simple-talk

DATA, DEV, ADMIN, CHOP SUEY

LATEST SQL, .NET AND SYS ADMIN ARTICLES, OPINION AND NEWS

Brush Your Teeth and Clean Your Room

Published Thursday, May 12, 2011 5:12 AM

Hello, Grant Fritchey here, one of these Scary DBAs. Tony isn't around at the moment so I stopped by to talk for minute. Yeah, I could go on about some of those cool and controversial topics that Tony always brings up, but I'm a DBA. By definition that makes me dull and irritable. I've noticed that a lot of you, a **lot** of you, have not been doing a good job at brushing your teeth and cleaning your room. By that I mean you're not running a good set of backups and testing them to be sure they work well. Yeah, I know, I've got two teenagers and when either phrase comes out of my mouth, "Hey, did you brush your teeth," or "Clean your room before you go with your friends" inevitably the groans and moans start. Same thing with so many IT Pros these days, "Have you set up log backups" and "Did you test your backups and backup process" elicit all kinds of groaning about "We don't have time" and "We tested it 18 months ago and nothing has changed" and "But I'll clean my room later". wait, that was my kids again.

Look, it's not my job to help you keep yours. oh, it actually sort of is. You enjoy bringing home that paycheck, right? Then you need to set up a good backup plan and most importantly, you need to test it. Testing it doesn't just mean you run RESTOREVERIFYONLY or something. It means you run a full restore, and not just through automated scripts. You need to know that your backup is good and you need to know that you can retrieve that backup. The number of times I see posts on forums talking about some failure or other that people have experienced, but, no, they don't have a backup, why do you ask, is enough to make you crazy. It's even worse when you hear the stories about people who set up the backup process but haven't checked it since and it's been failing for six months and they didn't know so what do they do now, or they have the backup but have never done a point in time recovery how does that work and can you hurry with the explanation because production is down and the boss and her boss and his boss are all standing in my cubicle.

My statement to you: set up your backup processes and test those processes. I have two questions for you and I'd love to get some good answers to these. Why don't people set up good backups and what do we, the IT community, have to do to get the word out to everyone, because we're not getting that job done.

by Grant Fritchey

PATINDEX Workbench

12 May 2011
by Phil Factor

> */*The PATINDEX function of SQL Server packs powerful magic, but it is easy to get it wrong. Phil Factor returns to the Workbench format to give a tutorial of examples, samples and cookbook ideas to demonstrate the ways that this underrated function can be of practical use. It is intended to be pasted into SSMS and used as a basis for experiment.*/*

```
/* The PATINDEX function is a treasure, and it is the obvious way in SQL Server to locate text in
strings. It uses the same wildcard pattern as the LIKE operator, works as fast, but gives back
more information. It is closest in functionality to CHARINDEX. People worry that it does not allow
a full RegEx, but that is missing the point. It will do useful string searching, as I'll show you,
and it is optimized for performance in a database. A RegEx search performs a different service and
is slower in doing the routine jobs that PATINDEX does so well.

It is great for quick checks on your data. Here, we do an elementary check on a stored  IP address
*/
SELECT PATINDEX ('%.%.%.%','23.245.6.49')--returns with the index of the first dot or 0 if there
aren't three in the string
SELECT PATINDEX ('%.%.%.%','23.245.6,49')--returns 0
/* so here the % 'wildcard'  character has a special meaning, which is 'any number of any
character'. BOL puts this as 'Any string of zero or more characters.' For some reason, most
developers seem to think that there is a rule that you can only use them at the start and/or end
of a pattern. Not true. You can use them anywhere, as many as you like, but not adjacent as these
are then ignored. 'One of any character' is denoted by the underscore wildcard character, '_'. So
what does the IP check we have shown you comprise? Nothing more than counting that there are three
dots in the string. We can do more but it gets rather more complicated.
note that, because we are just testing for existence, and aren't going to extract it, we could
also use LIKE */
SELECT case  WHEN '23.245.6.49' LIKE '%.%.%.%' THEN 1 ELSE 0 END
/*What if you want to specify the % character instead of using it in its wildcard meaning? Simple.
Do this [%] like here.*/
SELECT PATINDEX ('%[%]%','You have scored 90% in your exam')
/*and the same applies to the other wildcard characters*/
SELECT PATINDEX (
     '%[_][%][[]%',
     'You have scored 90% in your exam and shouted "D_%[]!"')
/*
Here are a few other more practical examples of using the '%' wildcard.

Select a list of all objects in the current database whose name begins with 'sp' and ends with
'table' (case insensitive)*/
SELECT name FROM sys.objects
  WHERE PATINDEX('sp%Table',NAME COLLATE Latin1_General_CI_AI)>0
/*Select all objects (not tables!) in the current database which have the following three words
FROM WHERE and ORDER in that order with gaps in between*/
SELECT name FROM sys.Objects
  WHERE PATINDEX ('%FROM%WHERE%AND%',object_definition(object_ID))>0
/*List all LATIN collations that are case and accent insensitive (ignoring case) */
SELECT name,description FROM fn_helpcollations()
  WHERE PATINDEX ('latin%case-insensitive%accent-insensitive%',
     [description] COLLATE Latin1_General_CI_AI)>0
/* you'll notice that we need to use the COLLATE keyword to enforce the type of search. It may not
make a difference if your database is already set to a suitable collation, but then database
collation can change!
This becomes more important if we specify the range of allowable characters.
We can specify the range of characters to look for by listing them between angle-brackets. For
example, in our IP search, we can do an obvious improvement by making sure there are at least one
number before the dots! */
SELECT PATINDEX ('%[0-9].%[0-9].%[0-9].%','278.2.6.49')--returns with the index of the first dot
or 0 if there aren't three in the string
SELECT PATINDEX ('%[0-9].%[0-9].%[0-9].%','278.A.6.49')--returns 0
```

```
/*Nice, but you can do it with the LIKE command. Where PATINDEX scores is where you need to
extract the information.
We can easily pick up a number embedded in other characters
if we were lucky to be given the task of picking up three-digit numbers from a string, (or zero if
there aren't any) that is ridiculously easy and can be done inline within a sql query */
DECLARE @SampleString VARCHAR(255)
SELECT @SampleString=' the current valve weight is not 56 mg as before, but 067 milligrams'
SELECT SUBSTRING(
   @SampleString+'000 ',--put the default on the end
   PATINDEX('%[^0-9][0-9][0-9][0-9][^0-9]%',@SampleString+'000 ')+1,
   3)--three characters
-- 067
/* See what we've done. We've added a default at the end so that we don't have to cope with
passing back a 0 from the PATINDEX when it hits a string without the correctly formatted number in
it. We look for the transition between a character that isn't numeric, to one that is. Then we
look for three valid consecutive numbers followed by a character that isn't a number.
The same technique can be used where you want to trim off whitespace before or after a string. You
might think that RTRIM and LTRIM do this but they are slightly broken, in that they only trim off
the space character. What about linebreaks or tabs? */
Declare @PaddedString VARCHAR(255), @MatchPattern VARCHAR(20)
SELECT  @PaddedString='

            Basically we just want this
                  ',
         @Matchpattern='%[^' + CHAR(0)+'- ]%'
/* this match pattern looks for the first occurrence in the string of a character that isn't a
control character. You'll need to specify a binary sort order to be certain that this works, so we
use the COLLATE clause to specify that we want a binary collation that understands that control
characters range from 0 to 32 (space character) We'd normally want to add all the other space
characters such as non-break space.*/

--now this will find the index of the start of the string
SELECT PATINDEX(@Matchpattern,@PaddedString collate SQL_Latin1_General_CP850_Bin)
/*

And we can easily then use this to create a function that really trims a string.
*/
IF OBJECT_ID(N'Trim') IS NOT NULL
  DROP FUNCTION Trim
GO
CREATE FUNCTION Trim
/**
summary:   >
This procedure returns a string with all leading and trailing blank space removed. It is similar
to the TRIM functions in most current computer languages. You can change the value of the string
assigned to @BlankRange, which is then used by the PATINDEX function. The string can be a
rangee.g. a-g or a list of characters such as abcdefg.

Author: Phil Factor
Revision: 1.1 changed list of control character to neater range.
Revision: 1.2 added explicit collation.
date: 28 Jan 2011
example:
     - code: dbo.Trim('  678ABC    ')
     - code: dbo.Trim('  This has leading and trailing spaces  ')
     - code: dbo.Trim('  left-Trim This')
     - code: dbo.Trim('Right-Trim This      ')
returns:   >
Input string without trailing or leading blank characters, however these characters are defined in
@BlankRange

**/ (@String VARCHAR(MAX))
RETURNS VARCHAR(MAX)
AS
BEGIN
  DECLARE @BlankRange CHAR(255),
    @FirstNonBlank INT,
    @LastNonBlank INT
  IF @String IS NULL
```

```
    RETURN NULL--filter out null strings
  SELECT  @BlankRange = CHAR(0)+'- '+CHAR(160)
  /* here is where you set your definition of what constitutes a blank character. We've just
chosen every 'control' character, the space character and the non-breaking space. Your
requirements could be different!*/
  SELECT  @FirstNonBlank = PATINDEX(
                              '%[^' + @BlankRange + ']%',
                              @String  collate SQL_Latin1_General_CP850_Bin)
  SELECT  @lastNonBlank = 1
                + LEN(@String + '|')
                - (PATINDEX(
                    '%[^' + @BlankRange + ']%',
                    REVERSE(@String)  collate SQL_Latin1_General_CP850_Bin))
  IF @FirstNonBlank > 0
    RETURN SUBSTRING(@String,@FirstNonBlank, @LastNonBlank-@firstNonBlank)
  RETURN '' --nothing would be left
END
GO

/* PATINDEX allows you to do some subtle things such as chopping strings into words. We'll start
by doing something really simple such as chopping up a series of numbers into a table of numbers
*/

DECLARE @string VARCHAR(255),@start INT, @LenString INT, @End int
DECLARE @numbers table (number NUMERIC (9,4))
SET NOCOUNT on
SELECT @String='   23 455  5.789 45.0 67 06978 000  ',
       @Start=PATINDEX( '%[^0-9.][0-9.]%',@string),
       @LenString=LEN(@string+'|')-1
WHILE @start<@lenString
  begin
  SELECT @end=PATINDEX(
                '%[0-9.][^0-9.]%',
                SUBSTRING(@String,@Start,@LenString)+' ')
  insert INTO @numbers (number) SELECT SUBSTRING(@String,@start,@end)
  SELECT @Start=@start
               +@End
               +PATINDEX(
                   '%[^0-9.][0-9.]%',
                   SUBSTRING(@String,@start+@end,@LenString)+'0')
  end
SELECT * FROM @number

/* ALL we're doing here is defining what we think a character within a valid number is in terms of
a wildcard, anything from 0 to 9, or a dot in this instance, and looking for the transitions
between 'number and 'not-number' */

/* Let's try something a bit trickier, and closer to a real chore. Let's find a UK postcode.
(apologies to all other nations who are reading this) */

SELECT PATINDEX('[A-Z][A-Z0-9]% [0-9][ABD-HJLNP-UW-Z][ABD-HJLNP-UW-Z]','CB4 0WZ')--0 if invalid.
>0, 1 if valid
/*This needs some explanation. the '[' and ']' brackets enclose a list of characters. you can list
them, without commas, or specify a range. Here, in the last two expressions, we have done both.
[ABD-HJLNP-UW-Z] is a shorthand for [ABDEFGHJLNPQRSTUWXYZ]. This looks laborious, but works
speedily, since SQL Server works hard to optimize LIKE and PATINDEX

The validation rules are that the length must be between 6 and 8 characters of which one is a
space. This divides the three-character local code to the right of the space from the sorting-
office code to the left of the space. The local characters are always a numeric character followed
by two alphabetic characters. The Sorting Office code the left of the gap, can be between 2 and 4
characters  and the first character must be alpha.
Before you get too excited, I must point out the the postcode validation is more complex. We can't
use it because PATINDEX uses only wildcards and hasn't the OR expression or the iterators
We can do quite well though...

This does everything but validate that the sorting office code is between two and four characters.
If you were determined to do this, you'd need to run three checks*/
'[A-Z][A-Z0-9] [0-9][ABD-HJLNP-UW-Z][ABD-HJLNP-UW-Z]'
```

```
'[A-Z][A-Z0-9]_ [0-9][ABD-HJLNP-UW-Z][ABD-HJLNP-UW-Z]'
'[A-Z][A-Z0-9]__ [0-9][ABD-HJLNP-UW-Z][ABD-HJLNP-UW-Z]'

--Which, if we just wanted to check the validity, we can do in a number of ways
Select case WHEN 'CB4 0WZ' like '[A-Z][A-Z0-9] [0-9][ABD-HJLNP-UW-Z][ABD-HJLNP-UW-Z]'
or 'CB4 0WZ' like '[A-Z][A-Z0-9]_ [0-9][ABD-HJLNP-UW-Z][ABD-HJLNP-UW-Z]'
or 'CB4 0WZ' like '[A-Z][A-Z0-9]__ [0-9][ABD-HJLNP-UW-Z][ABD-HJLNP-UW-Z]' then 1 else 0 end

--With 2008, we can do this
SELECT MAX(PATINDEX([matched],'CB4 0WZ'))
  FROM (VALUES
     ('[A-Z][A-Z0-9] [0-9][A-Z][A-Z]')
     ,('[A-Z][A-Z0-9]_ [0-9][A-Z][A-Z]')
     ,('[A-Z][A-Z0-9]__ [0-9][A-Z][A-Z]')) AS f([Matched])
)      --0 if invalid. >0, 1 if valid


/* What if you wanted to do the more common chore of extracting the postcode from an address-line
and putting it in its own field? This is where you have to stop using LIKE as it won't cut the
mustard. If you are of a nervous disposition in your SQL-writing please turn away now.*/

Select stuff([address],start+1,length-start-fromend,''),
       Substring([address],start,length-start-fromend)
from
(--we have a derived table with the results we need for the chopping
SELECT MAX(PATINDEX([matched],[address])) as start,
       MAX(PATINDEX([ReverseMatch],reverse([address]+' ')))-1 as fromEnd,
       len([address]+'|') as [length],
       [Address]
FROM (VALUES--first the forward match, then the reverse match
     ('% [A-Z][A-Z0-9] [0-9][A-Z][A-Z]%', '%[A-Z][A-Z][0-9] [A-Z0-9][A-Z] %' )
     ,('% [A-Z][A-Z0-9][A-Z0-9] [0-9][A-Z][A-Z]%', '%[A-Z][A-Z][0-9] [A-Z0-9][A-Z0-9][A-Z] %')
     ,('% [A-Z][A-Z0-9][A-Z0-9][A-Z0-9] [0-9][A-Z][A-Z]%', '%[A-Z][A-Z][0-9] [A-Z0-9][A-Z0-9][A-
Z0-9][A-Z] %'))
     AS f([Matched],ReverseMatch)
cross join --normally this would be a big table, of course
    (select 'Simple Talk Publications, Newnham House, Cambridge Business Park,
Cambridge, CB10 7EC' union all Select '20 Milton Street, Inverness  SWB7 7EC' ) as g([Address])
group by [address]
having MAX(PATINDEX([matched],[address]))>0) work

--and this technique allows you to process a huge table. It is surprisingly fast.
--first we create our sample table...
Create table [Addresses]
(Address_ID int identity primary key,
[Address] varchar(200) not null,
[alteredAddress] Varchar(200) null,
Postcode varchar(12))
/* ...and then we'll populate it with 100,000 rows via SQL Data Generator (we'll use a RegEx to
fill the address column in).
```

