# Rational.Explicit Operator

名前空間: WS.Theia.ExtremelyPrecise

アセンブリ: ExtremelyPrecise.dll

# オーバーロード

|  |  |
| --- | --- |
| Explicit(Rational to Byte) | Rational オブジェクトから Byte 値への明示的な変換を定義します。 |
| Explicit(Rational to SByte) | Rational オブジェクトから SByte 値への明示的な変換を定義します。 |
| Explicit(Rational to Int32) | Rational オブジェクトから Int32 値への明示的な変換を定義します。 |
| Explicit(Rational to UInt32) | Rational オブジェクトから UInt32 値への明示的な変換を定義します。 |
| Explicit(Rational to Int16) | Rational オブジェクトから Int16 値への明示的な変換を定義します。 |
| Explicit(Rational to UInt16) | Rational オブジェクトから UInt16 値への明示的な変換を定義します。 |
| Explicit(Rational to Int64) | Rational オブジェクトから Int64 値への明示的な変換を定義します。 |
| Explicit(Rational to UInt64) | Rational オブジェクトから UInt64 値への明示的な変換を定義します。 |
| Explicit(Rational to Single) | Rational オブジェクトから Single 値への明示的な変換を定義します。 |
| Explicit(Rational to Double) | Rational オブジェクトから Double 値への明示的な変換を定義します。 |
| Explicit(Rational to Boolean) | Rational オブジェクトから Boolean 値への明示的な変換を定義します。 |
| Explicit(Rational to Decimal) | Rational オブジェクトから Decimal 値への明示的な変換を定義します。 |

# Explicit(Rational to Byte)

Rational 値からByte 値への明示的な変換を定義します。

public static explicit operator byte(WS.Theia.ExtremelyPrecise.Rational value);

## パラメータ

value　 Rational   
Byte 値へと変換する値。

## 戻り値

Byte  
valueパラメータと等価なByte値。

## 例外

OverFlowException

ValueがMinValueより小さいかMaxValueより大きいです。

## 例

次の例では、Rational型をByte型に変換する方法示します。

// Rational to Byte conversion.  
Rational goodByte = Rational.One;

Rational badByte = 256;  
  
byte byteFromRational;   
  
// Successful conversion using cast operator.  
byteFromRational = (byte) goodByte;  
Console.WriteLine(byteFromRational);  
  
// Handle conversion that should result in overflow.  
try  
{  
 byteFromRational = (byte) badByte;  
 Console.WriteLine(byteFromRational);  
}  
catch (OverflowException e)  
{  
 Console.WriteLine("Unable to convert {0}:\n {1}",   
 badByte, e.Message);  
}  
Console.WriteLine();

## 注釈

Byte型に変換する前にvalueパラメータの小数部を切り捨てが行われます。また、Rational型からByte型への暗黙的な変換は行われません。Rational型の表現範囲が大きく、変換を行うとオーバーフローが発生する恐れがあるためです。

# Explicit(Rational to SByte)

⚠重要  
この API は CLS 準拠ではありません。  
CLS 準拠の代替　System.Int16

Rational 値からSByte 値への明示的な変換を定義します。

public static explicit operator sbyte(WS.Theia.ExtremelyPrecise.Rational value);

## パラメータ

value　 Rational   
SByte値 へと変換する値。

## 戻り値

SByte  
valueパラメータと等価なSByte値。

## 例外

OverFlowException

ValueがMinValueより小さいかMaxValueより大きいです。

## 例

次の例では、Rational型をSByte型に変換する方法示します。

// Rational to Byte conversion.  
Rational goodByte = Rational.One;

Rational badByte = 256;  
  
byte byteFromRational;   
  
// Successful conversion using cast operator.  
byteFromRational = (byte) goodByte;  
Console.WriteLine(byteFromRational);  
  
// Handle conversion that should result in overflow.  
try  
{  
 byteFromRational = (byte) badByte;  
 Console.WriteLine(byteFromRational);  
}  
catch (OverflowException e)  
{  
 Console.WriteLine("Unable to convert {0}:\n {1}",   
 badByte, e.Message);  
}  
Console.WriteLine();

## 注釈

SByte型に変換する前にvalueパラメータの小数部を切り捨てが行われます。また、Rational型からSByte型への暗黙的な変換は行われません。Rational型の表現範囲が大きく、変換を行うとオーバーフローが発生する恐れがあるためです。

# Explicit(Rational to Int32)

Rational 値からInt32 値への明示的な変換を定義します。

public static explicit operator int(WS.Theia.ExtremelyPrecise.Rational value);

