# Sending Outlook meeting requests without Outlook?

Asked 15 years, 11 months ago  Modified 9 years, 11 months ago

Viewed 65k times

40

I just wonder if it is possible to send Meeting Requests to people without having Outlook installed on the Server and using COM Interop (which I want to avoid on a server at all costs).

We have Exchange 2003 in a Windows 2003 Domain and all users are domain Users. I guess I can send 'round iCal/vCal or something, but I wonder if there is a proper standard way to send Meeting Requests through Exchange without Outlook?

This is C#/.net if it matters.

c# .net outlook

Share

Improve this question

Follow

asked Jan 20, 2009 at 15:57

Michael Stum

181k ● 119 ● 407 ● 540

1 The amount of time you chose to wait makes me think that my suggestion is not exactly what you had in mind. Did you at least get it to work? – Tomalak Jan 23, 2009 at 10:53

1 I just did not have time to test it earlier. A quick test of it seemed to work, and I sent an Outlook Meeting Invitation to an account that has POP3 enabled to get a "raw dump". These two bits are good enough for me to figure the rest out, since I only need to send meetings but not care about reply – Michael Stum Jan 23, 2009 at 13:42

1 In any case, your answer was tremendously helpful, many thanks! – Michael Stum Jan 23, 2009 at 13:43

1 Cheers! :) Good to hear. – Tomalak Jan 23, 2009 at 15:05

## 5 Answers

Sorted by: Highest score (default)

58

The way to send a meeting request to Outlook (and have it recognized) goes like this:

- prepare an iCalendar file, be sure to set these additional properties, as Outlook needs them:
  - `UID`
  - `SEQUENCE`
  - `CREATED`
  - `LAST-MODIFIED`
  - `DTSTAMP`
- prepare a `multipart/alternative` mail:

    - Part 1: `text/html` (or whatever you like) - this is displayed to "ordinary" mail readers or as a fall-back and contains a summary of the event in human readable form
    - Part 2: `text/calendar; method=REQUEST`, holds the contents of the ics file (the header `method` parameter must match the method in the ics). Watch out for the correct text encoding, declaring a `charset` header parameter won't hurt.
    - Part 3: Optionally, attach the .ics file itself, so ordinary mail readers can offer the user something to click on. Outlook does not really require the attachment because it just reads the `text/calendar` part.
- Send the mail to an outlook user. If you got everything right the mail shows up as a meeting request, complete with attendance buttons and automatic entry in the users calendar upon accept.
- Set up something that processes the responses (they go to the meeting organizer). I have not yet been able to get automatic attendee tracking to work with an Exchange mailbox because the event won't exist in the organizers calendar. Outlook needs the UIDs and SEQUENCES to match it's expectations, but with a UID you made up this will hardly work.

For help on the details and peculiarities of the ics file format, be sure to visit the [iCalendar Specification]

Excerpts by Masahide Kanzaki. They are a light in the dark, much better than gnawing your way through RFC 2445. But then again, maybe a handy library exists for .NET.

Share Improve this answer

Follow

edited Oct 7, 2021 at 5:46

Community Bot

1 • 1

answered Jan 20, 2009 at 16:45

Tomalak

338k • 68 • 545 • 635

I just did not have time to test it earlier. A quick test of it seemed to work, and I sent an Outlook Meeting Invitation to an account that has POP3 enabled to get a "raw dump". These two bits are good enough for me to figure the rest out, since I only need to send meetings but not care about reply – Michael Stum Jan 23, 2009 at 13:42

1 Removing the following header from Part 2 did the trick for me: Content-Disposition: attachment; filename=meeting.ics As long as this header was in the mail, Outlook would display the ICS as an attachment. – Bob Dec 2, 2009 at 20:14

This is what happens when a `Content-Disposition: attachment` header is present. ;-) – Tomalak Dec 3, 2009 at 7:42