	Address_ID	Address	alteredAddress	Postcode
1	1	The Lodge, 14 Cathedral Avenue, Belfast CN8B 9RR UK	NULL	NULL
2	2	Lilac Cottage, 903 Lower Acacia Lane, Leeds VVP 6AZ UK	NULL	NULL
3	3	The Barn, 52 Cathedral Road, Wakefield RHCI 1JR UK	NULL	NULL
4	4	Yew Tree Cottage, 94 North Sebastobol St., Cambridge IJZV 7FC UK	NULL	NULL
5	5	The Nook, 794 Sebastobol Parkway, Liverpool KDMU 8UC UK	NULL	NULL
6	6	531 Wellington Road, Salisbury H29F 7ED UK	NULL	NULL
7	7	68 High Drive, Coventry A02A 2CY UK	NULL	NULL
8	8	The Bungalow, 86 East Church St., Worcester E3 6GA UK	NULL	NULL
9	9	Hillside, 61 Church Avenue, Ripon EZ 0JV UK	NULL	NULL
10	10	52 Cathedral Lane, Dundee N2D 8BS UK	NULL	NULL
11	11	645 Cathedral Park, Lichfield IO 2ZI UK	NULL	NULL
12	12	584 Lower Acacia St., Bath DA 1CM UK	NULL	NULL
13	13	Corner Cottage, 49 Old Way, Newcastle upon Tyne CO 3BN UK	NULL	NULL
14	14	The Firs, 566 Manor Avenue, Exeter PDLB 7DR UK	NULL	NULL
15	15	The Cottage, 290 Wellington Road, Truro CRI3 1BE UK	NULL	NULL
16	16	Holly Cottage, 616 High Avenue, Cambridge FE6 4AG UK	NULL	NULL
17	17	78 West New Avenue, Truro M7 9EC UK	NULL	NULL
18	18	64 South Acacia Avenue, Nottingham R7 0CW UK	NULL	NULL
19	19	47 Wellington Road, Sheffield A58 2LQ UK	NULL	NULL
20	20	The Cottage, 13 High St., Oxford QFE 3GH UK	NULL	NULL
21	21	Highfield, 591 Acacia Mews, Armagh FG 2CM UK	NULL	NULL
22	22	176 New Mews, Londonderry PSYB 2MF UK	NULL	NULL
23	23	26 Manor Mews, Newport DL7UP UK	NULL	NULL

```
Hopefully, I'll remember to put it in the speech bubble at the head of the article for the other
SQLDG freaks). Then we are going to pull out the postcode information, place the modified address
without the postcode in a second column, and put the extracted postcode into its own column so we
can subsequently do lightning searches based on postcode. This whole messy process runs in five
seconds on my test machine. If you did a neat cursor-based process, it would take minutes.*/

Update Addresses
  Set alteredAddress=[modified],
    Postcode=[extracted]
from Addresses inner join
  (Select
        Address_ID,--the address ID
        --the modified address without the postcode (if there was one!)
        stuff([address],start,length-start-fromend+2,'') as [modified],
        --the postcode itself
        Substring([address],start,length-start-fromend+2) as [extracted]
  from
  (--we have a derived table with the results we need for the chopping
   --process to save having to calculate it more than once
      SELECT MAX(PATINDEX([matched],[address])) as start,
             MAX(PATINDEX([ReverseMatch],reverse([address]+' '))-1) as fromEnd,
             len([address]+'|')-1 as [length],
             [Address] as [address],
             min(Address_ID) as address_ID
        FROM (VALUES--first the forward match, then the reverse match
           ('% [A-Z][A-Z0-9] [0-9][A-Z][A-Z]%', '%[A-Z][A-Z][0-9] [A-Z0-9][A-Z] %' )
          ,('% [A-Z][A-Z0-9][A-Z0-9] [0-9][A-Z][A-Z]%', '%[A-Z][A-Z][0-9] [A-Z0-9][A-Z0-9][A-Z]
%')
          ,('% [A-Z][A-Z0-9][A-Z0-9][A-Z0-9] [0-9][A-Z][A-Z]%', '%[A-Z][A-Z][0-9] [A-Z0-9][A-Z0-
9][A-Z0-9][A-Z] %'))
          AS f([Matched],ReverseMatch)
      cross join addresses
      group by [address]
      having MAX(PATINDEX([matched],[address]))>0)
    work)
alteredData
on AlteredData.Address_ID=Addresses.Address_ID
```

	Address	alteredAddress	Postcode
1	The Lodge, 14 Cathedral Avenue, Belfast C...	The Lodge, 14 Cathedral Avenue, Belfast UK	CN8B 9RR
2	Lilac Cottage, 903 Lower Acacia Lane, Lee...	Lilac Cottage, 903 Lower Acacia Lane, Leeds UK	VVP 6AZ
3	The Barn, 52 Cathedral Road, Wakefield R...	The Barn, 52 Cathedral Road, Wakefield UK	RHCI 1JR
4	Yew Tree Cottage, 94 North Sebastobol St.,...	Yew Tree Cottage, 94 North Sebastobol St., Cambr...	IJZV 7FC
5	The Nook, 794 Sebastobol Parkway, Liverp...	The Nook, 794 Sebastobol Parkway, Liverpool UK	KDMU 8UC
6	531 Wellington Road, Salisbury H29F 7ED ...	531 Wellington Road, Salisbury UK	H29F 7ED
7	68 High Drive, Coventry A02A 2CY UK	68 High Drive, Coventry UK	A02A 2CY
8	The Bungalow, 86 East Church St., Worces...	The Bungalow, 86 East Church St., Worcester UK	E3 6GA
9	Hillside, 61 Church Avenue, Ripon EZ 0JV ...	Hillside, 61 Church Avenue, Ripon UK	EZ 0JV
10	52 Cathedral Lane, Dundee N2D 8BS UK	52 Cathedral Lane, Dundee UK	N2D 8BS
11	645 Cathedral Park, Lichfield IO 2ZI UK	645 Cathedral Park, Lichfield UK	IO 2ZI
12	584 Lower Acacia St., Bath DA 1CM UK	584 Lower Acacia St., Bath UK	DA 1CM
13	Corner Cottage, 49 Old Way, Newcastle up...	Corner Cottage, 49 Old Way, Newcastle upon Tyn...	CO 3BN
14	The Firs, 566 Manor Avenue, Exeter PDLB ...	The Firs, 566 Manor Avenue, Exeter UK	PDLB 7DR
15	The Cottage, 290 Wellington Road, Truro C...	The Cottage, 290 Wellington Road, Truro UK	CRI3 1BE
16	Holly Cottage, 616 High Avenue, Cambridge...	Holly Cottage, 616 High Avenue, Cambridge UK	FE6 4AG
17	78 West New Avenue, Truro M7 9EC UK	78 West New Avenue, Truro UK	M7 9EC
18	64 South Acacia Avenue, Nottingham R7 ...	64 South Acacia Avenue, Nottingham UK	R7 0CW
19	47 Wellington Road, Sheffield A58 2LQ UK	47 Wellington Road, Sheffield UK	A58 2LQ
20	The Cottage, 13 High St., Oxford QFE 3GH...	The Cottage, 13 High St., Oxford UK	QFE 3GH
21	Highfield, 591 Acacia Mews, Armagh FG 2...	Highfield, 591 Acacia Mews, Armagh UK	FG 2CM
22	176 New Mews, Londonderry PSYB 2MF ...	176 New Mews, Londonderry UK	PSYB 2MF
23	26 Manor Mews, Newport DI 7UP UK	26 Manor Mews, Newport UK	DI 7UP
24	The Stables, 741 High St., Aberdeen G5S ...	The Stables, 741 High St., Aberdeen UK	G5S 8II
25	Woodside, 15 Milton Lane, Oxford R0C 3C...	Woodside, 15 Milton Lane, Oxford UK	R0C 3CB

```
/* so here is a puzzle to end off with. You have a field with an email address  somewhere in it,
and you need to extract it. Here's one way of pulling it out. It looks a bit complicated, but it
is fast */

SELECT  CASE WHEN AtIndex=0 THEN '' --no email found
           ELSE RIGHT(head, PATINDEX('% %', REVERSE(head) + ' ') - 1)
        + LEFT(tail + ' ', PATINDEX('% %', tail + ' '))
        END EmailAddress
FROM (SELECT RIGHT(EmbeddedEmail, [len] - AtIndex) AS tail,
             LEFT(EmbeddedEmail, AtIndex) AS head, AtIndex
      FROM (SELECT PATINDEX('%[A-Z0-9-]@[A-Z0-9-]%', EmbeddedEmail+' ') AS AtIndex,
                   LEN(EmbeddedEmail+'|')-1 AS [len],
                   embeddedEmail
            FROM (
                  SELECT 'The Imperial Oil Company Phil.Factor@ImpOil.com 123 Main St'
                 ) AS ListOfCompanies (EmbeddedEmail)
           )f
     )g
/*
EmailAddress
-------------------------
Phil.Factor@ImpOil.com
*/
/* so let's test it out by recreating our addresses table, and stocking the address column with
aan additional email record. */

--DROP TABLE addresses
Create table [Addresses]
(Address_ID int identity primary key,
[Address] varchar(200) not null,
email Varchar(50))
/* now we stock it with 100,000 records whth an address column with an embedded email address.
Next we extract the email address
 */
Update Addresses
  Set email=emailAddress
from Addresses inner join
(SELECT  CASE WHEN AtIndex=0 THEN '' --no email found
           ELSE RIGHT(head, PATINDEX('% %', REVERSE(head) + ' ') - 1)
        + LEFT(tail + ' ', PATINDEX('% %', tail + ' '))
        END AS emailAddress, Address_ID
```

```
FROM (SELECT RIGHT(Address, [len] - AtIndex) AS tail,
             LEFT(Address, AtIndex) AS head, AtIndex,Address_ID
      FROM (SELECT PATINDEX('%[A-Z0-9-]@[A-Z0-9-]%', Address+' ') AS AtIndex,
                   LEN(Address+'|')-1 AS [len],
                   Address, address_ID
            FROM   Addresses
           )f
     )g
 )emails
 ON emails.address_ID=addresses.Address_ID

/*
So there we have it. In summary

When using PATINDEX,
• specify the collation if you use character ranges
• If a problem seems tricky, see if you can detect transitions between character types
• use the angle brackets to 'escape' wildcard characters.
• Think of the % wildcard as meaning 'any number of any character', or 'Any string of zero or more
  characters.'
• remember that you can specify the wildcard parameter as a column as well as a literal or
  variable.
• you can use cross joins to do multiple searches to simulate the OR condition of RegExes.
• experiment wildly when you get a spare minute. Occasionally, you'll be surprised.

*/
```

Office365 and Lync Online

10 May 2011
by Jaap Wesselius

> Having gotten us up and running with the basics of Office365, Jaap Wesselius now dives deeper, walking us through the setup process for Lync Online, the Office365 component which provides cloud-based Unified Communications. He even shows us how to integrate it with our on-premises systems in quick, easy steps.

In my previous article on Simple-Talk, I explained how to setup Exchange Online in Office365. As I demonstrated, it is really easy to set this up and start using either the Outlook Web App, and Outlook 2007, or Outlook 2010. As we also saw, Lync Online is one of the powerful services being offered as part of the Office365 product suite, and now we're going to take a closer look at setting that up.

If you've never used Lync before, then you should know that Lync is a core part of Microsoft's Unified Communications solution, pulling together voice calls, video calls, instant messages, live meetings and shared whiteboard sessions into a single interface. If this sounds familiar, that's because Lync is the next evolution of Office Communications Server.

If you're already familiar with regular Lync, the online flavor provides all of this functionality (although full telephony will not be available to start with), but with the added bonus of the services being hosted in Microsoft's cloud, so you don't have to worry about installation and maintenance.

If you're still reading, then that probably sounds like an attractive proposition to you! In this article, I will demonstrate how to correctly set up Lync Online in just 15 minutes, including using the Microsoft Online Services Connector to configure the client software running on the workstation. I'll also give you a taste of what this great product makes possible. This is going to be aimed at giving you the Office365-specific information you need to get this working, so

> *But first I'd like to do a heads-up warning: Office365 is still in beta which means it is a platform in development. There are things in the Office365 environment that aren't that nice at the moment but will probably be solved before Office365 enters the 'general availability'.*

Domain Management

In my previous article, I registered a domain in Office365, and enabled Exchange Online; to now enable Lync for this domain is literally just a matter of a few mouse clicks. To start, open the Microsoft online portal as the domain administrator, and select **Domains** under the **Management** Section (on the left of the screen):

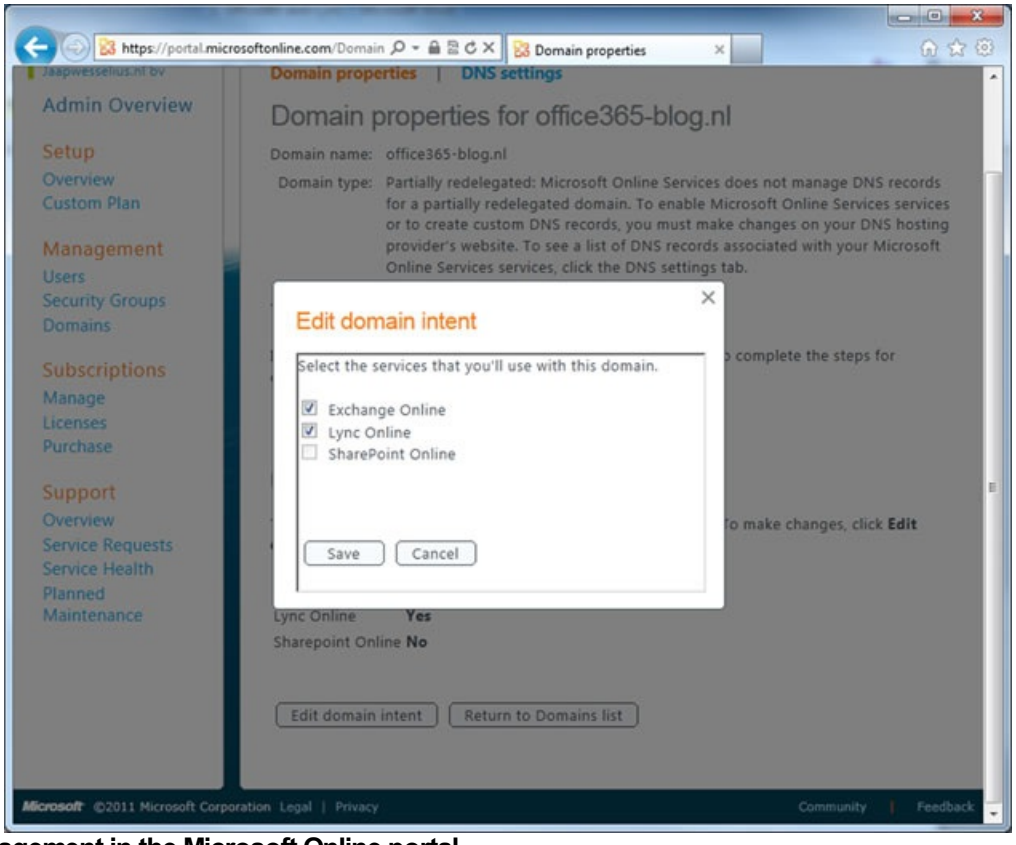


Figure 1. Domain Management in the Microsoft Online portal

Using the **Edit domain intent** button, additional services like Lync Online and Sharepoint Online can be enabled. Once you've selected the appropriate check-boxes and the domain is Lync-enabled, the DNS settings can be checked. Handily, the portal will show the exact settings for SIP

and Federation of the external DNS.

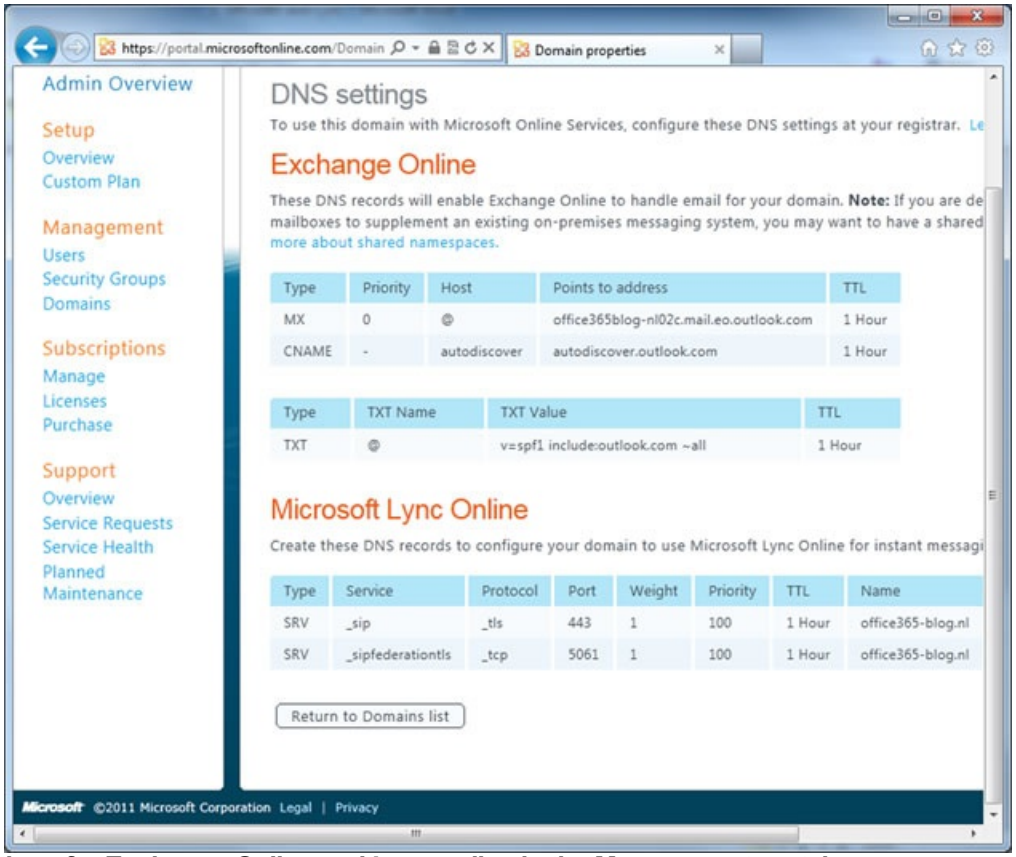


Figure 2. The DNS settings for Exchange Online and Lync online in the Management portal

Setting service records (SRV records) in an Active Directory integrated DNS (like in an internal Active Directory domain) is not that difficult, but for an external DNS it can be tricky. It also varies depending on provider and management interface.