## パラメータ

value　 Rational   
Int32 値へと変換する値。

## 戻り値

Int32  
valueパラメータと等価なInt32値。

## 例外

OverFlowException

ValueがMinValueより小さいかMaxValueより大きいです。

## 例

次の例では、Rational型をInt32型に変換する方法示します。

// Rational to Int32 conversion.  
Rational goodInteger = 200000;  
Rational badInteger = 65000000000;  
  
int integerFromRatioanl;  
  
// Successful conversion using cast operator.   
integerFromRatioanl = (int) goodInteger;  
Console.WriteLine(integerFromRatioanl);  
  
// Handle conversion that should result in overflow.  
try  
{  
 integerFromRatioanl = (int) badInteger;  
 Console.WriteLine(integerFromRatioanl);  
}  
catch (OverflowException e)  
{  
 Console.WriteLine("Unable to convert {0}:\n {1}",   
 badInteger, e.Message);  
}  
Console.WriteLine();

## 注釈

Int32型に変換する前にvalueパラメータの小数部を切り捨てが行われます。また、Rational型からInt32型への暗黙的な変換は行われません。Rational型の表現範囲が大きく、変換を行うとオーバーフローが発生する恐れがあるためです。

# Explicit(Rational to UInt32)

⚠重要  
この API は CLS 準拠ではありません。  
CLS 準拠の代替　System.Int64

Rational 値からUInt32 値への明示的な変換を定義します。

public static explicit operator uint(WS.Theia.ExtremelyPrecise.Rational value);

## パラメータ

value　 Rational   
UInt32 値へと変換する値。

## 戻り値

UInt32  
valueパラメータと等価なUInt32値。

## 例外

OverFlowException

ValueがMinValueより小さいかMaxValueより大きいです。

## 例

次の例では、Rational型をUInt32型に変換する方法示します。

// Rational to UInt32 conversion.  
Rational goodUInteger = 200000;  
Rational badUInteger = 65000000000;  
  
uint uIntegerFromRational;  
  
// Successful conversion using cast operator.   
uIntegerFromRational = (uint) goodUInteger;  
Console.WriteLine(uIntegerFromRational);  
  
// Handle conversion that should result in overflow.  
try  
{  
 uIntegerFromRational = (uint) badUInteger;  
 Console.WriteLine(uIntegerFromRational);  
}  
catch (OverflowException e)  
{  
 Console.WriteLine("Unable to convert {0}:\n {1}",   
 badUInteger, e.Message);  
}  
Console.WriteLine();

## 注釈

UInt32型に変換する前にvalueパラメータの小数部を切り捨てが行われます。また、Rational型からUInt32型への暗黙的な変換は行われません。Rational型の表現範囲が大きく、変換を行うとオーバーフローが発生する恐れがあるためです。

# Explicit(Rational to Int16)

Rational 値からInt16 値への明示的な変換を定義します。

public static explicit operator short(WS.Theia.ExtremelyPrecise.Rational value);

## パラメータ

value　 Rational   
Int16 値へと変換する値。

## 戻り値

Int16  
valueパラメータと等価なInt16値。

## 例外

OverFlowException

ValueがMinValueより小さいかMaxValueより大きいです。

## 例

次の例では、Rational型をInt16型に変換する方法示します。

// Rational to Int16 conversion.  
Rational goodShort = 20000;  
Rational badShort = 33000;  
  
short shortFromRational;  
  
// Successful conversion using cast operator.   
shortFromRational = (short) goodShort;  
Console.WriteLine(shortFromRational);  
  
// Handle conversion that should result in overflow.  
try  
{  
 shortFromRational = (short) badShort;  
 Console.WriteLine(shortFromRational);  
}  
catch (OverflowException e)  
{  
 Console.WriteLine("Unable to convert {0}:\n {1}",   
 badShort, e.Message);  
}  
Console.WriteLine();

## 注釈

Int16型に変換する前にvalueパラメータの小数部を切り捨てが行われます。また、Rational型からInt16型への暗黙的な変換は行われません。Rational型の表現範囲が大きく、変換を行うとオーバーフローが発生する恐れがあるためです。

# Explicit(Rational to UInt16)

⚠重要  
この API は CLS 準拠ではありません。  
CLS 準拠の代替　System.Int32

Rational 値からUInt16 値への明示的な変換を定義します。

public static explicit operator ushort(WS.Theia.ExtremelyPrecise.Rational value);

