

Общие выводы по декомпилированным методам

1. **DateTime.Parse()** и **DateTime.TryParse()**.

Оба метода реализованы через обращение к `internal static unsafe bool`

`DateTimeParse.TryParse`.

DateTime.Parse(). Если парсинг осуществлен, возвращается конвертированное значение, иначе - вызывается исключение.

DateTime.TryParse(). Если парсинг осуществлен, возвращается `true`, а по ссылке - конвертированное значение, иначе - возвращается `false`.

Т.е. методы отличаются в основном способом вывода результата конвертации и ошибок.

2. **int.Parse()** и **int.TryParse()**.

Оба метода вызывают

- `internal static unsafe int Number.ParseInt32()` и
- `internal static unsafe bool Number.TryParseInt32()` соответственно

В них (в обоих методах) происходит обращение к методам:

- к `private static bool HexNumberToInt32` и
- к `private static unsafe bool NumberToInt32`.

int.Parse().(Number.ParseInt32())

Если парсинг осуществлен, возвращается конвертированное значение, иначе - вызывается исключение.

int.TryParse(). (Number.TryParseInt32())

Если парсинг осуществлен, возвращается `true`, а по ссылке -конвертированное значение, иначе - возвращается `false`.

Аналогично, методы отличаются в основном способом вывода результата конвертации и ошибок.

3. **char.Parse(string s)** и **bool TryParse(string s, out char result)**

Эти методы устроены проще всего.

char.Parse(). Если длина строки 1, метод возвращает `s[0]`, иначе – исключение.

char.TryParse(). Если длина строки `null` или не равна 1, возвращает `false`, иначе – `true`, и по ссылке передает символ `s[0]`.

```

.method assembly hidebysig static valuetype System.DateTime
Parse(
    string s,
    class System.Globalization.DateTimeFormatInfo dtfi,
    valuetype System.Globalization.DateTimeStyles styles
) cil managed
{
    .maxstack 4
    .locals init (
        [0] valuetype System.DateTimeResult result
    )

    // [2182 7 - 2182 51]
    IL_0000: ldloc.s     result
    IL_0002: initobj     System.DateTimeResult

    // [2183 7 - 2183 20]
    IL_0008: ldloc.s     result
    IL_000a: call       instance void System.DateTimeResult::Init()

    // [2184 7 - 2184 63]
    IL_000f: ldarg.0     // s
    IL_0010: ldarg.1     // dtfi
    IL_0011: ldarg.2     // styles
    IL_0012: ldloc.s     result
    IL_0014: call       bool System.DateTimeParse::TryParse(string, class System.Globalization.DateTimeFormatInfo, valuetype
System.Globalization.DateTimeStyles, valuetype System.DateTimeResult&)
    IL_0019: brfalse.s   IL_0022

    // [2185 9 - 2185 34]
    IL_001b: ldloc.0     // result
    IL_001c: ldfld     valuetype System.DateTime System.DateTimeResult::parsedDate
    IL_0021: ret

    // [2186 7 - 2186 65]
    IL_0022: ldloc.s     result
    IL_0024: call       class System.Exception System.DateTimeParse::GetDateTimeParseException(valuetype System.DateTimeResult&)
    IL_0029: throw

} // end of method DateTimeParse::Parse

```

```

internal static DateTime Parse(
    string s,
    DateTimeFormatInfo dtfi,
    DateTimeStyles styles)
{
    DateTimeResult result = new DateTimeResult();
    result.Init();
    if (DateTimeParse.TryParse(s, dtfi, styles, ref result))
        return result.parsedDate;
    throw DateTimeParse.GetDateTimeParseException(ref result);
}

```

//Обращение к internal static unsafe bool TryParse.

// Если парсинг осуществлен, возвращается конвертированное значение, иначе -
вызывается исключение.

```

.method assembly hidebysig static bool
TryParse(
    string s,
    class System.Globalization.DateTimeFormatInfo dtfi,
    valuetype System.Globalization.DateTimeStyles styles,
    [out] valuetype System.DateTime& result
) cil managed
{
    .maxstack 4
    .locals init (
        [0] valuetype System.DateTimeResult result1
    )

    // [2210 7 - 2210 33]
    IL_0000: ldarg.3 // result
    IL_0001: ldsfld     valuetype System.DateTime System.DateTime::MinValue
    IL_0006: stobj     System.DateTime

    // [2211 7 - 2211 52]
    IL_000d: ldloc.s   result1
    IL_000d: initobj   System.DateTimeResult

    // [2212 7 - 2212 21]
    IL_0013: ldloc.s   result1
    IL_0015: call     instance void System.DateTimeResult::Init()

    // [2213 7 - 2213 65]
    IL_001a: ldarg.0 // s
    IL_001b: ldarg.1 // dtfi
    IL_001c: ldarg.2 // styles
    IL_001d: ldloc.s   result1
    IL_001f: call     bool System.DateTimeParse::TryParse(string, class System.Globalization.DateTimeFormatInfo, valuetype System.Globalization.DateTimeStyles, valuetype System.DateTimeResult&)
    IL_0024: brfalse.s  IL_0034

    // [2215 7 - 2215 34]
    IL_0026: ldarg.3 // result
    IL_0027: ldloc.0 // result1
    IL_0028: ldfld     valuetype System.DateTime System.DateTimeResult::parsedDate
    IL_002d: stobj     System.DateTime

    // [2216 7 - 2216 19]
    IL_0032: ldc.i4.1
    IL_0033: ret

    // [2214 9 - 2214 22]
    IL_0034: ldc.i4.0
    IL_0035: ret

} // end of method DateTimeParse::TryParse