In our example, the **office365-blog.nl** domain is hosted with a provider based in Holland, who is using a Linux DNS with their own interface. I did a bit of experimentation, and in the end it turned out that these were the settings (in this specific scenario!) that needed to be set:

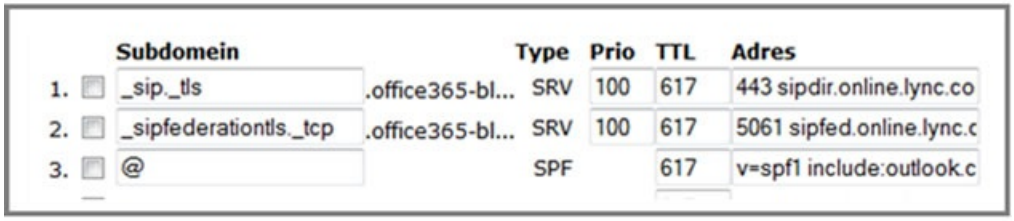


Figure 3. Setting the Lync Online DNS records in a Linux based DNS

If you're going to have to do a little bit of testing to get this working, then you're probably wondering "*Howdo I check if my DNS settings are correct for Lync Online?*" Luckily for you, in the management portal you'll find an option called **Troubleshoot Domain**. It's on the same page as the **Edit Domain Intent** option in Figure 1, and that's where you'll find an option to check the DNS settings. Once you've got everything set up correctly, you should see a message like this:

Figure 4. The external DNS settings are fine according to the management portal

It's also possible to check the DNS settings using the **NSLOOKUP** command line tool. When NSLOOKUP is started, change the type to **SRV** and enter the **FQDN**'s you want to check (see figure 5 for an example).

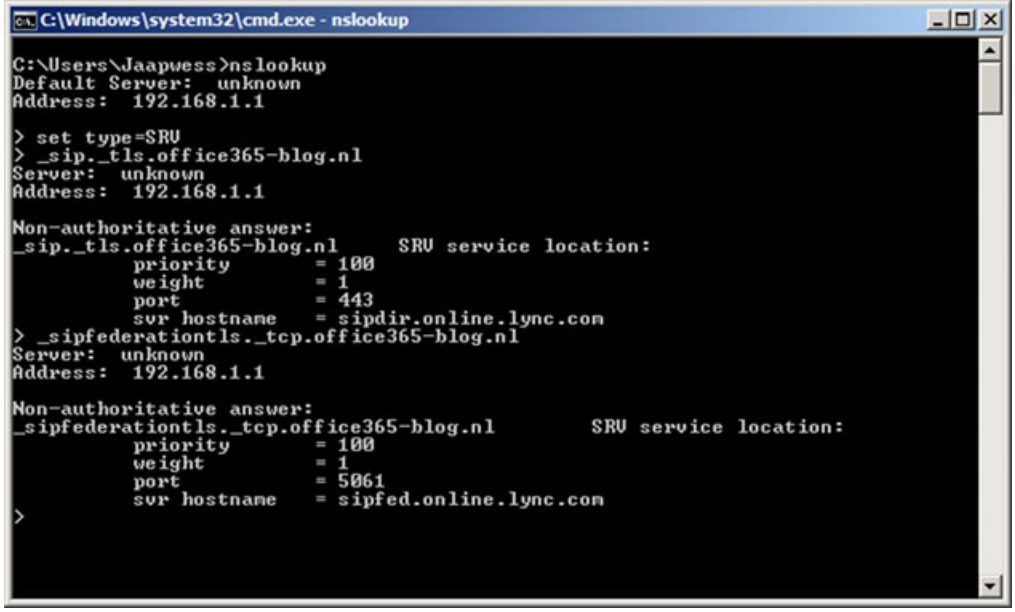


Figure 5. Checking the DNS records using NSLOOKUP

Domain Settings

That's basically all it takes to enable Lync for your domain (I *did* say that it was simple). For comparison, consider the work required to implement even a basic Lync environment on-premises! It's rather refreshing to not have to do all the hard work for once, isn't it?

Now that the services are enabled and the DNS settings are correct, the domain itself can now be configured in detail. Don't worry, it's not nearly as daunting as it sounds. Inthe Management portal, select **Admin Overview**. In the center pane, below the Microsoft Office 365 banner, there's an option for **Lync Online**. Click on the **Manage** button in this section to enter the **Lync Online Control Panel**:

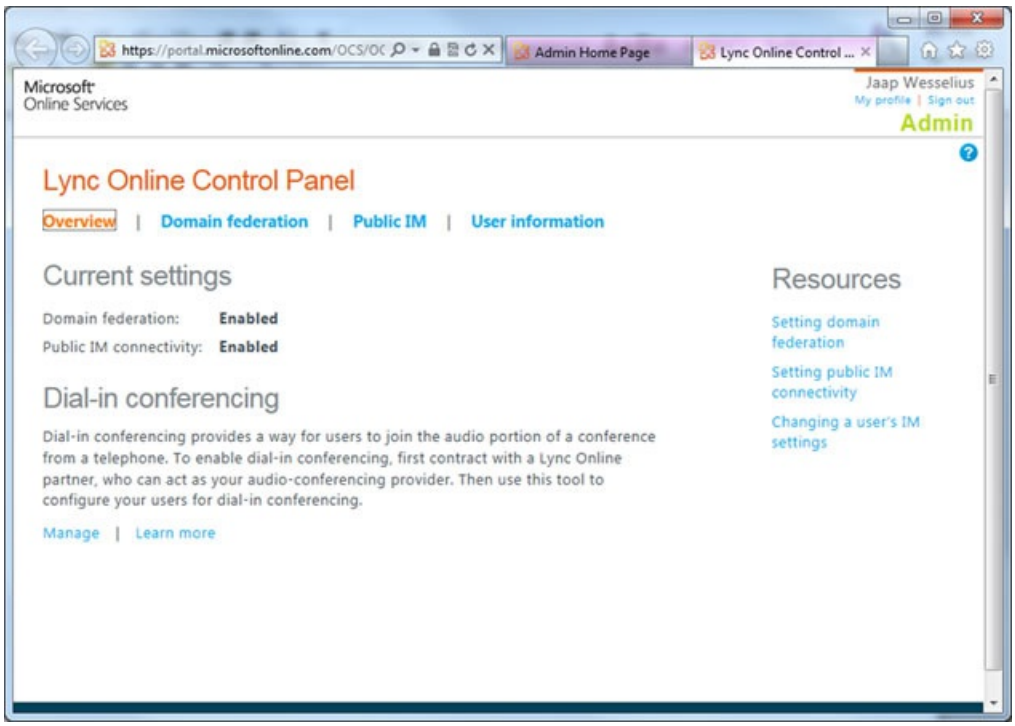


Figure 6. The Lync Online Control Panel

In the Lync Online Control Panel you can control:

- **Domain Federation** – this lets the Lync Online domain communicate with other Lync-enabled domains. These can be regular on-premise Lync environments, or other Lync Online enabled environments. Crucially, the partner you want to federate with either has to have an **open federation,** or they need to add you as a **federation partner**. The default setting for domain federation is **closed**, which means that you have to add every new domain as a federation partner. Of course, it's also possible to opt for you domain federation to be **open**, so that everybody can freely federate through the Lync Online environment. That might sound a bit careless but, if needed, certain domains can be blocked from federation.

- **Public IM** – Using the public IM features, it is possible for the Lync Online enabled domain to interact with public IM networks like MSN or Yahoo networks.

- **User Information** – This section allows you to apply very fine-grained controls on your users' behavior and permissions. For example, it is possible to do things like enable or disable the file transfer, domain federation, public IM, or audio/video settings on a per-user basis.

Unfortunately, although federation between external on-premise Lync Server 2010 and Office Communications Server (OCS) 2007 R2 environments is possible, it is not possible *out of the box*. To make the federating process work, the Office 365 environment needs to be added as a 3rd party IM provider in your on-premise environment. Once this has been done, the whole process works like a charm. For more details on implementing this check out the write-up on my blog. Hopefully this will change (i.e. become easier / automatic) when Office 365 is officially released.

Lync Online Client

Once the domain in fully configured, the client software can be installed on the workstation to make all that communication functionality available to your users. This can be done by your users, but I'll walk you through the process now so that you know what they'll need to do.

To get started, go to the Microsoft Online portal and logon as a user. Under the **Start here** heading, click on **Set up now** in step 1. This will redirect you to the download pages where you can get the client software you need. Of course, if you know what you're doing, you can also go to the download page directly.

The first step is to download and install the Lync Client software; as with most Microsoft downloads, it is possible to select your preferred language and whether you want the 32-bit or the 64-bit version of the software:

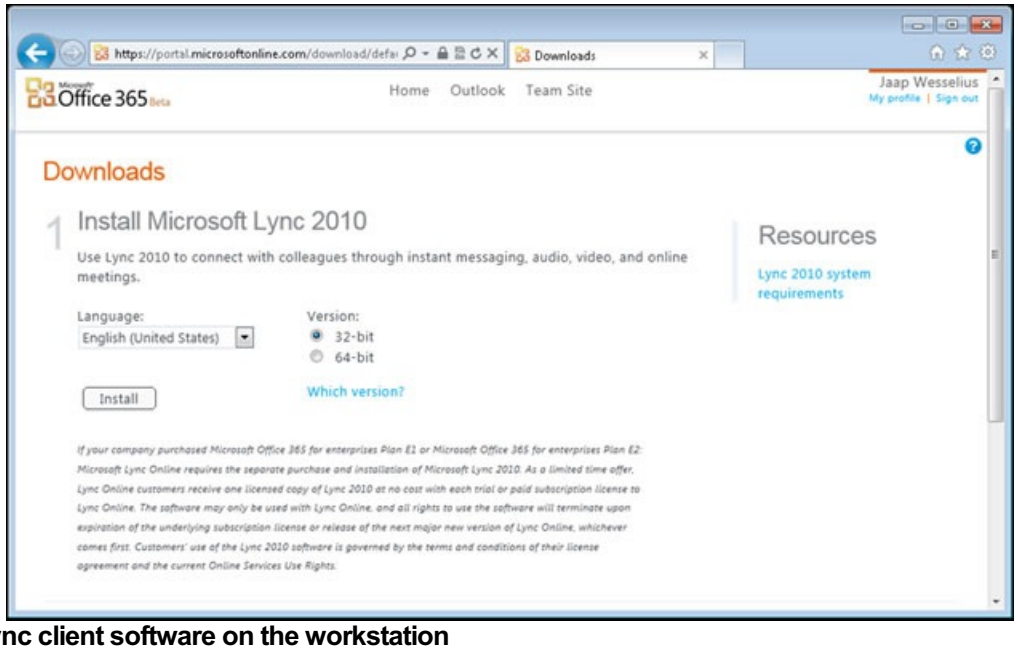


Figure 7. Install the Lync client software on the workstation

The second step is to actually configure the workstation. In the same browser window from which you just downloaded the Lync 2010 client application, scroll down to the bottom of the page (where you'll find step 2) and click the **Set up** button. This will download the **Microsoft Online Services Connector**, sometimes referred to as the **Office365 Desktop Setup application**. This connector is just a small utility which will automatically configure your client software.

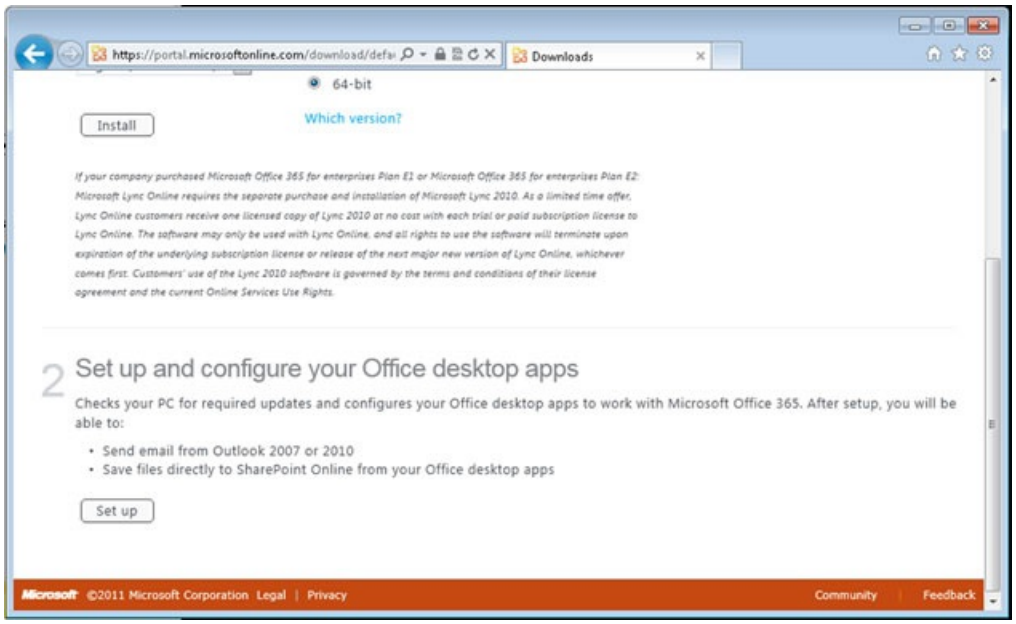


Figure 8. Download and run the Office365 Desktop Setup application to configure the client software

When the connector is started, a security warning is presented, asking you if you *really* want to run this application. Naturally, you just need to select **Run** to continue.

The setup application itself just looks like an Office365 web application. Logon using your normal Office365 user credentials, which were created earlier:

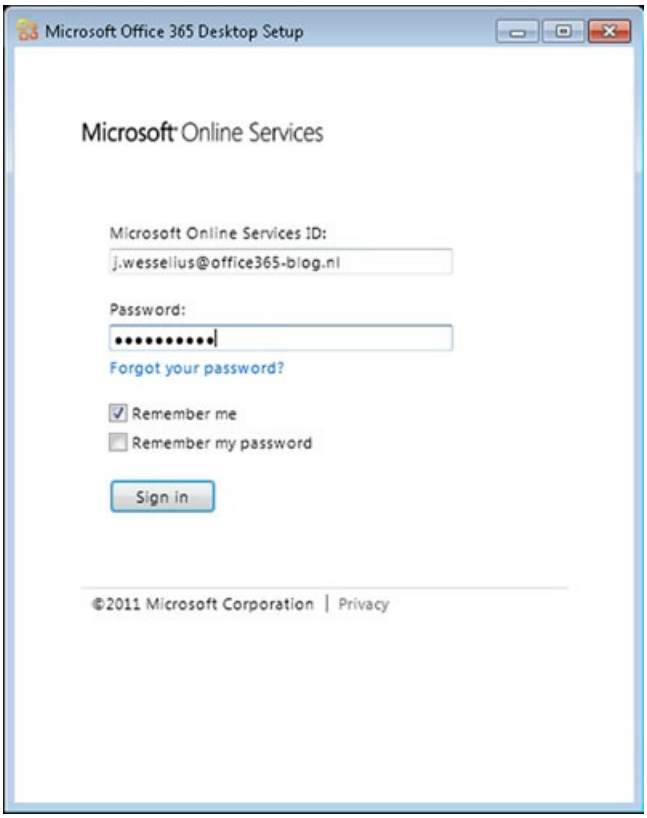


Figure 9: starting the Microsoft Online Services Connector.

