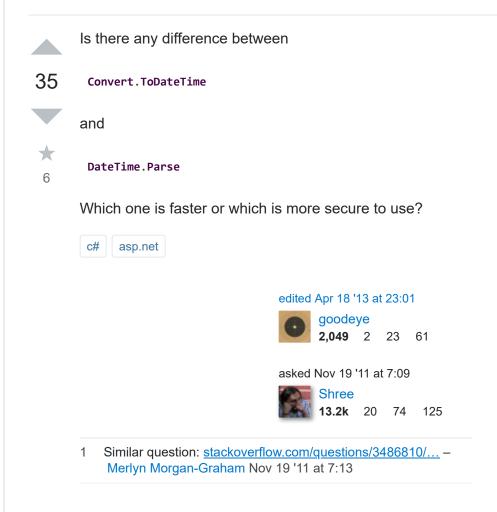
Any difference between DateTime.Parse and Convert.ToDateTime?

Ask Question



3 Answers

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.



Per an answer on another forum from Jon Skeet...

29



Convert.ToDateTime uses DateTime.Parse internally, with the current culture - unless you pass it null, in which case it returns DateTime.MinValue.



If you're not sure string is a valid DateTime, use neither and instead, use DateTime.TryParse()

If you're sure the string is a valid DateTime, and you know the format, you could also consider the DateTime.ParseExact() or DateTime.TryParseExact() methods.

edited Feb 12 '13 at 17:04

answered Nov 19 '11 at 7:16



David 62.2k

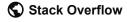
2k 13 116 158

"In short, Convert.ToDateTime() eliminates the code necessary to set the CultureInfo, which you'd have to do to use DateTime.Parse()" That's not true. You donn't have to set the CultureInfo to use DateTime.Parse(). DateTime.Parse has an overload that takes only 1 string and nothing else and it uses the current culture info without you having to pass it in. – Nick Feb 12 '13 at 16:32

@Nick - You're right. Thank you! I edited this. – David Feb 12 '13 at 17:05

Home

PUBLIC



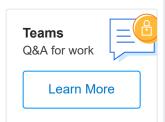
Tags



DateTime.Parse has an overload that takes only one String and nothing else and it uses the current Locale

info without you having to page it in

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.





answered Apr 10 '13 at 11:29





The overloads of Convert.ToDateTime which take string as input parameter, internally invoked DateTime.Parse. Following is the implementation of Convert.ToDateTime.



```
public static DateTime ToDateTime(string value)
{
    if (value == null)
      {
        return new DateTime(OL);
    }
    return DateTime.Parse(value, CultureInfo.CurrentCulture)
}
```

In case of other overload, the parameter is casted into IConvertible interface and then the corresponding ToDateTime method is invoked.

```
public static DateTime ToDateTime(ushort value)
{
    return ((IConvertible) value).ToDateTime(null);
}
```

answered Nov 19 '11 at 7:25



By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.