```

```

internal static bool TryParse(
    string s,
    DateTimeFormatInfo dtfi,
    DateTimeStyles styles,
    out DateTime result)
{
    result = DateTime.MinValue;
    DateTimeResult result1 = new DateTimeResult();
    result1.Init();
    if (!DateTimeParse.TryParse(s, dtfi, styles, ref result1))
        return false;
    result = result1.parsedDate;
    return true;
}

```

//Обращение к internal static unsafe bool TryParse.

// Если парсинг осуществлен, возвращается true, а по ссылке -конвертированное значение, иначе -возвращается false.

```
[__DynamicallyInvokable]
public static int Parse(string s) => Number.ParseInt32(s, NumberStyles.Integer,
NumberFormatInfo.CurrentInfo);
```

```
.method public hidebysig static int32
    string s
    ) cil managed
{
    .custom instance void __DynamicallyInvokableAttribute::.ctor()
        = (01 00 00 00 )
    .maxstack 8

    // [71 42 - 71 114]
    IL_0000: ldarg.0          // s
    IL_0001: ldc.i4.7
    IL_0002: call          class System.Globalization.NumberFormatInfo
System.Globalization.NumberFormatInfo::get_CurrentInfo()
    IL_0007: call          int32 System.Number::ParseInt32(string, valuetype
System.Globalization.NumberStyles, class System.Globalization.NumberFormatInfo)
    IL_000c: ret
```

[SecuritySafeCritical]

```
internal static unsafe int ParseInt32(string s, NumberStyles style, NumberFormatInfo info)
{
    byte* stackBuffer = stackalloc byte[Number.NumberBuffer.NumberBufferBytes];
    Number.NumberBuffer number = new Number.NumberBuffer(stackBuffer);
    int int32 = 0;
    Number.StringToNumber(s, style, ref number, info, false);
    if ((style & NumberStyles.AllowHexSpecifier) != NumberStyles.None)
    {
        if (!Number.HexNumberToInt32(ref number, ref int32))
            throw new OverflowException(Environment.GetResourceString("Overflow_Int32"));
    }
    else if (!Number.NumberToInt32(ref number, ref int32))
        throw new OverflowException(Environment.GetResourceString("Overflow_Int32"));
    return int32;
}
```

//Происходит обращение к функциям:

// private static bool HexNumberToInt32

//private static unsafe bool NumberToInt32

// Если парсинг осуществлен, возвращается конвертированное значение, иначе -
вызывается исключение.

```

.method assembly hidebysig static int32
ParseInt32(
    string s,
    valuetype System.Globalization.NumberStyles style,
    class System.Globalization.NumberFormatInfo info
) cil managed
{
    .custom instance void System.Security.SecuritySafeCriticalAttribute::.ctor()
    = (01 00 00 00 )
    .maxstack 5
    .locals init (
        [0] unsigned int8* stackBuffer,
        [1] valuetype System.Number/NumberBuffer number,
        [2] int32 'int32'
    )

    // [333 7 - 333 81]
    IL_0000: ldsfld          int32 System.Number/NumberBuffer::NumberBufferBytes
    IL_0005: conv.u         IL_0006: ldc.i4.1
    IL_0007: mul.ovf.un
    IL_0008: localloc
    IL_000a: stloc.0         // stackBuffer

    // [334 7 - 334 72]
    IL_000b: ldloc.s         number
    IL_000d: ldloc.0         // stackBuffer
    IL_000e: call          instance void System.Number/NumberBuffer::.ctor(unsigned int8*)

    // [335 7 - 335 20]
    IL_0013: ldc.i4.0
    IL_0014: stloc.2         // 'int32'

    // [336 7 - 336 63]
    IL_0015: ldarg.0         // s
    IL_0016: ldarg.1         // style
    IL_0017: ldloc.s         number
    IL_0019: ldarg.2         // info
    IL_001a: ldc.i4.0
    IL_001b: call          void System.Number::StringToNumber(string, valuetype System.Globalization.NumberStyles, valuetype System.Number/NumberBuffer&, class System.Globalization.NumberFormatInfo, bool)

    // [337 7 - 337 73]
    IL_0020: ldarg.1         // style
    IL_0021: ldc.i4         512 // 0x00000200
    IL_0026: and
    IL_0027: brfalse.s     IL_0044

    // [339 9 - 339 61]
    IL_0029: ldloc.s         number
    IL_002b: ldloc.s         'int32'
    IL_002d: call          bool System.Number::HexNumberToInt32(valuetype System.Number/NumberBuffer&, int32&)
    IL_0032: brtrue.s      IL_005f

    // [340 11 - 340 88]
    IL_0034: ldstr          "Overflow_Int32"
    IL_0039: call          string System.Environment::GetResourceString(string)
    IL_003e: newobj         instance void System.OverflowException::.ctor(string)
    IL_0043: throw

    // [342 12 - 342 61]
    IL_0044: ldloc.s         number
    IL_0046: ldloc.s         'int32'
    IL_0048: call          bool System.Number::NumberToInt32(valuetype System.Number/NumberBuffer&, int32&)
    IL_004d: brtrue.s     IL_005f

    // [343 9 - 343 86]
    IL_004f: ldstr          "Overflow_Int32"
    IL_0054: call          string System.Environment::GetResourceString(string)
    IL_0059: newobj         instance void System.OverflowException::.ctor(string)
    IL_005e: throw

    // [344 7 - 344 20]
    IL_005f: ldloc.2         // 'int32'
    IL_0060: ret

} // end of method Number::ParseInt32