The setup application installs a few updates, and configures the workstation to make the most of the Office365 applications (It is possible to select which applications are configured, so you'll always be in full control of what's happening):

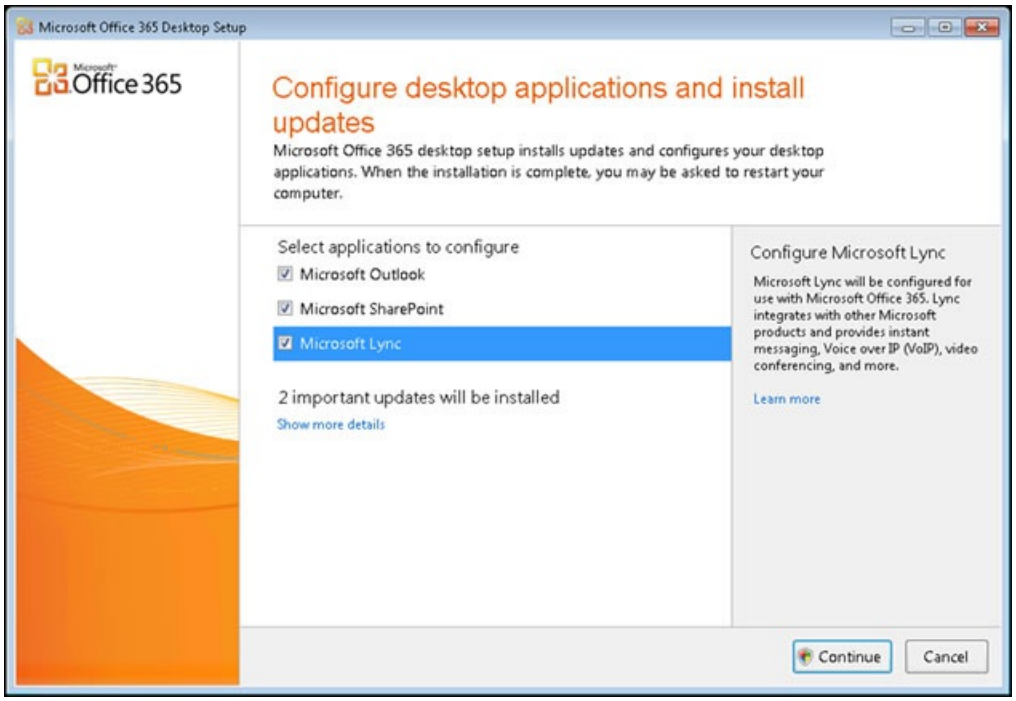


Figure 10: Deciding which applications you want to configure.

Once you're happy with that will be done, just click **Continue** to start installing the necessary fixes and configure the workstation. If the Lync 2010 client application is already running as a result of the previous steps, a warning message is shown - the Lync 2010 application should be closed and exited before the Setup process continues. Just follow the wizard, and if you want more information you can always click on **show more details**:

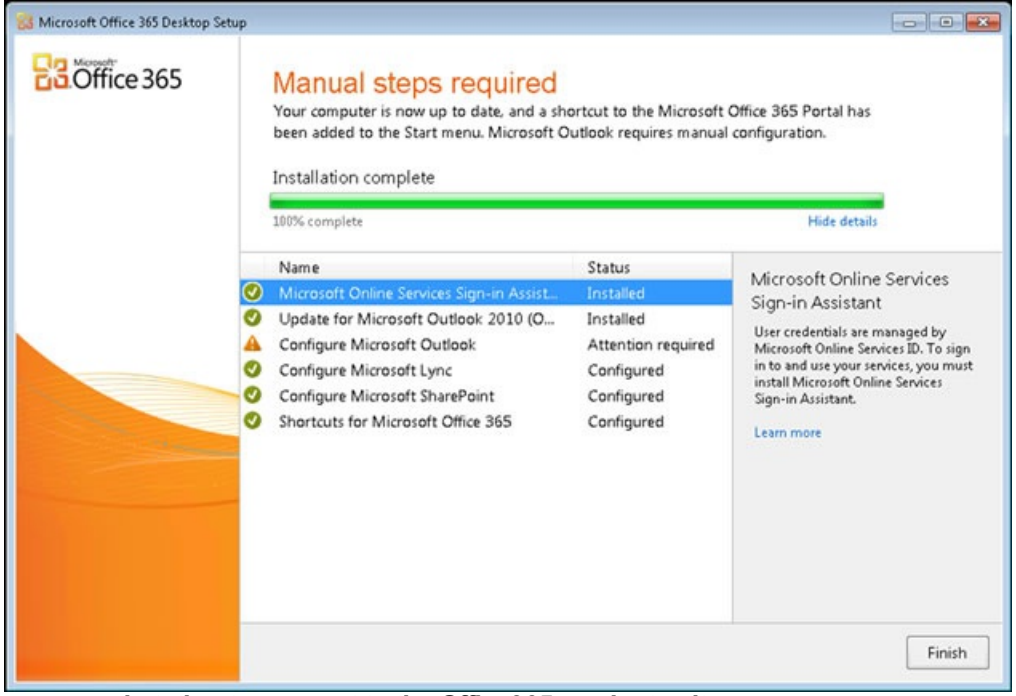


Figure 11: Configuring your workstation to connect to the Office365 product suite.

Once the wizard is finished, an overview provides more details on what has changed, and there are a few things to keep in mind at this stage. For example additional hotfixes for Outlook 2010 will be downloaded and installed, but configuration is still a manual process. This means that, ultimately, an appropriate Outlook profile will still be created manually, although autodiscover can be used to make this easier, as explained in my previous article.

When requested, reboot the workstation, and everything is good to go.

Start the Lync client

If you've used Lync in the past, then this section will be familiar to you. The joy of Lync online is precisely that it behaves the same as on-premises Lync, so don't be surprised if you find this next section a little uninformative.
Now that the setup and reboot has been performed, the Lync 2010 client will now automatically start during the boot sequence of the workstation, so you just need to open the client and click the **Sign in** button. If everything is configured correctly, and it should be, it will logon within 20 seconds (although please remember the beta heads-up warning). Otherwise, it will ask for the username and password.

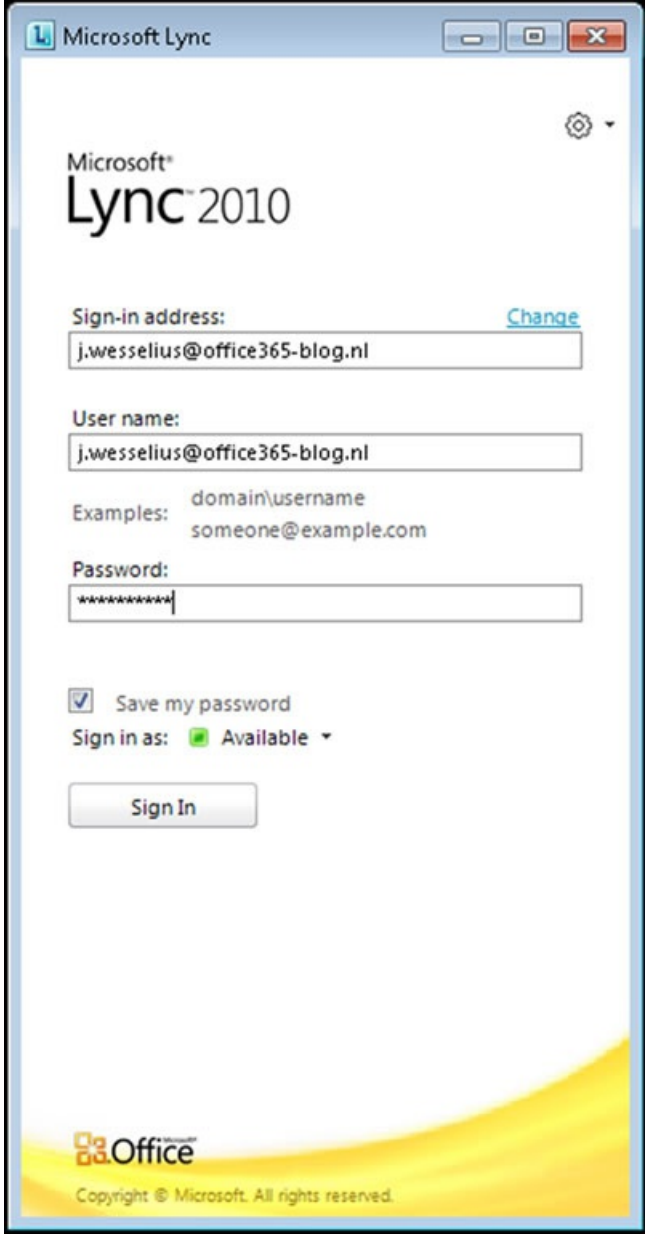


Figure 12. Enter the username and password

Once you're logged in to the client, you're ready to go; you can start adding contacts, and other people can start adding you as a new contact. This all behaves exactly as if your Lync service was being hosted locally. So, when somebody *does* add you as a contact, a pop-up invitation is shown, which you can either accept the invitation (and add this contact to a group in your Lync client), or just ignore it:

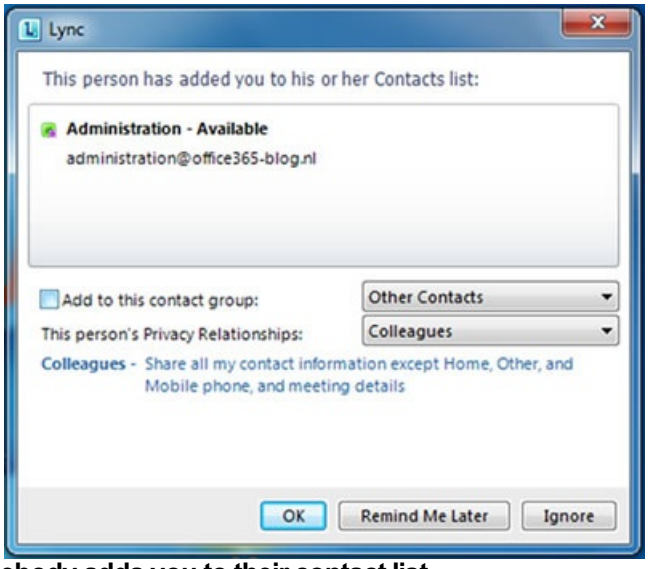


Figure 13. You are notified when somebody adds you to their contact list

One of the nice things about Lync is that is gives presence information for your contact list. By that I mean not only information about whether the user is behind their desk, but also calendar information, if published. When your contact is in a meeting, or when their out-of-office message is

enabled, those details are shown in the Lync client. To accomplish this, the Lync client is using the Exchange Web Services to communicate with the Exchange Server (in this case the Exchange Online environment) - a very cool feature!

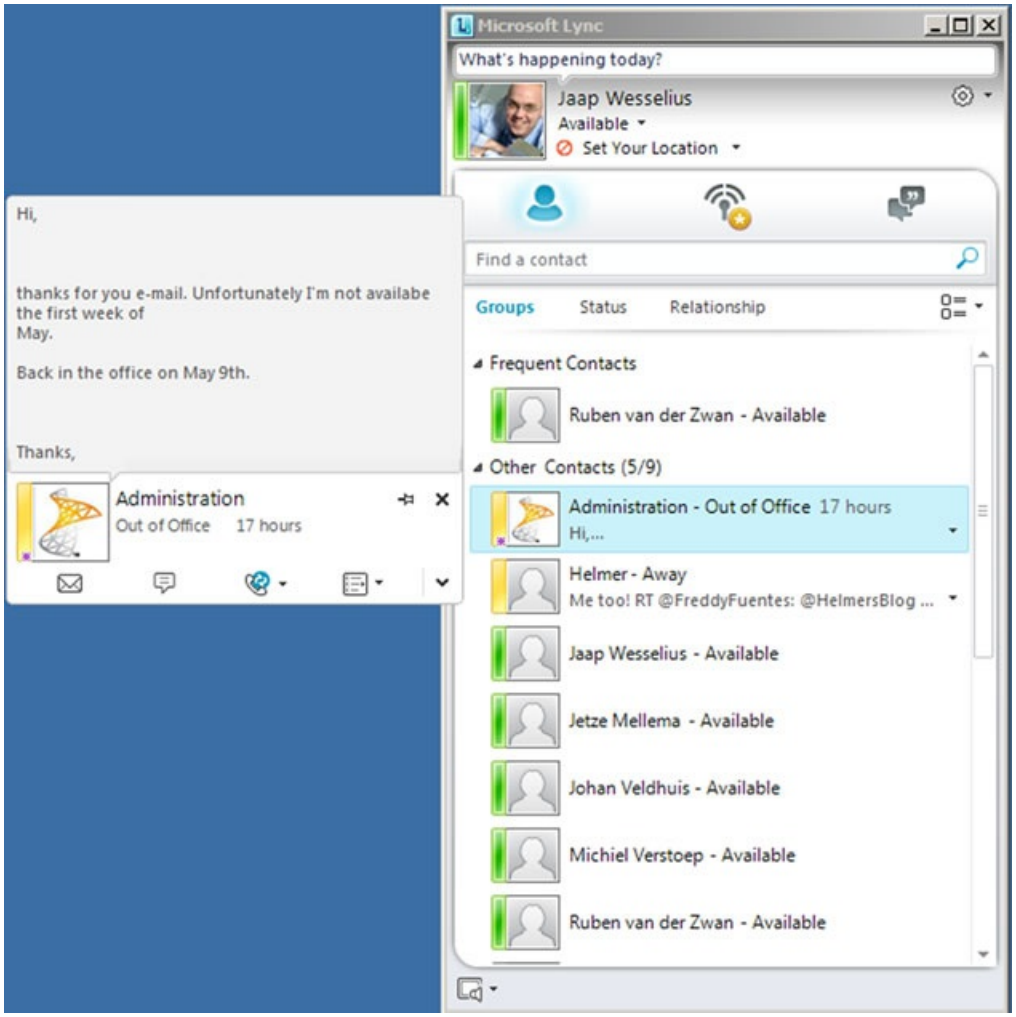


Figure 14. The Lync Client is using Exchange Web Services to retrieve status information

In addition to chat, you can also now use the audio and video features in the Lync client. Most microphone and speaker systems are supported (both built-in and plug-in options) and, when a web cam is connected to your PC, you can use the video capabilities of Lync Online and create a video conference (which is actually peer-to-peer functionality).

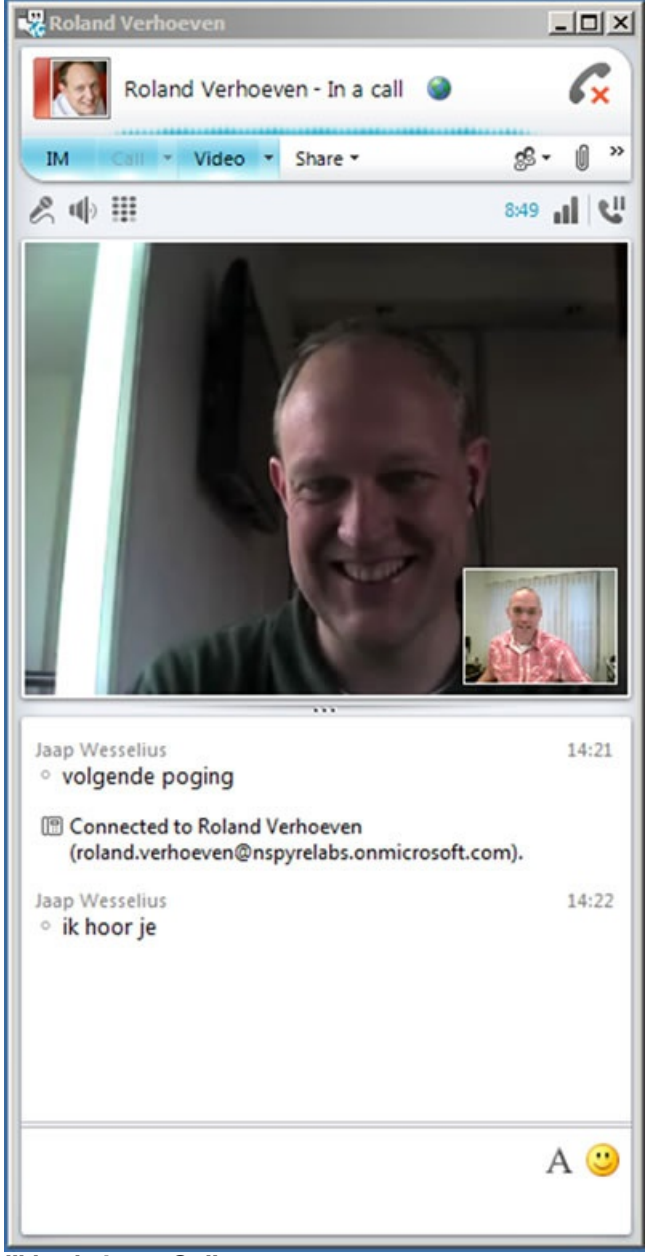


Figure 15. Video conferencing possibilities in Lync Online

At this moment it is not possible to use the enterprise voice functions which you would normally have available in an on-premise Lync Server 2010 implementation, so calls to a 'normal' telephone are not possible at this moment. Other than that, it rocks!

> The Lync team have implied that the full suite of enterprise functionality will be made available in later updates, but this isn't stated in so many words, and no time frames are given, so don't hold your breath just yet.