1 Thanks, this was really useful - spent a long time wondering why it didn't work but realised I was sending mail as multipart/mixed not mulitpart/alternative. Duh! – Mat Mannion Mar 14, 2011 at 17:02

8

See the DDay.iCal C# library on sourceforge:
http://sourceforge.net/projects/dday-ical/

Then read this codeproject article:
http://www.codeproject.com/Articles/17980/Adding-iCalendar-Support-to-Your-Program-Part-1

And read this:
Export event with C# to iCalendar and vCalendar format

Share Improve this answer

Follow

edited May 23, 2017 at 12:32

Community Bot

1 ● 1

answered Oct 17, 2012 at 18:59

Stefan Steiger

81.9k ● 69 ● 399 ● 454

---

6

iCalendar is a great general-purpose solution, and the DDay.iCal library is a great way to do this from .NET, but I believe Exchange Web Services (EWS) are a better solution in the context of the original question (Exchange, C#/.NET).

And if you're using a .NET language such as C#, you should use the EWS Managed API wrapper which greatly simplifies working with EWS.

From the docs, here's how to use the EWS Managed API to create a meeting and send the request to invitees:

```
// Create the appointment.
Appointment appointment = new Appointment(service);

// Set properties on the appointment. Add two required
optional attendee.
appointment.Subject = "Status Meeting";
appointment.Body = "The purpose of this meeting is to
appointment.Start = new DateTime(2009, 3, 1, 9, 0, 0);
appointment.End = appointment.Start.AddHours(2);
appointment.Location = "Conf Room";
appointment.RequiredAttendees.Add("user1@contoso.com")
appointment.RequiredAttendees.Add("user2@contoso.com")
appointment.OptionalAttendees.Add("user3@contoso.com")

// Send the meeting request to all attendees and save
folder.
appointment.Save(SendInvitationsMode.SendToAllAndSaveC
```

Share Improve this answer

Follow

answered Aug 18, 2014 at 23:34

Oran Dennison

**3,297** ● 1 ● 30 ● 38

1 I agree, this solution is much simpler than the others. It should also be noted that EWS doesn't require the Outlook COM interop. – JordanTDN Jul 11, 2017 at 15:29

**5**

The code below will send a meeting request in such a way that Outlook will render Accept/Decline buttons.

Note that UID must be unique per meeting, I've used a GUID.

Also note you need to replace CREATED, DTSTART, DTEND, DTSTAMP, LAST-MODIFIED. These are UTC

date/times.

```
    var m = new MailMessage();

    m.Subject = "Meeting";

    m.Body = "";

    string iCal =
@"BEGIN:VCALENDAR
PRODID:-//Microsoft Corporation//Outlook 14.0 MIMEDIR/
VERSION:2.0
METHOD:PUBLISH
X-MS-OLK-FORCEINSPECTOROPEN:TRUE
BEGIN:VEVENT
CLASS:PUBLIC
CREATED:20140423T045933Z
DESCRIPTION:desc
DTEND:20140430T080000Z
DTSTAMP:20140423T045933Z
DTSTART:20140430T060000Z
LAST-MODIFIED:20140423T045933Z
LOCATION:location...
PRIORITY:5
SEQUENCE:0
SUMMARY;LANGUAGE=en-us:Summary...
TRANSP:OPAQUE
UID:D8BFD357-88A7-455C-86BC-C2CECA9AC5C6
X-MICROSOFT-CDO-BUSYSTATUS:BUSY
X-MICROSOFT-CDO-IMPORTANCE:1
X-MICROSOFT-DISALLOW-COUNTER:FALSE
X-MS-OLK-AUTOFILLLOCATION:FALSE
X-MS-OLK-CONFTYPE:0
BEGIN:VALARM
TRIGGER:-PT60M
ACTION:DISPLAY
DESCRIPTION:Reminder
END:VALARM
END:VEVENT
END:VCALENDAR";

    using (var iCalView = AlternateView.CreateAlternat
System.Net.Mime.ContentType("text/calendar")))
```

```
    {
        m.AlternateViews.Add(iCalView);

        var c = new SmtpClient();

        // Send message
        c.Send(m);
    }
```