```

[__DynamicallyInvokable]

```
public static bool TryParse(string s, out int result) => Number.TryParseInt32(s,
NumberStyles.Integer, NumberFormatInfo.CurrentInfo, out result);
```

```
.method public hidebysig static bool
TryParse(
    string s,
    [out] int32& result
) cil managed
{
    .custom instance void __DynamicallyInvokableAttribute::.ctor()
    = (01 00 00 00 )
    .maxstack 8

    // [91 62 - 91 149]
    IL_0000: ldarg.0          // s
    IL_0001: ldc.i4.7
    IL_0002: call          class System.Globalization.NumberFormatInfo System.Globalization.NumberFormatInfo::get_CurrentInfo()
    IL_0007: ldarg.1          // result
    IL_0008: call          bool System.Number::TryParseInt32(string, valuetype System.Globalization.NumberStyles, class
System.Globalization.NumberFormatInfo, int32&)
    IL_000d: ret

} // end of method Int32::TryParse
```

[SecuritySafeCritical]

```
internal static unsafe bool TryParseInt32(
    string s,
    NumberStyles style,
    NumberFormatInfo info,
    out int result)
{
    byte* stackBuffer = stackalloc byte[Number.NumberBuffer.NumberBufferBytes];
    Number.NumberBuffer number = new Number.NumberBuffer(stackBuffer);
    result = 0;
    if (!Number.TryStringToNumber(s, style, ref number, info, false))
        return false;
    if ((style & NumberStyles.AllowHexSpecifier) != NumberStyles.None)
    {
        if (!Number.HexNumberToInt32(ref number, ref result))
            return false;
    }
    else if (!Number.NumberToInt32(ref number, ref result))
        return false;
    return true;
}
```

//Происходит обращение к функциям:

// private static bool HexNumberToInt32

//private static unsafe bool NumberToInt32

// Если парсинг осуществлен, возвращается true, а по ссылке -конвертированное значение,
иначе -возвращается false.

```

.method assembly hidebysig static bool
TryParseInt32(
    string s,
    valueType System.Globalization.NumberStyles style,
    class System.Globalization.NumberFormatInfo info,
    [out] int32& result
) cil managed
{
    .custom instance void System.Security.SecuritySafeCriticalAttribute::ctor()
    = (01 00 00 00)
    .maxstack 5
    .locals init (
        [0] unsigned int8* stackBuffer,
        [1] valueType System.Number/NumberBuffer number
    )

    // [728 7 - 728 81]
    IL_0000: ldsfld      int32 System.Number/NumberBuffer::NumberBufferBytes
    IL_0005: conv.u
    IL_0006: ldc.i4.1
    IL_0007: mul.ovf.un
    IL_0008: localloc
    IL_000a: stloc.0 // stackBuffer

    // [729 7 - 729 72]
    IL_000b: ldloc.s      number
    IL_000d: ldloc.0 // stackBuffer
    IL_000e: call      instance void System.Number/NumberBuffer::ctor(unsigned int8*)

    // [730 7 - 730 17]
    IL_0013: ldarg.3 // result
    IL_0014: ldc.i4.0
    IL_0015: stind.i4

    // [731 7 - 731 72]
    IL_0016: ldarg.0 // s
    IL_0017: ldarg.1 // style
    IL_0018: ldloc.s      number
    IL_001a: ldarg.2 // info
    IL_001b: ldc.i4.0
    IL_001c: call      bool System.Number::TryStringToNumber(string, valueType System.Globalization.NumberStyles, valueType System.Number/NumberBuffer&, class System.Globalization.NumberFormatInfo, bool)
    IL_0021: brtrue.s      IL_0025

    // [732 9 - 732 22]
    IL_0023: ldc.i4.0
    IL_0024: ret

    // [733 7 - 733 73]
    IL_0025: ldarg.1 // style
    IL_0026: ldc.i4      512 // 0x00000200
    IL_002b: and
    IL_002c: brfalse.s      IL_003a

    // [735 9 - 735 62]
    IL_002e: ldloc.s      number
    IL_0030: ldarg.3 // result
    IL_0031: call      bool System.Number::HexNumberToInt32(valueType System.Number/NumberBuffer&, int32&)
    IL_0036: brtrue.s      IL_0046

    // [736 11 - 736 24]
    IL_0038: ldc.i4.0
    IL_0039: ret

    // [738 12 - 738 62]
    IL_003a: ldloc.s      number
    IL_003c: ldarg.3 // result
    IL_003d: call      bool System.Number::NumberToInt32(valueType System.Number/NumberBuffer&, int32&)
    IL_0042: brtrue.s      IL_0046

    // [739 9 - 739 22]
    IL_0044: ldc.i4.0
    IL_0045: ret

    // [740 7 - 740 19]
    IL_0046: ldc.i4.1
    IL_0047: ret

} // end of method Number::TryParseInt32

```

```
[__DynamicallyInvokable]
public static char Parse(string s)
{
    if (s == null)
        throw new ArgumentNullException(nameof(s));
    return s.Length == 1 ? s[0] : throw new
FormatException(Environment.GetResourceString("Format_NeedSingleChar"));
}
//Если длина строки 1, метод возвращает s[0], иначе – исключение.
```

```
.method public hidebysig static char
    Parse(
        string s
    ) cil managed
{
    .custom instance void __DynamicallyInvokableAttribute::.ctor()
        = (01 00 00 00 )
    .maxstack 8

    // [327 7 - 327 21]
    IL_0000: ldarg.0          // s
    IL_0001: brtrue.s       IL_000e

    // [328 9 - 328 53]
    IL_0003: ldstr          "s"
    IL_0008: newobj         instance void System.ArgumentNullException::.ctor(string)
    IL_000d: throw

    // [329 7 - 329 119]
    IL_000e: ldarg.0          // s
    IL_000f: callvirt         instance int32 System.String::get_Length()
    IL_0014: ldc.i4.1
    IL_0015: beq.s         IL_0027

    IL_0017: ldstr          "Format_NeedSingleChar"
    IL_001c: call         string System.Environment::GetResourceString(string)
    IL_0021: newobj         instance void System.FormatException::.ctor(string)
    IL_0026: throw
    IL_0027: ldarg.0          // s
    IL_0028: ldc.i4.0
    IL_0029: callvirt         instance char System.String::get_Chars(int32)
    IL_002e: ret

} // end of method Char::Parse
```

```

[__DynamicallyInvokable]
public static bool TryParse(string s, out char result)
{
    result = char.MinValue;
    if (s == null || s.Length != 1)
        return false;
    result = s[0];
    return true;
}

```

//Если длина строки **null** или не равна 1, возвращает false, иначе – **true**, и по ссылке передает символ s[0].

```

.method public hidebysig static bool
    TryParse(
        string s,
        [out] char& result
    ) cil managed
{
    .custom instance void __DynamicallyInvokableAttribute::.ctor()
    = (01 00 00 00 )
    .maxstack 8

    // [335 7 - 335 29]
    IL_0000: ldarg.1          // result
    IL_0001: ldc.i4.0
    IL_0002: stind.i2

    // [336 7 - 336 38]
    IL_0003: ldarg.0          // s
    IL_0004: brtrue.s         IL_0008

    // [337 9 - 337 22]
    IL_0006: ldc.i4.0
    IL_0007: ret

    IL_0008: ldarg.0          // s
    IL_0009: callvirt         instance int32 System.String::get_Length()
    IL_000e: ldc.i4.1
    IL_000f: beq.s         IL_0013
    IL_0011: ldc.i4.0
    IL_0012: ret

    // [338 7 - 338 20]
    IL_0013: ldarg.1          // result
    IL_0014: ldarg.0          // s
    IL_0015: ldc.i4.0
    IL_0016: callvirt         instance char System.String::get_Chars(int32)
    IL_001b: stind.i2

    // [339 7 - 339 19]
    IL_001c: ldc.i4.1
    IL_001d: ret

} // end of method Char::TryParse

```