Conclusion

Lync Online is at the very least the Instant Messaging and presence solution in Office 365, and a full Unified Communications solution if you take full advantage of it. It integrates very nice with Exchange Online, is configured in only 15 minutes, and is really easy to use. Right now it still has some rough edges, but these will probably be smoothed off when the official release of Office 365 becomes available later this year.

The only way in which Lync Online currently falls short (relative to an on-premises environment) is that doesn't contain the enterprise voice functionality at this stage. You can make voice calls between Lync clients, but there's currently no connectivity to external phone numbers. To achieve this, an on-premise Lync implementation is needed. Other than that, Lync Online is a great solution!

Database Management for SharePoint 2010

10 May 2011
by Bert Jan van der Steeg

With each revision, SharePoint becomes more a SQL Server Database application, with everything that implies for planning and deployment. There are advantages to this: SharePoint can make use of mirroring, data-compression and remote BLOB storage. It can employ advanced tools such as data file compression, and object-level restore. DBAs can employ familiar techniques to speed SharePoint applications. Bert explains the way that SharePoint and SQL Server interact.

Introduction

If you're an IT Pro or dedicated DBA, and your company will soon be rolling out SharePoint within the organization, then this article is for you. If you're a SharePoint consultant and you need to talk to the DBAs in your company, then hopefully this will help. I'm going to walk you through a core overview of SharePoint 2010's relationship with its databases, and what you should know before you start working with either.

As you probably know, SharePoint cannot do without databases. SharePoint's relation to databases is like tracks to a train, like a reader to a smartcard… you get the idea. Specifically, SharePoint stores all of its information, both configuration data as well as content, in SQL databases. As much as this was true for Microsoft Office SharePoint Server 2007 (or MOSS, as some like to call it), it's even more the case for SharePoint Server 2010 (SP2010).

In this article, to start with, we'll touch upon how SharePoint's relationship has changed in the its most recent incarnation, and then we'll look at what's stayed the same, and what new factors you'll need to bear in mind. So, map in hand, let's dive in.

Where There were Few, Now There are Many

You see, in MOSS, the number of databases in a typical farm would range from five to seven, depending on the way the farm was setup. That number will increase dramatically when you move to SharePoint 2010. Just for reference purposes, the table below lists the databases in a typical MOSS deployment:

Database	Purpose	Remarks
SharePoint_Config	Holds farm configuration data	One database per farm
SharedServiceProvider_Config	Holds configuration data for the Shared Service Provider, such as search configuration, My Sites configuration, Excel Services	One database per Shared Service Provider (SSP)
SharedServiceProvider_Search	Holds Search-related data, such as document metadata and NTFS permissions	One database per SSP
SharedServiceProvider_Content	Holds the content of the SSP management pages	One database per SSP
MySites_Content	Holds the My Site host web application, and a site collection for each My Site	At least one per SSP
<SiteName>_Content	Holds the content for a web application and its site collections	At least one, probably more, per web application.

Table 1: Databases in Office SharePoint Server 2007

Note: Companies often store site collections in their own database, which increases the number of databases significantly.

With the introduction of Service Applications as a replacement for the Shared Service Provider in SP2010, the number of databases has increased dramatically to at least 11 for a typical farm (i.e. without any fancy stuff, such as Excel Services, PerformancePoint Services or the Office Web Applications). To give you a sense of what that means, just Search, User Profiles, and Managed Metadata are configured in the example below:

Note: In the table below, I chose the database names myself, based on my own naming convention. We'll get to naming conventions later but, for the sake of clarity, when SharePoint creates the databases through the Configuration Wizard, it will append GUIDs to the databases names, like this: **User Profile Service Application_ProfileDB_bb44087df8ed4c57a19a0d4c3305b6ff**. In my convention and the table below, the content databases use the **_Content** suffix, and the databases that are associated with a Service application use the **_SA** suffix.

Database	Purpose	Remarks
Service Application related databases		
SharePoint_Config	Holds farm configuration data	One database per farm
Search_Crawler_SA	Holds crawl information, such as URLs and paths to crawl	
Search_Admin_SA	Maintains Search Administration data	
Search_Index_SA	Holds metadata and NTFS permissions for crawled content	
ManagedMetadata_SA	Holds Managed Metadata configuration and content	
UserProfiles_SA	Holds the user profile data, which is imported from Active Directory or another source	
UserProfiles_Social_SA	Holds all social tagging information	
UserProfiles_Staging_SA	Contains data that is to be synchronized to Active Directory or another source	

Content related databases for web applications		
MetadataHub_Content	Holds a site collection for publishing site columns and content types.	Required for the Managed Metadata service, and recommended to be a dedicated web application
My_Site_Content	Holds the MySite Host site collection, and a site collection for each user's MySite.	
Portal_Content	Holds content that is associated with the Corporate Portal	

Table 2: SQL Databases in SharePoint Server 2010

Some Things Haven't Changed: Planning and Deployment

While the number of databases has changed, what *didn't* change is how you should manage those databases from initial creation to configuration and administration. The principles that were true for MOSS still apply to SP2010, and that's generally a good thing for DBAs and SharePoint folks.

Planning your deployment is ever-so-important in SharePoint 2010, especially for a controlled and error-free roll out in either production or a test environments. Having scripts available for a fully automated installation will, of course, greatly reduce the time involved for IT-Pros, especially when you have an DTAP environment. How to write these scripts and what they should include is enough material for another article, so let's not dip into that right now.

Database Servers

As you might expect, SharePoint will only (and I do mean exclusively) accept Microsoft SQL Server as its database server, keeping it all in the family. Depending on your version of SharePoint, you do have a choice as far as versions and builds are concerned, but keep in mind that SharePoint is a 64-bit application from front to back, and so SQL Server (together with your OS) must *also* be 64-bit. Although you *could* get a farm up-and-running with a 32-bit SQL Server, it's just not supported.

SharePoint Version	SQL Server Version (64-bit)
SharePoint Server 2010	SQL Server 2008 R2 SQL Server 2008 SP2 SQL Server 2005 SP3
Office SharePoint Server 2007 Office SharePoint Server 2007 SP1 Office SharePoint Server 2007 SP2	SQL Server 2000 SP4 SQL Server 2005 SP1 SQL Server 2008 SQL Server 2008 R2
SharePoint Portal Server 2003 SharePoint Portal Server 2003 SP2	SQL Server 2000 SQL Server 2005

Table 3: SharePoint and SQL versions

For specific information regarding hardware and software requirements for SQL Server, read this article on Hardware and software requirements (SharePoint Server 2010).

Creating databases

Let's start with creating the databases for SharePoint, for which there are two methods: you've guessed it - the default way, and the way that DBAs and SharePoint professionals prefer. We're going to take a quick look at these two methods, and touch upon some points that the DBAs amongst us will want to know.

The default way to create a new database is to have SharePoint do it for you automatically during the creation of a web application, via the Central Administration console. In most cases you'll be asked for a database name, and the database will be created on whichever server is designated as the default database server.

The problem with this is that SharePoint will happily ignore whatever the DBA has configured in the model database, and the databases will be created with whatever settings are configured as the server default. So, if you let SharePoint create your databases for you, be prepared to go in afterwards and modify settings.

> **In case you don't know:**
>
> SQL uses the model database as a template for new databases, and this model is configured with specific configuration settings, such as file location, growth settings and more. Whenever a database is created from within SQL Management Studio, those settings are also applied to the new database.

The preferred way for creating databases is to have your friendly local DBA pre-create them for you, so that you can just reference the database names when you configure your farm, or create service or web applications. If they want to do this using PowerShell scripts, that is of course entirely up to them, and that is also the method which I would highly recommend. Either way, the goal is to have the databases available ahead of time, complete with all the settings required by SharePoint.

> **On PowerShell scripting:**
>
> The TechNet Script Center is a good starting point for finding and creating scripts to create your databases. You might like to start with this article on how to Create a SQL Database (using VBScript). Another script, using PowerShell and DMO (SQL Database Management Objects) can be found in this blog post by Phil Factor.

Things to Bear in Mind

To be fair, with regards to database creation, SharePoint only really cares about the Collation, which should be **LATIN1_General_CI_AS_KS_WS**. Growth settings, file locations and the like are configured at your own discretion. Naturally, you should figure them out when planning your deployment, taking into account the fact that content databases will grow larger (and more quickly) than configuration databases, and that databases for Search can experience some heavy hits in large environments.

You should *also* take into account the fact that Web applications are not restricted to a single content database, which means that you can spread site collections within that web application over multiple content databases, keeping them manageable from a backup point of view.

Security

SharePoint has this concept of a **Farm Administrator account**, which is responsible for everything that goes on in the farm, and therefore needs *access to* everything in the farm; I usually have an account named **DOMAIN\SPFarm** created for this role. During setup and configuration, this user account will be used to create new databases, or attach pre-created databases to web applications and service applications.

For manageability purposes, when a new content database is created for a web application that runs under the identity of another user account (e.g. **DOMAIN\SPAppPool**), that user account will be granted the appropriate permissions by the Farm Administrator. Therefore, to make that possible, this Farm Administrator account requires **dbcreator** and **securityadmin** privileges on the SQL box.

Naming Convention

Having a naming scheme that allowed for quick identification of a databases purpose and associated application already made sense in SharePoint 2007 and, with the vast increase in the number of databases in 2010, that argument now makes even more sense.

One rather annoying, albeit very useful habit of SharePoint is the use of GUIDS, which are all over the place. You'll find them in logs, for instance, and 'under the hood', where they're called **CorrelationID**s. This latter implementation actually make sense, because a GUID allows you to track things rather well, and the CorrelationID is associated with a particular event, which can then be very easily traced in the logs.

However, when GUIDs start appearing in database names, that's another story, and that's also when it becomes frustrating. Unfortunately, in the situations where SharePoint creates databases for you (that is, without you specifying the exact database name), it will do exactly that.

The only way to avoid this is to use PowerShell for creating the different components that have associated databases, and provide the exact names you want to use, according to your preferred naming convention. I like to use prefix of **SP2010_**, followed by the name of the service or web application, and whether it is a content or service oriented database (as seen in Table 2). It should therefore be obvious what these two databases represent:

SP2010_UserProfileTagging_Service	The social tagging information of the UserProfile Service Application
SP2010_Portal_Content	The content that is in the Portal Web Application

Failover

SharePoint 2010 has the ability to take advantage of SQL's database mirroring feature, which allows for automatic failover to another database server instance if the currently-active one should fail. For practically all databases, you can enter the name of a failover instance which SharePoint will revert to after a small timeout.

As you might expect, this requires teamwork and cooperation between SQL and SharePoint folks, so be prepared to discuss this between the teams.

Remote Blob Storage

Imagine a company that deals with large files, and I mean really, really, large files. Think images, video files, sound clips, etc. Files that require the information management capabilities of SharePoint in terms of versioning, workflows, or perhaps just search. These files could potentially take up so much space in the database that it would significantly (and negatively) influence backup schemes, and possibly impact on system performance.

To avoid these problems, SQL Server 2008 provides Remote Blob Storage or RBS, which moves (streams) Binary Large Objects (BLOB) from the database to the file system. By doing so, it decreases database size while still maintaining control over the files as if they were in the database, which means that SharePoint won't see the difference when accessing these files.

My recommendation is that you only use RBS if you'll need it, not just because you can. Make sure there's a business case for it, because it requires specific expertise on the part of your DBAs.

Read more about Remote Blob Storage here in this overview of Remote BLOB Storage (SharePoint Server 2010)

Database Maintenance

In addition to the fact that SharePoint will generate a lot of databases, the size of all these new databases can grow at really high speed, depending on how popular your deployment is. Trying to regain some of the disk space that a database uses might be tempting, but be careful. As with all of

your high-use systems, plan database shrink activities during normal maintenance windows, rather than implementing auto shrink through maintenance plans on a database.

Shrinking a database is possibly the most resource-intensive operation that SQL can perform, and you certainly don't want to do that during office hours. Plus, it would only be beneficial to you when your database has lost a lot of its content, e.g. after moving a site collection.

There is plenty more information on database maintenance available in this Whitepaper: Database Maintenance for Microsoft SharePoint Products and Technologies.

Summary

If you didn't know already, then you'll certainly have gather from this article that SharePoint relies heavily on databases. Knowing the roles of those databases and how to manage them can have a dramatically positive impact on the performance of your SharePoint deployment, and hopefully this article was helpful in that regard. If you're already familiar with SharePoint 2007, then the basic concepts behind the management of the relevant databases will also be familiar territory, albeit with a few new considerations. Either way, the pointers we've covered here should give you the edge you need to make sure you manage the databases, not the other way around.

If you want to learn more about managing the database elements of SharePoint 2010, here are the links mentioned in this article for further reading:

- Hardware and software requirements (SharePoint Server 2010)
- Determine hardware and software requirements (Office SharePoint Server)
- Whitepaper: Database Maintenance for Microsoft SharePoint Products and Technologies
- Overview of Remote BLOB Storage (SharePoint Server 2010)

SmartAssembly Sync for JIRA

Published Thursday, May 12, 2011 1:59 PM

JIRA is a powerful bug *tracking* system, but it can't help you to actually collect error reports from your users. For that we've started using SmartAssembly (SA), but we quickly realised that there was no easy (or, more importantly, automatic) way to get error reports *out* of SA and into JIRA.

Automation is the Mother of Invention

David Simner, Ben Challenor and I got together in one of our regular Down-Tools weeks to see if we could do something to ease the pain of importing error reports into JIRA. SA had an SDK, and JIRA had a SOAP API, so we automated!

Given that we were all already users of both systems, we knew how we liked to handle bugs and error reports, and wanted to make sure we supported those processes. So, after 4 days of intensive coding and testing, SmartAssembly Sync for JIRA was born. We started using it in our own teams, and before long the rest of Red Gate were clamouring to have it set it for them as well.

The other teams really appreciated the way that duplicate bug reports are grouped together (letting them see how many people had hit specific issues). Combined with all the other information that our tool provided, and JIRA's powerful searching, prioritisation was a piece of cake.

What's going on under the hood?

What the tool actually does is fairly obvious if you know how JIRA and SA work - we just put all the pieces together. Without going into too much gory detail, here's the whole process that SmartAssembly Sync for JIRA runs through every 10 minutes on a scheduled task:

First, it makes SA download any new reports. Then it connects to your JIRA server, and downloads its configuration (which is stored as special "SA Project Link" issues in JIRA so anyone can configure it easily).

For each report it downloaded, it does the following:

a. It looks to see if the bug is already in JIRA.

b. If it *isn't:*

> *i. It opens a newJIRA issue;*
>
> *ii. If the report is from some library code that is used by more than one product, it opens another JIRA issue in the JIRA project for the appropriate library, and links the two.*

c. If the bug is already in JIRA:

> *i. It comments on the existing JIRA issue*
>
> *ii. If the JIRA issue is marked as resolved or closed, and the report is from a later version, the issue is re-opened;*
>
> *iii. It updates several fields in JIRA as necessary, such as the earliest version seen, latest version seen, report count, etc.;*
>
> *iv. If all your systems are set up appropriately, it can send an auto-email to the customer who sent the report (assuming they gave an email address).*

It then has some pretty excessive extra features. For example you can configure it so that if a report is from some shared library code, it'll open a second JIRA issue in the JIRA project for the library, and link the two.

Once you've got it working, the tool automatically categorizes reports that are probably the same bug (by the type and location of the exception) providing everything that's specific to the bug (i.e. stack trace, exception type) in the description. Then each report becomes a comment on the bug, including all the per-report information (i.e. build number, email address, etc.).

All in all, I think the tool is cool, and I know Ben and David do too. It's also some of the nicest code I've been involved in writing, which is lucky, because we've decided to open-source it :)

Making the most of your error reports

Anyone who uses SmartAssembly error reporting will benefit from this tool, as it will completely remove the need to manually import issues into JIRA, which is potentially a huge time-sink. Quite apart from the time-sink, most developers will probably find that tracking bugs in two different places also makes it easy to forget about the SmartAssembly reports, which is going to mean their software ends up lower quality, which is bad for everyone.

It'd be great if you could take SmartAssembly Sync for a spin, and let us know what you think :)

If your bug-tracking system isn't JIRA, SmartAssembly Sync is written very modularly, and it should be very easy to modify to work with any other bug-tracking system with a decent API. I'd really appreciate hearing from you if you try this, especially if you're willing to share your code!

You can find out more on the JIRA blog.

by Alex.Davies

Migrating from OCS 2007 R2 to Lync: Part I

09 May 2011
by Johan Veldhuis

> Lync Server is more than just a name-change for OCS, it is a significant upgrade. How then do you make that migration from and existing Microsoft Office Communications Server (OCS) 2007 environment to Lync? Johan explains, from experience, all the necessary steps to the point that one can confidently say 'Gentlemen, start your engines!'