## パラメータ

value　 Rational   
UInt16 値へと変換する値。

## 戻り値

UInt16  
valueパラメータと等価なUInt16値。

## 例外

OverFlowException

ValueがMinValueより小さいかMaxValueより大きいです。

## 例

次の例では、Rational型をUInt16型に変換する方法示します。

// Rational to UInt16 conversion.  
Rational goodUShort = 20000;  
Rational badUShort = 66000;  
  
ushort uShortFromRational;  
  
// Successful conversion using cast operator.   
uShortFromRational = (ushort) goodUShort;  
Console.WriteLine(uShortFromRational);  
  
// Handle conversion that should result in overflow.  
try  
{  
 uShortFromRational = (ushort) badUShort;  
 Console.WriteLine(uShortFromRational);  
}  
catch (OverflowException e)  
{  
 Console.WriteLine("Unable to convert {0}:\n {1}",   
 badUShort, e.Message);  
}  
Console.WriteLine();

## 注釈

UInt16型に変換する前にvalueパラメータの小数部を切り捨てが行われます。また、Rational型からUInt16型への暗黙的な変換は行われません。Rational型の表現範囲が大きく、変換を行うとオーバーフローが発生する恐れがあるためです。

# Explicit(Rational to Int64)

Rational 値からInt64 値への明示的な変換を定義します。

public static explicit operator long(WS.Theia.ExtremelyPrecise.Rational value);

## パラメータ

value　 Rational   
Int64 値へと変換する値。

## 戻り値

Int64  
valueパラメータと等価なInt64値。

## 例外

OverFlowException

ValueがMinValueより小さいかMaxValueより大きいです。

## 例

次の例では、Rational型をInt64型に変換する方法示します。

// Rational to Int64 conversion.  
Rational goodLong = 2000000000;  
Rational badLong = Math.Pow(goodLong, 3);  
  
long longFromRational;  
  
// Successful conversion using cast operator.   
longFromRational = (long) goodLong;  
Console.WriteLine(longFromRational);  
  
// Handle conversion that should result in overflow.  
try  
{  
 longFromRational = (long) badLong;  
 Console.WriteLine(longFromRational);  
}  
catch (OverflowException e)  
{  
 Console.WriteLine("Unable to convert {0}:\n {1}",   
 badLong, e.Message);  
}  
Console.WriteLine();

## 注釈

Int64型に変換する前にvalueパラメータの小数部を切り捨てが行われます。また、Rational型からInt64型への暗黙的な変換は行われません。Rational型の表現範囲が大きく、変換を行うとオーバーフローが発生する恐れがあるためです。

# Explicit(Rational to UInt64)

⚠重要  
この API は CLS 準拠ではありません。  
CLS 準拠の代替　System.Double

Rational 値からUInt64 値への明示的な変換を定義します。

public static explicit operator ulong(WS.Theia.ExtremelyPrecise.Rational value);

## パラメータ

value　 Rational   
UInt64 値へと変換する値。

## 戻り値

UInt64  
valueパラメータと等価なUInt64値。

## 例外

OverFlowException

ValueがMinValueより小さいかMaxValueより大きいです。

## 例

次の例では、Rational型をUInt64型に変換する方法示します。

// Rational to UInt64 conversion.  
Rational goodULong = 2000000000;  
Rational badULong = Math.Pow(goodULong, 3);  
  
ulong uLongFromRational;  
  
// Successful conversion using cast operator.   
uLongFromRational = (ulong) goodULong;  
Console.WriteLine(uLongFromRational);  
  
// Handle conversion that should result in overflow.  
try  
{  
 uLongFromRational = (ulong) badULong;  
 Console.WriteLine(uLongFromRational);  
}  
catch (OverflowException e)  
{  
 Console.WriteLine("Unable to convert {0}:\n {1}",   
 badULong, e.Message);  
}  
Console.WriteLine();

## 注釈

UInt64型に変換する前にvalueパラメータの小数部を切り捨てが行われます。また、Rational型からUInt64型への暗黙的な変換は行われません。Rational型の表現範囲が大きく、変換を行うとオーバーフローが発生する恐れがあるためです。

# Explicit(Rational to Single)

Rational 値から単精度浮動小数点への明示的な変換を定義します。

public static explicit operator float(WS.Theia.ExtremelyPrecise.Rational value);

## パラメータ

value　 Rational   
単精度浮動小数点へと変換する値。

## 戻り値

Single  
valueパラメータと等価な単精度浮動小数点。

## 例外

OverFlowException

ValueがMinValueより小さいかMaxValueより大きいです。

## 例

次の例では、Rational型を単精度浮動小数点型に変換する方法示します。

// Rational to Single conversion.  
Rational goodSingle = 102.43e22F;  
Rational badSingle = float.MaxValue;   
badSingle = badSingle \* 2;  
  
float singleFromRational;  
  