This assumes you have a local SMTP server configured in your config file:

```
<system.net>
  <mailSettings>
    <smtp deliveryMethod="Network" from="no-reply@ex
      <network defaultCredentials="true" host="smtp.
    </smtp>
  </mailSettings>
</system.net>
```

Share Improve this answer Follow

edited Apr 24, 2014 at 2:43

answered Apr 24, 2014 at 1:57

saille

**9,151** ● 5 ● 47 ● 58

---

5

You can send meeting requests by mail to outlook using the [iCal Standard (RFC 5545)]

You can't send todo items this way. You may send "Appointments" but these appear in outlook as .ics attachments which have to be accepted "blindly".

Meeting requests appear in outlook with a nice preview and can be accepted or rejeted. The sending program may modify or cancel the meeting after it was sent.

It's easieset to create a valid iCal item with the DDay.iCal .Net Library

The code below is a complete working example. It builds a string with a valid iCal meeting request and sends it by mail.

The code creates a mail with:

- plain text body for simple mail clients
- HTML body for diplay in modern mail clients
- iCal meeting request as AlternateView (will display in Outlook)
- iCal meeting request as Attachment (usable in mail clients other than outlook)

The code shows how to add:

- description text as HTML, looks nicer in outlook
- Priority, visibility (public/private/confidential)
- optional organizer (will show in outlook instead of the mail sender)
- optional attendees
- optional alarm

- optional attachments to the meeting. will show up in outlook's calendar

Some important details:

- mail sender (or optional organizer) and mail receiver must be different to make this work in outlook
- METHOD in .ics and METHOD in Mime.ContentType must match
- The meeting must lie in the future to make this work in outlook
- the .ics part must be the last alternateView part in the MIME mail

The exact details about the way outlook interprets .ics files are detailed in [MS-OXCICAL]: iCalendar to Appointment Object Conversion Algorithm

We'll use these assemblies:

```
using System;
using System.IO;
using System.Net.Mail;
using DDay.iCal;
using DDay.iCal.Serialization.iCalendar;
```