In my previous article, we had a look at a few of the new features of Microsoft Lync Server 2010. I also mentioned the installation process, and we will now, in this article, have a look at how to migrate from Microsoft Office Communications Server (OCS) 2007 R2 to Microsoft Lync Server 2010. Let's first have a look at which of the possible migrations are actually supported by Microsoft.

Supported Migrations

As with all its other products, Microsoft provides support for migrations from previous versions. You can, it turns out, migrate both OCS 2007 and OCS 2007 R2 environments to Lync Server 2010. Earlier versions of the product, such as Live Communications Server (LCS) 2005 are not supported.

Current Environment

First, let's describe our typical current environment.

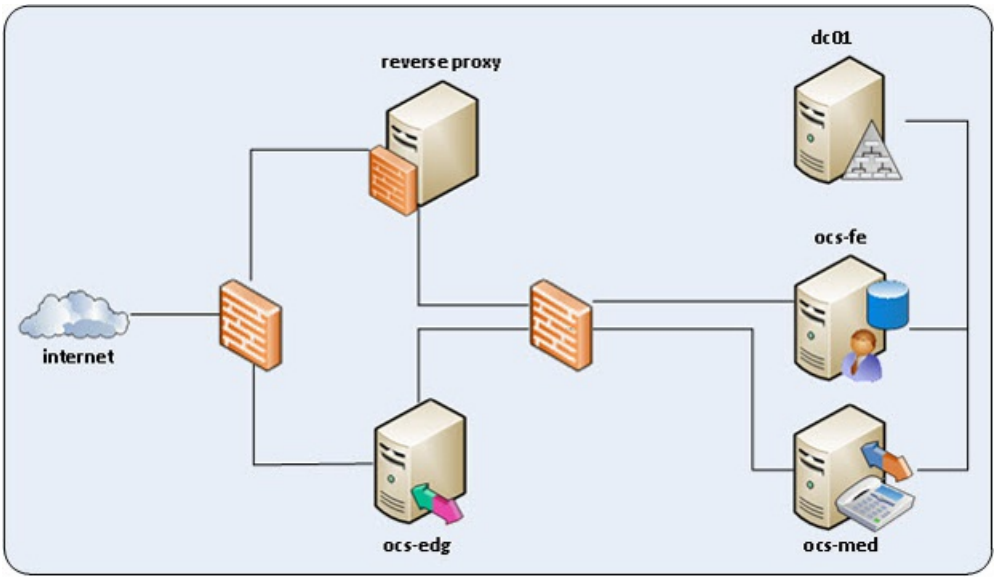


Figure 1. Overview of our current environment

In the diagram above you will find the servers currently implemented:

- DC01, our domain controller which also contains the certificate authority role
- OCS-FE, our OCS 2007 R2 Front-end Server
- OCS-MED, our OCS 2007 R2 Mediation Server
- OCS-EDG, our OCS 2007 R2 Edge Server

All servers are running Windows 2008 R2 and are located in our domain called CORP. Both the forest and domain functional levels are set to Windows 2008 R2.

The users in the domain are all enterprise voice-enabled and are allowed to connect from the internet. As well as this, federation with external companies and the Microsoft Live network is allowed.

In addition, both dial-in conferencing and two response groups are active.

The company is using a SIP trunk from a provider which is identified on both the OCS and Lync interoperability sites of Microsoft.

Requirements

Let's now list all the requirements which must be met.

- During the implementation of Lync Server 2010, the current environment must remain available for users;

- The contact lists for all users must be migrated to Lync;
- All other functionality such as federation, enterprise voice, dial-in conferencing and response groups must be migrated to Lync;

Now we know how our current environments looks like, and which requirements must be met, we are ready to start installing Lync.

Preparations

Before starting the complete process, we must create a backup of the following components:

- Active Directory
- OCS 2007 R2 Front-end Server
- OCS 2007 R2 Edge Server

Because all our users have client applications based on OCS 2007 R2, we will need to perform some preparations on that side also.

The best practice is to upgrade the client side with the most recent cumulative update (CU) which can be found on this site. Beside the normal client apps, don't forget the attendant client and phone clients.

Installing Lync Server 2010

I mentioned in my previous article that there are two methods of starting the implementation of Lync:

- Using the planning tool
- Using the setup

We will use the setup in this article, but before doing this we will need to install the prerequisites.

Install prerequisites

Because Windows 2008 R2 already contains .NET 3.5 SP1, you won't need to download it but just install it. .NET and all other prerequisites can be installed using the following PowerShell command lines:

```
Import-Module ServerManager

Add-WindowsFeature  AS-NET-Framework  ,Web-Static-Content,Web-Default-Doc,Web-Http-Errors,Web-Asp-
Net,Web-Net-Ext,Web-ISAPI-Ext,Web-ISAPI-Filter,Web-Http-Logging,Web-Log-Libraries,Web-Http-
Tracing,Web-Windows-Auth,Web-Filtering,Web-Stat-Compression,Web-Mgmt-Console,Web-Scripting-
Tools,Web-Client-Auth
```

Optionally, you can install RSAT-ADDS which makes it possible to prepare the Active Directory using the Front End Server.

```
%systemroot%\system32\dism.exe                    /online                    /add-package
/packagepath:%windir%\servicing\Packages\Microsoft-Windows-Media-Format-
Package~31bf3856ad364e35~amd64~~6.1.7600.16385.mum /ignorecheck
```

As a final step before starting the setup, you should create a share. This share will be used by Lync to store the application server files, central management files and files related to web services. These include the files which must be replicated from the Central Management Store and vice-versa; custom prompts for response groups, meeting content and the address books.

Ensure that at least the user which will be used to create a configuration using the Topology Builder has full control on the share. As well as this, the user needs to be a member of the domain users group and the local administrator group. If the permissions are not set correctly, then the process of publishing the new configuration will fail.

Once these steps have been completed then it's time to launch the setup.

When you start the setup process for the first time, it will check whether Visual C++ 2008 is installed. If not, then it will prompt you to install it. If it does so, then press yes to install the required software.

Figure 2. Install Visual C++ 2008 Redistributable

Once the software is installed, you will be prompted to select a location for the installation files and the future Lync Server components. When the installation files have been copied, you will see an interface which is reminiscent of OCS 2007 R2. The next step we need to perform is to prepare the Active Directory.

Prepare Active Directory

Select the option **Prepare Active Directory for Lync Server:** This will bring you to a new page. This page contains all steps which must be completed in order to prepare Active Directory for Lync Server 2010.

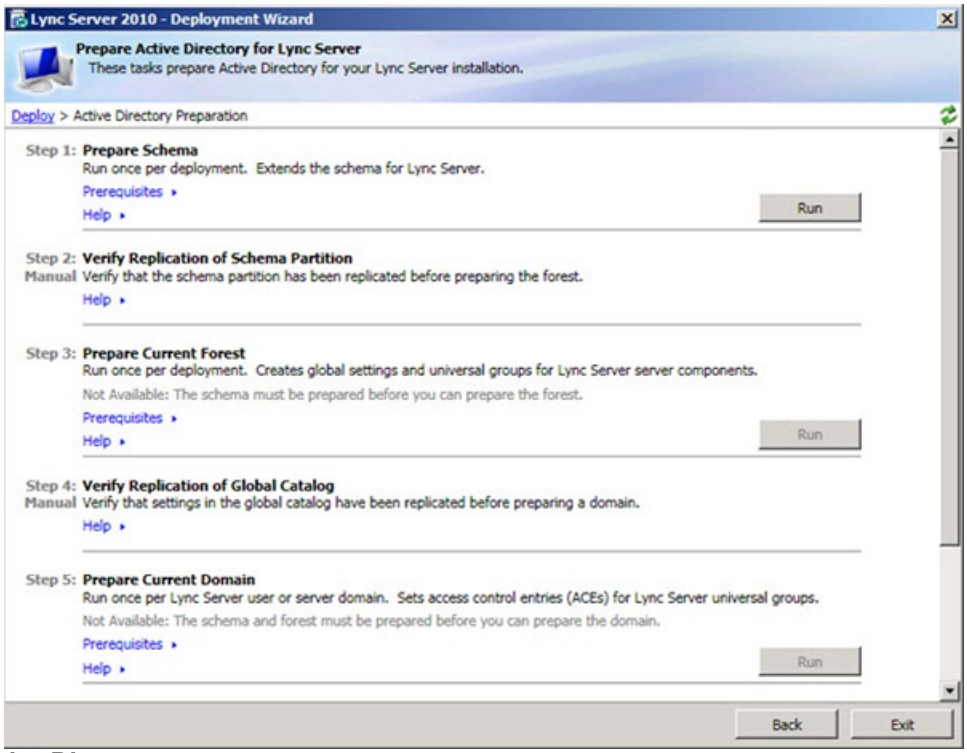


Figure 3. Prepare the Active Directory

Before running these steps, you will need to make sure you have a correct backup of your Active Directory in case something goes wrong. When this has been confirmed, we can start with the first step: **Prepare Schema**. This will extend the current Active Directory schema with those objects that are necessary for Microsoft Lync Server 2010.

Since we don't have multiple domain controllers, we don't have to check whether replication has occurred after the schema is prepared. So we can continue with **Prepare Current Forest**. During this process, several universal groups will be created in your Active Directory. This step is completed pretty fast compared to the schema preparation.

As a final step, we need to prepare the domain. This can be done by selecting the **Prepare Current Domain** option. During this process, several ACLs are configured on the necessary Active Directory objects. When this step has been completed we can continue with the next step: **Prepare first Standard Edition Server**.

Prepare first Standard Edition Server

Before you can publish a topology to the Central Management Store (CMS), you will first need to have one. To create the CMS, you will have to select the option Prepare first Standard Edition server. During this process, the SQL connectivity tools will be installed along with an SQL 2008 Express Edition instance. Additionally, the UCMA Redistributable will be installed. As final step of the preparation, a firewall exception is added to allow connections to the SQL server instance.

Figure 4. Prepare first Standard Edition server

Now that our environment has been completely prepared, it's time to define our new Lync Topology. To do this we will need to install the Topology Builder.

Install and use the Topology Builder

When the **Install Topology Builder** option is selected, a wizard will be started which guides you through the installation process. Follow the steps in the installation wizard in order to complete the installation.

When the Topology Builder has been installed, we can define our new Lync Topology.

To start designing your topology, choose the **New Topology** option, which will then prompt you to choose a location to store the topology file.

The first step in the wizard will ask for the SIP domain; this must be exactly the same domain as currently configured in our OCS 2007 R2 deployment. If these two are not the same, then the merging of the topologies will fail.

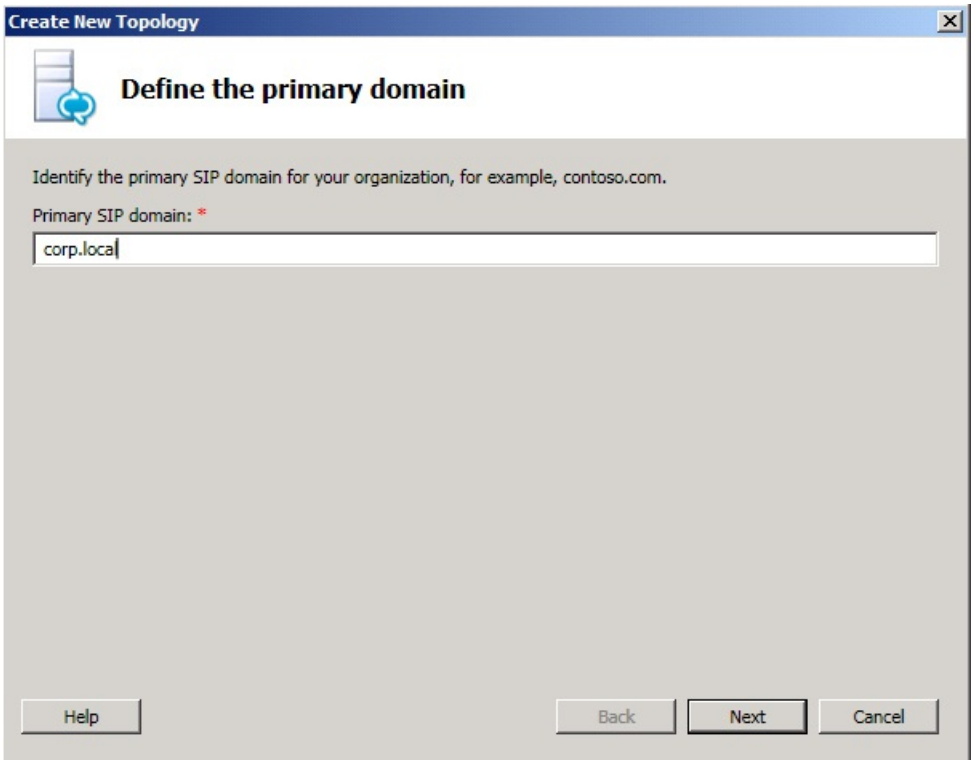


Figure 5. Define the primary domain

Once the SIP domain and optional other SIP domains have been specified, you will need to create a site. Each deployment can contain multiple sites, but requires at least one site. Additional sites can be added later on. When the site, and additional information about the site, have been provided you will then have the option to launch the next wizard.

This wizard will guide you through the process of configuring the Front End Server. Using the first step you will have to specify the fully qualified domain name (FQDN) of the Front End Server or Front End Pool. This last one is only applicable when deploying an Enterprise Edition Front End Server. Because high-availability was not a requirement, we will create a Standard Edition Front End Server. So we will choose the option **Standard Edition Server** and specify the FQDN of our Front End Server. This is the local FQDN of the server which is `lync-fe.corp.local` in our case.

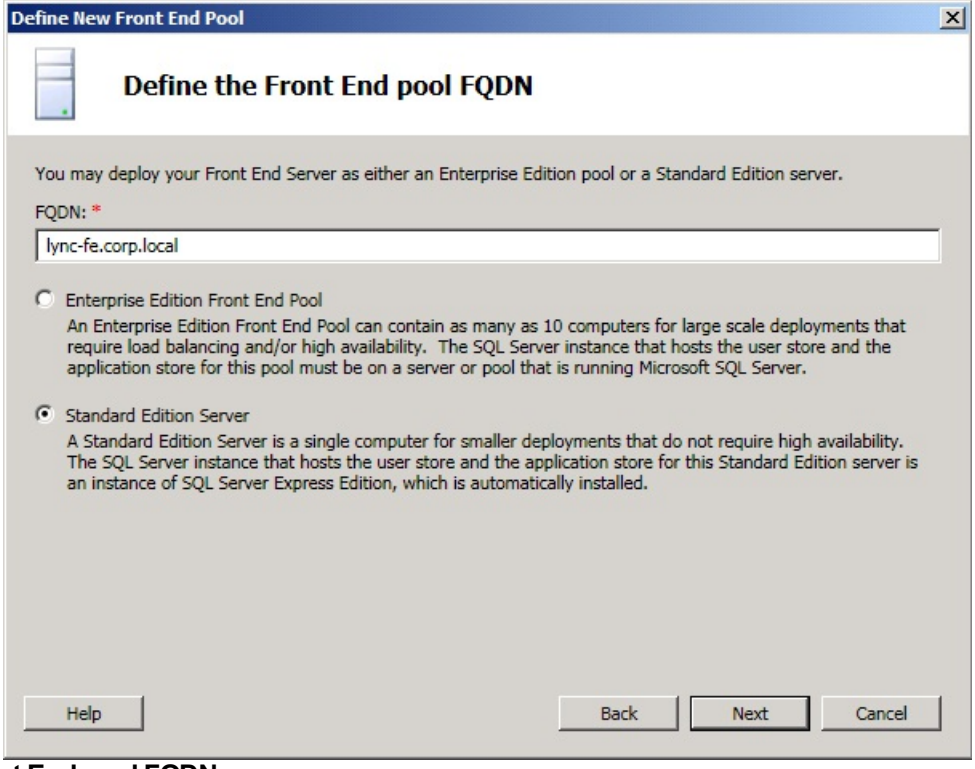


Figure 6. Define the Front End pool FQDN

Next step is to specify which features will be enabled on the Front End Server. The features which you enable must be exactly the same as the features enabled in OCS 2007 R2. This due to the fact that if you have not selected them, you will be able to migrate the feature but it won't work. In this case we will only enable the **Conferencing, Dial-in** and **Enterprise Voice** options.

Once the features have been selected, it's time to enable the roles which need to be collocated on the Front End Server. Microsoft highly recommends that one should co-locate the Mediation Server during a migration. After the migration process you can decide to build a dedicated Mediation Server and remove the co-located one.

The next step is to associate additional servers such as Edge, Monitoring or Archiving Servers to the pool. Don't select the **Enable an Edge pool to be used by the media component of this Front End,** this will be enabled later in this article. Because the Monitoring and Archiving Servers are not a requirement we don't select them either.