// Successful conversion using cast operator.   
singleFromRational = (float) goodSingle;  
Console.WriteLine(singleFromRational);  
  
// Convert an out-of-bounds Rational value to a Single.  
singleFromRational = (float) badSingle;  
Console.WriteLine(singleFromRational);

## 注釈

Rational型から単精度浮動小数点型への暗黙的な変換は行われません。Rational型の表現範囲が大きく、変換を行うとオーバーフローが発生する恐れがあるためです。

# Explicit(Rational to Double)

Rational 値から倍精度浮動小数点値への明示的な変換を定義します。

public static explicit operator double(WS.Theia.ExtremelyPrecise.Rational value);

## パラメータ

value　 Rational   
倍精度浮動小数点値へと変換する値。

## 戻り値

Double  
valueパラメータと等価な倍精度浮動小数点値。

## 例

次の例では、Rational型を倍精度浮動小数点型に変換する方法示します。

// Rational to Double conversion.  
Rational goodDouble = 102.43e22;  
Rational badDouble = Double.MaxValue;   
badDouble = badDouble \* 2;  
  
double doubleFromRational;  
  
// successful conversion using cast operator.  
doubleFromRational = (double) goodDouble;  
Console.WriteLine(doubleFromRational);  
  
// Convert an out-of-bounds Rational value to a Double.  
doubleFromRational = (double) badDouble;  
Console.WriteLine(doubleFromRational);

## 注釈

Rational型から倍精度浮動小数点型への暗黙的な変換は行われません。Rational型の表現範囲が大きく、変換を行うとオーバーフローが発生する恐れがあるためです。また、変換元の値がMaxValueより大きい場合はPositiveInfinity、MinValueより小さい場合はNegativeInfinityを返却します。

# Explicit(Rational to Boolean)

Rational 値からBoolean値への明示的な変換を定義します。

public static explicit operator bool(WS.Theia.ExtremelyPrecise.Rational value);

## パラメータ

value　 Rational   
Boolean値へと変換する値。

## 戻り値

Boolean  
valueパラメータと等価なBoolean値。（0の場合false、それ以外の場合true）

## 例

次の例では、Rational型をBoolean型に変換する方法示します。

// Rational to Boolean conversion.  
Rational trueBoolean = 102.43e22;  
Rational falseBoolean = 0;   
  
bool booleanFromRational;  
  
// successful conversion using cast operator in value of true.  
booleanFromRational = (bool) trueBoolean;  
Console.WriteLine(booleanFromRational);  
  
// successful conversion using cast operator in value of false.  
booleanFromRational = (bool) falseBoolean;  
Console.WriteLine(booleanFromRational);

# Explicit(Rational to Decimal)

Rational 値からDecimal値への明示的な変換を定義します。

public static explicit operator decimal(WS.Theia.ExtremelyPrecise.Rational value);

## パラメータ

value　 Rational   
Decimal値へと変換する値。

## 戻り値

Decimal  
valueパラメータと等価なDecimal値。

## 例外

OverFlowException

ValueがMinValueより小さいかMaxValueより大きいです。

## 例

次の例では、Rational型をDecimal型に変換する方法示します。

// Rational to Decimal conversion.  
Rational goodDecimal = 761652543;  
Rational badDecimal = Decimal.MaxValue;   
badDecimal += Rational.One;  
  
Decimal decimalFromRational;  
  
// Successful conversion using cast operator.  
decimalFromRational = (decimal) goodDecimal;  
Console.WriteLine(decimalFromRational);  
  
// Handle conversion that should result in overflow.  
try  
{  
 decimalFromRational = (decimal) badDecimal;  
 Console.WriteLine(decimalFromRational);  
}  
catch (OverflowException e)  
{  
 Console.WriteLine("Unable to convert {0}:\n {1}",   
 badDecimal, e.Message);  
}  
Console.WriteLine();

## 注釈

Rational型からDecimal型への暗黙的な変換は行われません。Rational型の表現範囲が大きく、変換を行うとオーバーフローが発生する恐れがあるためです。

# 適用対象

### .NET Core

2.0

### .NET Framework

4.6.1

### .NET Standard

2.0

### UWP

10.0.16299

### Xamarin.Android

8.0

### Xamarin.iOS

10.14

### Xamarin.Mac

3.8