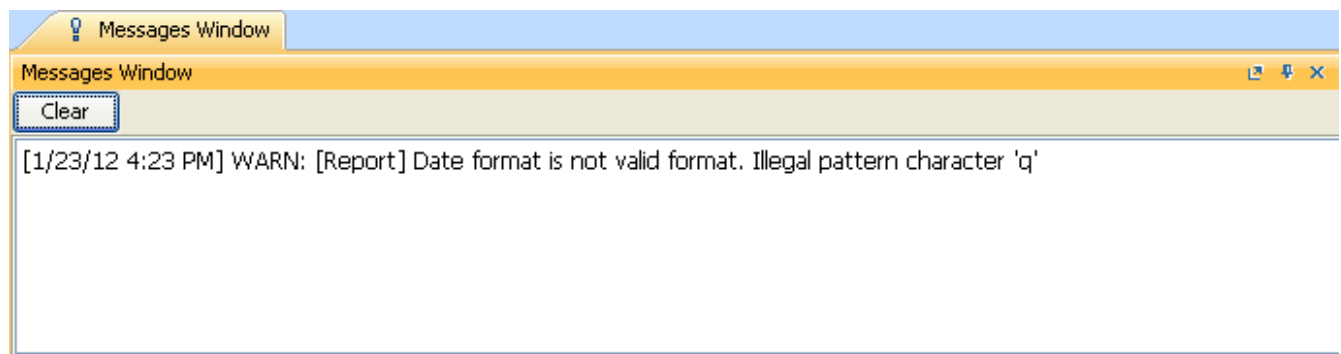


A Warning Message of a Variable that Does not Exist in a Template Variable.

For example:

```
C:\OutputReport\index_${date.get('yyyy-MM-dd qq')}.html
```

In this example, 'yyyy-MM-dd qq' is an invalid parameter. Therefore, output filename would be "index\_\${date.get('yyyy-MM-dd qq')}.html" and a warning message would open, see figure below.

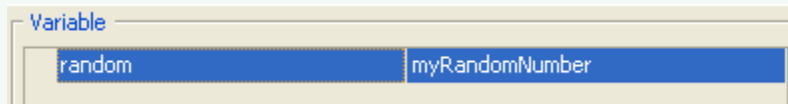


A Warning Message of an Invalid Parameter.



- All variables should be able to be combined together. For example:  
C:\OutputReport\index\_\${project.name}\_\${date}.html  
The output filename would be "index\_myProject\_2011-12-21 11.55.44.html".
- The template variable is the highest priority of all other variables. For example:  
C:\OutputReport\index\_\${random}.html

If you specify a template variable whose name is at random in the Report Data, the variable value will be used as the filename.



The output filename would be "index\_myRandomNumber.html".

If you do not specify a template variable whose name is at random in the Report Data, the random number with the default distributed integer value will be used as the filename.

The output filename would be "index\_12345.html".