Since we are installing a Standard Edition Front End Server, the next step can be skipped since you can't store the CMS on a separate SQL Server.

You'll remember the share which we described earlier on, which was created as prerequisite. In the next step we will need to specify the name of the share which was created, and the server on which it is located. Once this is done we will arrive at one of the final steps, **provide the external base URL**. This URL will be used for accessing the web services offered by the Front End Server. For easier transitioning later on, you can keep the same URL as used for OCS or decide to create a new one. Keep in mind that the last method does require some DNS changes and more reverse proxy changes then the first method.

As a last step you can configure a gateway which is used by the Mediation server. This gateway is used for outgoing calls to the PSTN. Don't create one yet, but press **finish** instead.

Once you are satisfied with the topology, you can publish the topology by selecting the **Publish Topology** option in the actions menu.

Follow the steps in the wizard to publish your new topology. When you perform this action the first time, the CMS database will be created. Once it is created, the topology will be published and enabled.

Install Lync Server Components

When the topology has been published we can install the required Lync Server Components. This can be done by selecting the **Install or Update Lync Server Components** which will install the required software components for each role. Each server will determine which components will be installed by looking in the CMS. If you subsequently wish to make major changes by using the Topology Builder, then rerun this option to add or remove the components.

Before the components can be installed, we first need to install a local replica of the CMS. This can be done by selecting the option **Install Local Configuration Store**. This will install an additional SQL Express 2008 instance on the server. You will be prompted to select a source, in this case select the **Retrieve Directly from Central Management Store** option which will retrieve the data directly from the CMS.

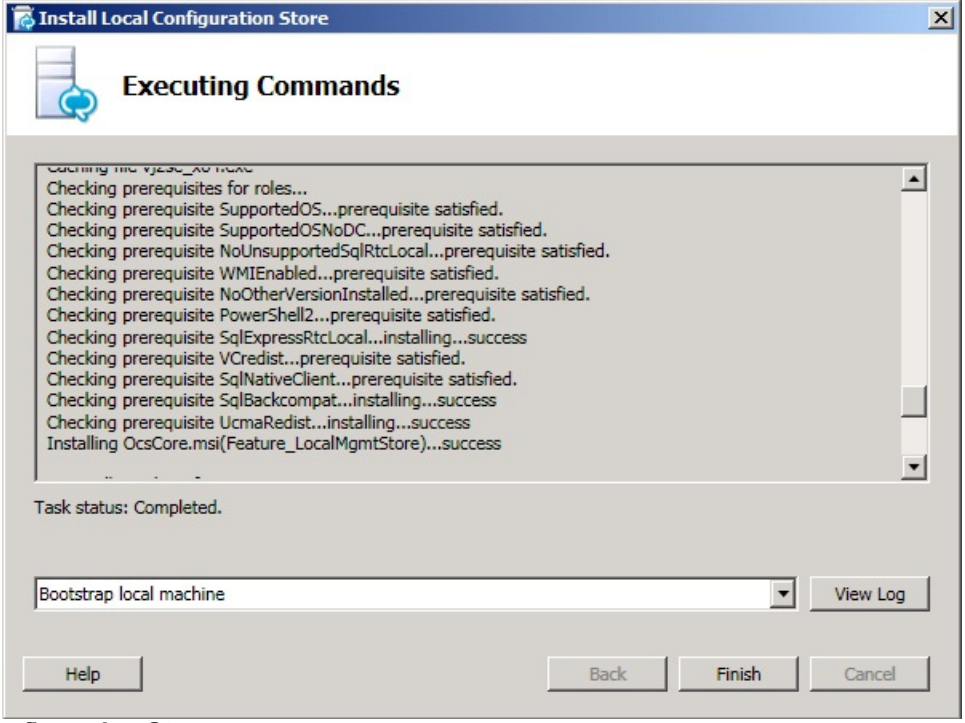


Figure 7. Install Local Configuration Store

Once the installation of the Local Configuration Store is completed, you can then install the server components. This can be done by selecting **Setup or Remove Lync Server Components**.

The setup will use the data from the CMS to decide which components to install, so you don't have to select the components yourself. Once all the components are installed we've reached the next step: **request and install the certificates**.

This process can be started by selecting the option **Request, Install or Assign Certificates**. First we need to request a new certificate, so press the **request** button to start the wizard.

The first question you need to answer is whether you are creating an offline request or wish to submit the request directly. In most cases this will be the latter, since an internal certificate authority will be used. In our situation this is also the case, so we select **Send the request immediately to an online certificate authority**.

In the next step, make sure that you select the correct certificate authority and continue with the next step. The next two steps are optional and are not always required. These steps will let you:

- Specify an alternate credentials to request the certificate
- Specify an alternate certificate template

Now the two steps have been completed, it's time to specify the **friendly name** which is used to easily identity the certificate. Besides this, it will give you the opportunity to make it possible to export the certificate including a private key.

The next two steps are both company-related settings and thus we will skip them in this article. Now we reached the most important steps: **specify the names which need to be located on the certificate**.

In most cases, all the correct names are pre-populated automatically using the data retrieved from the CMS, but be sure to check nonetheless whether all names are there, and add additional names later using the wizard.

After this step, you will need to select the SIP domain which must be listed on the certificate. Selecting this option will add a SAN entry sip.domain.com to the certificate, so in our case `sip.corp.local`.

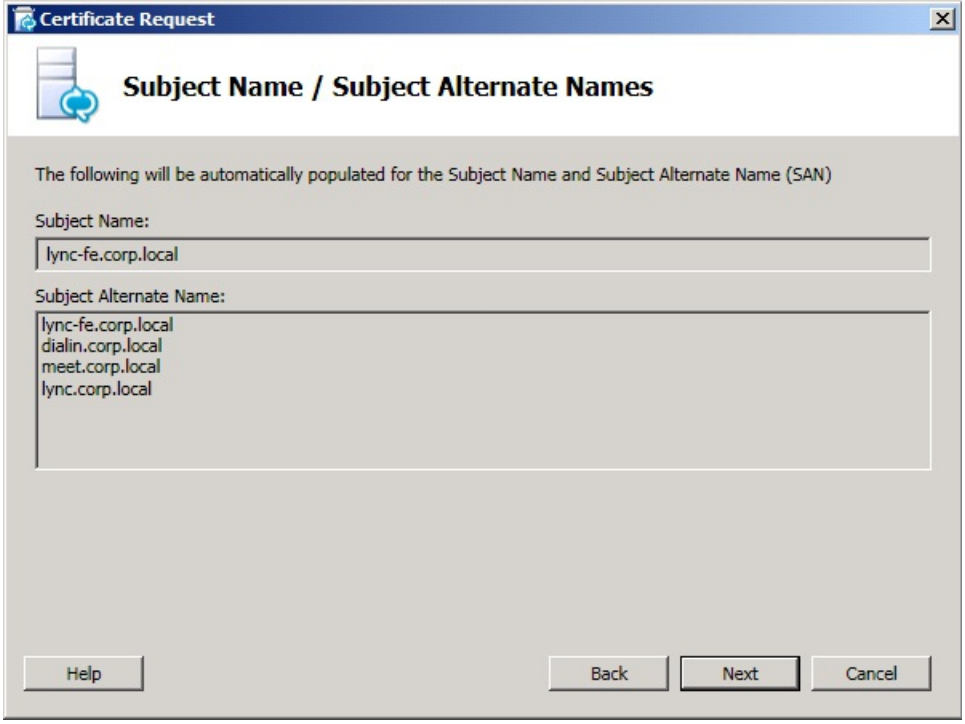


Figure 8. Provide Subject Name/Subject Alternate Names

Besides this, all the short URLs that are configured by default are automatically added as Subject Alternate Names (SAN).

As I mentioned earlier, it is possible to add additional SAN entries and this can be done in the next step. Normally you won't have to add additional names but if you need to, you can do it in a few moments.

Before the actual request is submitted, you will receive a summary. Be sure to check all settings before submitting the request. Once the request has been submitted and the certificate has been installed, a new wizard will be automatically started.

This wizard will guide you through the process of assigning the certificate to the server. Just select the correct certificate and follow the steps of the wizard to assign the certificate.

Once all these steps have been completed, it's time for the next step: **Gentlemen start your engines!**

Start the Lync Services

Now all components have been installed and the certificate is installed it's time to start the services. This can be done by selecting the **Start Services** option from the setup. Follow the steps from the wizard and wait till the services are started.

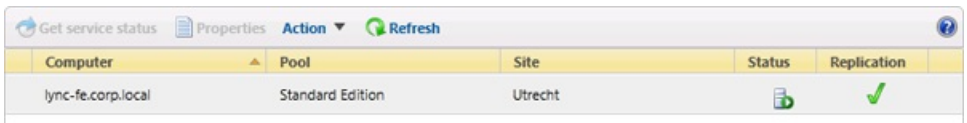
If you want to make sure the services are really started you can use the **Service Status (optional**) option to check if all services are started.

Check Both Environments

Now the Lync deployment has been completed for now, it's time to perform some checks before continuing.

The check starts with verifying that both the OCS and Lync services are running. This can be done by opening the services MMC or by opening the management consoles from both products.

Before you can use the Lync Control Panel don't forget to add the user account to the **Cs-Administrators** group.



As well as doing this, confirm that the users who are hosted on OCS can still use all the functionality of OCS.

This brings us to the end of the first part of this article about migrating from OCS 2007 R2 to Lync Server 2010. In the next part we will start with merging the topologies and connecting Lync to the internet using the OCS 2007 R2 Edge Server.

When this is finished we can perform a test migration and check if all functionality works correctly before migrating the other resources.

Benjamin Pollack: Geek of the Week

06 May 2011
by Richard Morris

Benjamin Pollack is well known for his work on Fog Creek Copilot, and Kiln. He is famous amongst young geeks for his role in a documentary film and website 'Aardvark'd: 12 Weeks with Geeks', which plotted his internship with Fog Creek back in 2005.

It's easy to get a sense of what the software firm Fog Creek's office is like by clicking through their website. The pictures bear witness to a pleasantly light-filled office with great views overlooking Broadway, the wide avenue in New York City that runs the full length of Manhattan.

Fog Creek, founded by Joel Spolsky and Michael Pryor to combat that strange world inhabited by 'Napoleon-complex junior managers', is the home of some great programmers.

To many of us, the Fog Creek office is most familiar as the home of Project Aardvark, made into a cult film 'Aardvark'd: 12 Weeks with Geeks'. The documentary, filmed in 2005, showed how four interns developed Fog Creek Copilot, a remote assistance software tool, having been given 12 weeks to design, develop, debug and ship it. One of those geeks was Benjamin Pollack, who then went on to help make the original prototype of what became Kiln, a version control system (DVCS) which effectively means fast access and no need to wait to check in code.

In his free time Benjamin works on his screenplay and his novel, plays Chopin and Rachmaninoff on his piano, and occasionally spends time coding for fun - usually in Pharo, Smalltalk or Python

RM: Benjamin, what's your background in technology? How did you learn to program?

BP: I started to program when I was really young. My dad had worked at SRI on robots like Shaky (the first general-purpose mobile robot to be able to reason about its own actions) back in the 1970s, so we had a computer in the house as far back as I can remember. One day, my dad came home with a new computer, a 286, which came with a copy of GW-BASIC, and there was a game called DONKEY.BAS, which was written in it. My dad showed me the general idea, and gave me the GW-BASIC manual. So I taught myself some GW-BASIC. The stuff I wrote was really silly, but it was enough to pique my interest and get me started.

Things kind of grew from there. I somehow got my hands on a copy of Turbo Pascal later on, and was impressed by how much easier programming got with functions and structures - bleeding-edge even now, I know. Being eight is fun, because everything is totally new.

In middle school, I started to realize that I really liked programming, and by early high school, I had been really fortunate to discover Squeak Smalltalk. Smalltalk was the first language and environment I actually understood 'all the way down': the fact that Squeak in particular was written in itself all the way down to the VM meant that I could actually do things like debug into the compiler, figure out how my text was turned into a syntax tree, figure out how that was turned into byte codes, then debug those right into the VM to figure out how those ultimately got executed on the hardware. I was blown away by how incredibly powerful this environment was compared to anything that I'd used before.

That's when I really got hooked: it was so easy to make amazing stuff using Squeak that I just got addicted. I wrote something, even if it was dinky, nearly every day. Eventually, I got to college, realized that I really wanted to make a career out of this, and did. I'd say it's worked out well.

RM: Are there languages that you use all the time and ones that you stay well clear of?

BP: The languages I use all the time are C# and Python.

Python's simply wonderful because it's not only easy to read and write; it's always really easy to figure out exactly what the language is going to be doing behind the scenes with any given piece of code. Combine that with a really active community and great tools, and it's a hard language to beat.

C# gets flack for being a Microsoft product, but to be honest, I think it's one of my favourite languages these days. People are always saying it's just Microsoft's Java, but I just can't get that. It has lambda expressions. It has a sane event model. It's got great asynchronous support that's only going to get better with C# 5. It has type inference, real generics, LINQ, partials, just tons and tons of stuff that makes a tremendous difference when you're writing real code. I honestly can't stand using Java, but I love coding in C#.

I try really hard to avoid what I'd call ambiguous languages. What I mean is languages where it's very hard to look at any given line of code and know what actually gets run. C++ is a great example: unless you take great pains to avoid operator overloading, mark your

constructors as explicit, and so on, it's extremely difficult for me to look at a line of C++ and have the faintest idea what's actually going on.

I don't even really mean this in terms of the machine code generated; I mean in terms of the actual semantics. I mean, take how iterators work. The right way to do iterators in C++ is to override the pointer dereferencing operator on your iterator object. But pointers are fundamental to the language!

So now you're saying that I can have this pointer-like thing that isn't necessarily a pointer, but might be, so maybe I can put it in a collection, maybe not, maybe it copies if I hand it off somewhere else, maybe it doesn't. Who knows? Throw in reference parameters, and you have a variable whose pointer you might be able to get in the caller, but not the callee, even though the usage semantics are identical.

It's not you can't write good code in languages like C++; that's patently false. I just believe my time is much better spent solving problems than second-guessing the language's syntax.

RM: Do you remember the first interesting program that you wrote?

BP: That depends on what you count as interesting. The first program I wrote that I thought was really cool was that I played 20 questions on one of the Apple IIs at school, and wondered how it recorded what your answers were so that it could learn. So I wrote my own version. That forced me to learn about data structures in a way that I hadn't really understood before. Before that point, I think I'd thought that all the data had to be hard-coded in the program; making something that could learn meant I had to learn about arrays and saving to disk.

That's totally uninteresting in a grand sense, but I learned so much writing it that it stuck with me all these years, so it's certainly interesting to me.

RM: What are the things that if someone had sat you down at the very beginning of your career and said, 'You need to know X, Y and Z' that your life would have been much easier?

BP: There's really only one thing that I wish someone had told me: I wish I'd had someone point out to me is how important it is that you know how older technology works, because it never goes away.

In college, they like to focus on new, up-to-date stuff, so we were writing in Java and template-heavy C++ for group projects, and I used Smalltalk or Ruby everywhere I could for individual projects.

But in the real world, there's lots of old code that you need to deal with, so while it's great that you know how to write GUIs in Swing or play with object databases like Gemstone or use an SCM like Subversion, what you actually ended up working with back then was probably use VBScript to talk to Access and store your code in SourceSafe. Or if you were really lucky, you got to write Win32 GUIs with MFC and maybe use CVS.

Anyone will figure this out in three seconds on the job, but I think a little warning would have been nice to lessen the shock.

RM: Let me ask you about your time at Fog Creek. How did Kiln start and was there a specific problem you were trying to solve? Did you write mocks in the test-first sense so you could test it as you went along?

BP: Kiln's history's actually really interesting to me. The way it began was that a friend and colleague of mine, Tyler, really wanted to do this 48-hour coding competition called Django Dash. At the time, he and I both wanted to start doing code reviews more rigorously, but we hated all of the tools that we had available to us.

They all revolved around the workflow of, 'take a patch, upload it onto a server, and then comment on it.'

This meant that the reviews were totally outside source control and weren't versioned in any sense, so even the ones that had some decent discussion and some concept of history, you'd lose that discussion and history as soon as you checked in your code.

About the same time, I'd helped Fog Creek move from Subversion to a relatively early version of Mercurial, and we were still trying to figure out exactly how to make good workflows with this new tool. So Tyler and I decided that maybe our code review tool should simply be based on Mercurial.

So we did a mad dash in the code sprint to make a code review tool based on Mercurial. We ended up with a decent tool, and we won the competition, and that was the end of it for about six months. Then Tyler and I were talking one day, and we realized we really wanted to do something other than Copilot, so we decided to try to pitch bringing Kiln in-house. I used my Thanksgiving vacation to refactor the entire code base, named the product Kiln, and gave it the first stylesheet that I didn't think deliberately looked horrible. Tyler and I then bounced that around for a month, demoed to the company in January 2009, and convinced them to make a real commercial product out of it.

