

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 型に変換する方法を示します。

```
// 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 = 650000000000;

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 型に変換する方法を示します。

```
// Rational to UInt32 conversion.
Rational goodUInteger = 200000;
Rational badUInteger = 650000000000;

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 型に変換する方法を示します。

```
// 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 型に変換する方法を示します。

```
// 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 型に変換する方法を示します。

```
// Rational to Int64 conversion.
Rational goodLong = 20000000000;
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 型に変換する方法を示します。

```
// 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を示します。

```
// 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