

ConvertToDecimalExpression

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

ConvertToDecimalExpression

Table of Contents

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

What it is

ConvertToDecimalExpression returns an Decimal, as a result of a conversion from another type, for example from a string to an Decimal.

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 [CanConvertToDecimalExpression](#)

Code example

```
public StringProperty ExampleString => StringProperty(description: "string example");public DecimalProperty ExampleDecimal => DecimalProperty(description: "decimal example");public IMethodCall AssignDecimal(){ return Method(() => new Body { new If(ExampleString.CanConvertToDecimal()) { ExampleDecimal.Assign(Example
```

```
String.ConvertToDecimal()) } }));}
```

Explanation of the example

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

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

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

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

Online URL:

<https://wiki-ai-framework.abstract-it.nl/article/converttodecimalexpression-238.html>

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