

CanConvertToEnumExpression

Tue, Jan 14, 2020 [Linq](#)

CanConvertToEnumExpression

Table of Contents

- [CanConvertToEnumExpression](#)
 - [What it is](#)
 - [Code example](#)
 - [Explanation of the example](#)

What it is

CanConvertToEnumExpression returns true if a given property (or field or variable) can be converted to an Enum.

Before converting properties from one type to another, it may be necessary to check if the conversion is possible, in order to prevent runtime errors. When there is no doubt, the conversion can be done without checking.

- See [LINQ](#) for details about LINQ and the AI-Framework.
- Go to the category for [all LINQ expressions](#).

Compare with [ConvertToEnumExpression](#)

Code example

```
public StringProperty ExampleString => StringProperty(description: "string example");public EnumProperty ExampleEnum => EnumProperty(description: "enum example");public IMethodCall AssignEnum()  
{ return Method(() => new Body { new If(ExampleString.CanConvertToEnum()) { ExampleEnum.Assign(ExampleString.ConvertToEnum(
```

```
)) } });}
```

Explanation of the example

A StringProperty that contains the value "A", can **not** be converted to a Enum.

A StringProperty that contains the value "123.45", can be converted to a Enum. It results in the Enum 123.45.

In this example, the LINQ expression in line 8 CanConvertToEnum checks if the ExampleString can be converted to the type Enum.

The actual conversion (ConvertToEnum) is done in line 10.

Online URL:

<https://wiki-ai-framework.abstract-it.nl/article/canconverttoenumexpression-233.html>

```
SyntaxHighlighter.config.stripBrs=true; SyntaxHighlighter.all();
```