For DDay.iCal its enough to download the DDay.iCal binary Files. If you want to add some features it's best to look at the DDay.iCal sources because the documentation is outdated and the sources contain pretty complete tests which excercise all its features.

```
const string filepath = @"C:\temp\ical.test.ics";
// use PUBLISH for appointments
// use REQUEST for meeting requests
const string METHOD = "REQUEST";

// Properties of the meeting request
// keep guid in sending program to modify or cancel th
Guid uid = Guid.Parse("2B127C67-73B3-43C5-A804-5666C2C
string VisBetreff = "This is the subject of the meetin
string TerminVerantwortlicherEmail = "mr.asker@myorg.c
string bodyPlainText = "This is the simple iCal plain
string bodyHtml = "This is the simple <b>iCal HTML mes
string location = "Meeting room 101";
// 1: High
// 5: Normal
// 9: low
int priority = 1;
//=====================================
MailMessage message = new MailMessage();

message.From = new MailAddress("sender@myorg.com");
message.To.Add(new MailAddress(TerminVerantwortlicherE
message.Subject = "[VIS-Termin] " + VisBetreff;

// Plain Text Version
message.Body = bodyPlainText;

// HTML Version
string htmlBody = bodyHtml;
AlternateView HTMLV = AlternateView.CreateAlternateVie
  new System.Net.Mime.ContentType("text/html"));

// iCal
ICalendar iCal = new iCalendar();
iCal.Method = METHOD;
iCal.ProductID = "My Metting Product";

// Create an event and attach it to the iCalendar.
Event evt = iCal.Create<Event>();
evt.UID = uid.ToString();

evt.Class = "PUBLIC";
// Needed by Outlook
```

```
evt.Created = new iCalDateTime(DateTime.Now);

evt.DTStamp = new iCalDateTime(DateTime.Now);
evt.Transparency = TransparencyType.Transparent;

// Set the event start / end times
evt.Start = new iCalDateTime(2014, 10, 3, 8, 0, 0);
evt.End = new iCalDateTime(2014, 10, 3, 8, 15, 0);
evt.Location = location;

//var organizer = new Organizer("the.organizer@myCompa
//evt.Organizer = organizer;

// Set the longer description of the event, plain text
evt.Description = bodyPlainText;

// Event description HTML text
// X-ALT-DESC;FMTTYPE=text/html
var prop = new CalendarProperty("X-ALT-DESC");
prop.AddParameter("FMTTYPE", "text/html");
prop.AddValue(bodyHtml);
evt.AddProperty(prop);

// Set the one-line summary of the event
evt.Summary = VisBetreff;
evt.Priority = priority;

//--- attendes are optional
IAttendee at = new Attendee("mailto:Peter.Black@MyOrg.
at.ParticipationStatus = "NEEDS-ACTION";
at.RSVP = true;
at.Role = "REQ-PARTICIPANT";
evt.Attendees.Add(at);

// Let's also add an alarm on this event so we can be
Alarm alarm = new Alarm();

// Display the alarm somewhere on the screen.
alarm.Action = AlarmAction.Display;

// This is the text that will be displayed for the ala
alarm.Summary = "Upcoming meeting: " + VisBetreff;

// The alarm is set to occur 30 minutes before the eve
```

```
alarm.Trigger = new Trigger(TimeSpan.FromMinutes(-30))

//--- Attachments
string filename = "Test.docx";

// Add an attachment to this event
IAttachment attachment = new DDay.iCal.Attachment();
attachment.Data = ReadBinary(@"C:\temp\Test.docx");
attachment.Parameters.Add("X-FILENAME", filename);
evt.Attachments.Add(attachment);

iCalendarSerializer serializer = new iCalendarSerializ
serializer.Serialize(iCal, filepath);

// the .ics File as a string
string iCalStr = serializer.SerializeToString(iCal);

// .ics as AlternateView (used by Outlook)
// text/calendar part: method=REQUEST
System.Net.Mime.ContentType calendarType =
  new System.Net.Mime.ContentType("text/calendar");
calendarType.Parameters.Add("method", METHOD);
AlternateView ICSview =
  AlternateView.CreateAlternateViewFromString(iCalStr,

// Compose
message.AlternateViews.Add(HTMLV);
message.AlternateViews.Add(ICSview); // must be the la

// .ics as Attachment (used by mail clients other than
Byte[] bytes = System.Text.Encoding.ASCII.GetBytes(iCa
var ms = new System.IO.MemoryStream(bytes);
var a = new System.Net.Mail.Attachment(ms,
  "VIS-Termin.ics", "text/calendar");
message.Attachments.Add(a);

// Send Mail
SmtpClient client = new SmtpClient();
client.Send(message);
```

Here the ReadBinary() function:

```
private static byte[] ReadBinary(string fileName)
{
    byte[] binaryData = null;
    using (FileStream reader = new FileStream(fileName
      FileMode.Open, FileAccess.Read))
    {
        binaryData = new byte[reader.Length];
        reader.Read(binaryData, 0, (int)reader.Length)
    }
    return binaryData;
}
```

Its easiest to configure the SmtpClient in the config file like this:

```
<configuration>
  ...
  <system.net>
    <mailSettings>
      <smtp>
        <network host="mysmtp.server.com" port="25" us
password="myPassword" />
      </smtp>
    </mailSettings>
  </system.net>
  ...
```

Share Improve this answer

Follow

edited Oct 7, 2021 at 5:54

Community Bot

1 ● 1

answered Oct 4, 2014 at 16:34

DrKoch

9,752 ● 2 ● 35 ● 44

Thanks for such a thorough response! Saves a lot of trial and error – Tom Carver Apr 23, 2016 at 12:40