We did not do TDD, nor do we now. Tyler wanted to, but we were changing how the app worked so much so frequently that we'd have spent the entire sprint writing tests. We had similar issues when we began working on the "real" version. I don't think we'll ever use TDD at this point.

RM: What form did the design take? Pseudocode? Actual code? Whiteboard scribbles?

BP: For the prototype, it was all actual code. We had no idea how the workflow should really work; just some loose ideas. So we talked for a few minutes about something we thought might work, then did an insane code sprint to just get something working, and then

immediately began using it so we could try to figure out how the tool should actually work. We ended up surprisingly close to the mark, but being able to use the tool highlighted a bunch of problems we didn't think of that we fixed before the contest ended. So basically: come up with a rough idea, code it up, try it, iterate.

When we brought Kiln into the company, things changed. We spec'd out Kiln's high-level architecture on whiteboards, while our designers worked with us to develop relatively complete paper specs with images and descriptions of various workflows. We'd take those specs around to people in the company and see how they tried to use the paper interfaces. Once we had a pretty good idea that what we were doing didn't completely stink, we'd code it up, distribute it to everyone, and promptly realize that the interface didn't work at all. So then we'd do what we did in the prototype: do a bunch of revisions on the code based on feedback and whiteboard discussions until we had a usable interface.

Unsurprisingly, we now do only whiteboard specs, followed by working prototypes as quickly as possible. Having the iteration loop is much more helpful than trying to plan up front.

RM: You say on your blog that Kiln has a mission, not a mission statement. What do you mean by that?

BP: I didn't say that Kiln has an indefinable mission. In fact, I defined that mission, in broad terms, as bringing distributed source control to as many developers as possible. We really do believe that distributed source control is that much better than classic solutions, like Subversion, that we're on an almost religious drive right now to try to spread that to as many people as possible.

What Kiln doesn't have is a canonical mission statement. I think mission statements are a bad idea for two reasons: first, you can just go too narrow. If I'd defined Kiln's mission statement when we got started as 'be the best code review system,' we would have made an entirely different product. I don't think it would've been bad, but I don't think it'd be as interesting or useful as what we actually ended up delivering.

On the flip side, whether your mission statement is broad or specific, it can end up sounding trite and becoming meaningless after a while. You end up just parroting it back without really thinking it anymore. I'd liken it to Orwell's lament about 'dying metaphors' phrases where people just regurgitated a metaphor without envisioning the picture it was trying to convey.

By not defining a word-by-word mission, my hope is we're forced to actually think about what we're doing, instead of just parroting something we've memorized. So far, that's worked very well for us.

RM: Did you have a big architectural picture of how Kiln was going to work before launching it, so you knew what the hard to solve areas were likely to be?

BP: Thanks to the code sprint, we had a good idea what scaled well and what didn't before we started on the 'real' version. Right from the get-go on that, we split Kiln up so that the disk-intensive parts would be on separate boxes than everything else, which honestly solved huge swaths of our performance issues before we got started. We've had to make a variety of tweaks to the architecture since then, but we managed to get everything basically correct from the beginning due to the prototype.

That said the prototype was completely instrumental in that realization. The prototype's design would have performed abysmally if we hadn't modified it. I think the real lesson here is, "Build a prototype. Smack it around. Find what doesn't work. Fix that before you design the real version."

RM: How does distributed source control make Kiln better to use? Why should people adopt it?

BP: There are so many things that distributed source control makes better. Having wonderful merging means that it's easy for us to maintain multiple versions of Kiln and FogBugz at the same time. Having trivial branching means that it's easy for us to do quick experiments, with full history, and then merge them later if they work out, or throw them out if they don't. The distributed nature of source control means that we can keep working from home or on the train when we want to, and the speed of the whole thing means that we can do these incredibly frequent commits of tiny changes that make figuring out what change actually broke the build a tremendous amount easier. These are all advantages to any team of any size. While any little piece may sound kind of boring, when they're combined, they make a tremendous difference to your development process. It's why you get people who are such evangelists of Mercurial, Git, and similar tools. They really feel that DVCS makes a palpable difference.

You get all of that with any DVCS. What Kiln provides on top is all the tools that make coordinating with a team a lot easier. Have all of your personal repositories and branches in one place. Have a way of seeing exactly what changes you have or have not accepted. Get a way to see which bugs are fixed in any given version. Use code reviews to make sure that you're not approving changes that haven't been vetted and improved upon.

I think literally every feature in Kiln is a direct result of some problem that we've had at Fog Creek with source control, so not using Kiln these days feels very weird to me. I can get lots of work done using vanilla Mercurial: it's a truly wonderful distributed system, even on its own. But it feels kind of like going from a whiz-bang modern graphing calculator to an HP-41.

RM: What was the biggest challenge?

BP: Oh, definitely getting all of the timeout issues working. Mercurial pushes are done as these single HTTP POST requests, and that's a real problem if you're looking at a big repository, like Firefox or OpenJDK, because it means you're trying to jam sometimes gigabytes of POST data into a single HTTP request. Web stacks just generally aren't designed to work that way. It took us months in the beta period to fully iron out all the things that needed to be changed, from our load balancers to the website to the backend to the client boxes.

The real challenges are rarely the interesting ones; the interesting ones are fun to solve, so they're fun to work on, and usually get

solved relatively quickly. It's the ones that just involve a pile of tedious testing of bad hypotheses that end up being the real killers.

RM: Ok, let's move on to a couple of general questions. How do you tackle understanding a piece of code that you didn't write? Do you just dive in and start reading it? How do you start?

BP: I know this is heresy, but, when possible, I like to start with a debugger. Find a button or link that looks interesting, figure out one thing that fires when that button gets pushed, attach a breakpoint, then push the link or button. Doing that allows you to quickly follow through the real flow of the program, including any behind-the-scenes voodoo that implicitly runs code and might be hidden by a cursory read of the source-code. It's also great for giving a hands-on view of how the designers intended the code to be put together.

Alongside that, or in place of it when there's no reasonable way to explore with a debugger, I'll usually poke around the source code, find a file that looks interesting, and poke through it. As I go, I'll constantly grep through the rest of the code to figure out who uses these functions or objects and where. If you pick a good candidate—say, Repo.cs in Kiln's source code, which ends up being in charge of everything to d with a repository—doing this will naturally expand out in a way that gives you a good narrative for how the program's put together.

RM: What are the characteristics that make code easier to read?

BP: Good names, as always. Name variables and functions consistently and appropriately after what they're supposed to do. If you mean them to be internal (for whatever 'internal' might mean on that language and framework—private variables, module-level variables, or the like), indicate that in some way in the naming so that I can quickly pick out the interface from the implementation.

Having short functions and classes helps a lot, too. I can only hold so much of a program in my head at any given point, so the higher level abstractions I can be using at any given point in the program, the more I can reason about at once. That means better optimizations and faster debugging.

And finally, and this is one of the biggest in my opinion these days: magic is bad. Don't use super-clever tricks to mask what your code is doing. While that makes code superficially easy to read, it also does a wonderful job masking bugs and making understanding the actual control flow of what you're reading extremely complicated.

RM: Speaking as a programmer how have your ideas about language design and software changed over time?

BP: My attitude towards language design hasn't changed too much. For better or for worse, Smalltalk remains my biggest influence. I want languages that make functional-style code easy, by giving you lambdas or a similar construct. I prefer languages that are explicit over ones that are implicit or "magical." I like languages with relatively simple grammars, so that their syntax doesn't become its own source of bugs for your program. And I like some pragmatism, so while I like the error checking provided by static type systems, or the general safety provided by immutable variables in functional languages, I want the right to break those rules when I need to. I think Python, Go; CoffeeScript, C#, F#, and OCaml are all good examples of these guidelines, to greater or lesser extents.

RM: A few of the people I've interviewed such as L Peter Duetsch and Don Knuth compose and play music and you're something of a musician yourself. Is there a link between music and technology do such as source code versus music notation?

BP: Maybe. Personally, I think that's something of a red herring, though. Pretty much everyone likes music; bright people seem to be more likely to play music or compose it. I think whatever helps you be good at math or programming may help you be slightly better at composing or playing music, or even vice-versa, but I know too many good musicians who are lousy mathematicians to believe there's some type of cosmic underlying link going on here.

I certainly don't believe there's any underlying link in aptitude between source code and music notation. Music is aural. I'm sure some people compose on paper first, but I promise that playing with my friends happens on the instruments, not on paper, and while I reason about source code visually, I almost never visualize what the music I'm banging out would look like on paper while I'm playing. In other words: source code is a program; music notation is not music.

RM: How did you manage to trump The Social Network, the Zuckerberg biopic, by five years by appearing in a movie called Aardvark'd: 12 Weeks with Geeks? What was it about and has Columbia Pictures been in touch about optioning it?

BP: Twelve Weeks with Geeks tracked the summer where we built Copilot. Three other interns and I got brought in by Joel to build this new product entirely from scratch. We started off the summer with nothing but a 40-page Word document, and somehow finished twelve weeks later with a real product that was immediately useful, and began making some actual revenue. In this sense, I guess, it's the opposite of The Social Network: the Winkelvii (played by Joel and Michael) came to us and told us what they wanted, we built it quickly, and we immediately started making money.

I'll be honest: I absolutely hated having that Aardvark'd made. It's hard enough trying to make a product from nothing to selling in twelve weeks with three guys you until now had no working relationship with. It's even harder when there's a camera in your face constantly, recording every lame joke you tell and every idiotic thing you do.

But in retrospect, I love having Aardvark'd around, because it's absolutely hilarious. A great drinking game is to take a shot every time I'm a jerk in the film. Unfortunately this game is probably directly responsible for most of my friends having sclerotic livers at this point. Thankfully, I've been told by reliable sycophants that I'm no longer a jerk, so I can just sit back and laugh at how I've grown up and move on.

Amazingly, Columbia Pictures did initially express interest, but ended up deciding to do a documentary on a banana stand owned by a real estate company in Orange County instead. I don't honestly know what happened with that; they may have arrested development of the project.

RM: Final question Benjamin: you're writing a novel. Is that for fun or has it been commissioned and what is it about?

RM: Oh, that's just for fun. I love to write in general, and have always written for fun. When I was in college, I briefly flirted with the idea of writing screen plays, so I wrote a few of those, realized that my chance of making it in LA was effectively zero even if they happened to be good, and gave up on that idea. But one screen play, which I actually wrote well after I graduated, made a pretty good story, so I've been lazily working on novelizing it.

The story is basically an exaggerated version of a kind of soul-searching mission I went on back in 2007. I'd just gotten out of a crappy relationship, and was having some temporary burnout at work, so my best friend pretty much dragged me to JFK and shipped me out on the first flight to San Francisco and told me to do just about anything except code for a week.

I spent the week hiking, visiting old friends, and talking with people in the Valley about how the scene looked like out there. I came back feeling tremendously refreshed, but with a tremendous pile of hilarious stories and antics. I'm basically taking those stories and wrapping them in a plot based loosely on friends and acquaintances I had who were in banking got laid off when the mortgage crisis hit a head.

I think it's an interesting story that will successfully sell at least two copies, provided my mom thinks it's okay. Or maybe it'll outsell Harry Potter. Who knows? One of the great things about just writing as a hobby, and for fun, is I don't have to worry about that. Combine that with options like Kindle and Lulu, and I even have the option of publishing if I feel like it's good enough. But, honestly, I write because I love writing. The end. If something great happens because of that, it'd be wonderful, but I'm not going to be upset if nothing comes of it.

A shameless plug for the SQL community

Published Thursday, May 12, 2011 2:00 PM

Last night I made my debut as a speaker at a user group meeting in Southampton, UK. As I have mentioned before, I have been invited by Red Gate to talk at their SQL in the City event in July so I was looking for an opportunity to practice with a smaller audience so when the user group option came up I was keen to give it a shot.

The fact that the Southampton user group was this week is also fortunate as next week I am running the first meeting of the SQLSouthWest user group and I learned as much watching how the guys ran the meeting last night as I did about speaking when it was my turn.

First speaker of the evening was Kev Chant (Twitter) who gave us an introduction to Log Shipping and how it varies between different version of SQL Server.

I was next, giving my feelings about how 3rd party backup solutions really help a busy DBA. Having used SQLBackup for many years now I have had a lot of time to become familiar with the ways it helps me keep a handle on all the back ups/restores, log shipping etc. so the content was the easy bit. I floundered around on my wholly under-powered laptop and didn't really do the demo's justice. The slide deck was OK, except that I had only installed the PowerPoint Viewer so all my notes were invisible!

The evening was rounded off by Andrew Fryer (Twitter) who is a Technical Evangelist for Microsoft with some very interesting information about data centre technology.

Mark Pryce-Maher (Twitter|Blog) runs the Southampton user group and I am sure he'd love to hear from you if you'd like to join in with one of his meetings, they are a really nice group of SQL users.

Now on to the shameless plug. It all started at SQLBits in York last year and I bravely stated my intentions here. It's been a lot of work with lots of rejections and dead ends on the way but we are now about to have our first meeting. It is the first meeting of the SQL South West user group next week - 18th May. All the details are available at www.sqlsouthwest.co.uk, please come along if you can.

Please also visit www.sqlinthecity.com and register for a great event in July that is wholly free to attend.

by fatherjack
Filed Under: community

Remote Debugging in Visual Studio: Squashing Bugs in their Native Environment

05 May 2011
by Clive Tong

There are already third-party applications that help you to debug .NET applications remotely, or at least get enough information to diagnose the fault successfully. However, VS2010 allows you to debug remote applications. Is this the ideal way of doing it? How do you set it up? What are the pitfalls? Clive Tong explains.

As a developer, you get to write your application in a little sandbox that's isolated from the real world. You've configured your machine the way you like it, you've installed all manner of add-ins into Visual Studio, you've added a number of seemingly random assemblies to your global assembly cache. Then, you're totally disappointed when you copy the executable to another machine and it crashes.

Of course, it's not really quite like that. Build systems try to ensure that the hidden dependencies are brought to the surface, and large collections of untainted virtual machines allow you to quickly test an application in a clean room environment. Still, it's impossible to test all possible configurations, and so all too often bugs turn up. In those scenarios, it would be nice get access to the customer's machine. Using some of the modern support technologies such as NTRsupport, this is fairly easy to do, and by using WinDbg and the SOS Debugging Extension it is possible to install very little on the target machine while still being able to get enough data to diagnose the problem.

Recently, someone pointed out to me that Visual Studio has an alternative for debugging remote applications, which ought to make it a lot easier to debug problems in .NET applications. It would be nice if all that was necessary was to select Debug, Attach to Process in Visual Studio on the host machine, and then enter the name of the target machine.

Of course, it isn't really quite that easy…

Making the target machine available for remote debugging

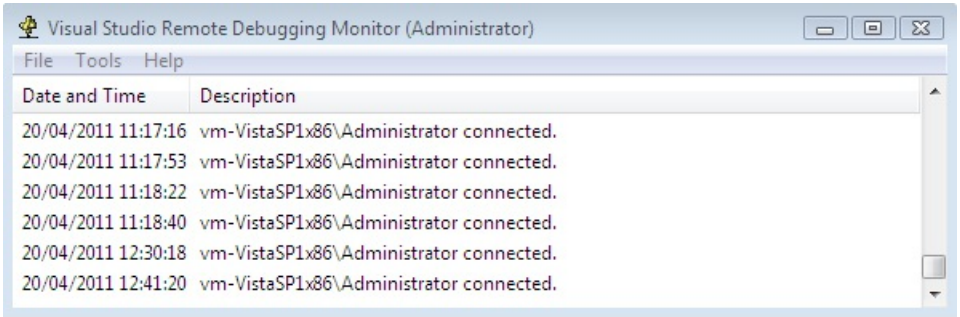
Before you can get started, the target machine has to make itself available for remote debugging.

The host machine talks to the target machine via DCOM in order to get the data it needs for debugging the application. The way to set this up is to run the appropriate version of msvsmon.exe, which you can find in the architecture-specific subdirectory:

```
\program files\microsoft Visual Studio 10.0\Common7\IDE\Remote Debugger
```

This executable is a DCOM server that handles the requests from remote clients, attaching to the relevant processes and getting data from them.

When the Debugging Monitor is running, it displays a view of the connections as they come in, as shown below. This is useful for debugging connection failures.



Search location for pdb files and sources

So the target machine is now available for remote debugging, but we're still not ready yet.

With .NET applications, even though the debugger is running on the host machine, it will look for pdb files on the target machine. Most importantly, if the symbol path is set on the host machine, it will still look for files on the symbol path, BUT they will be searched for in those locations on the target machine.

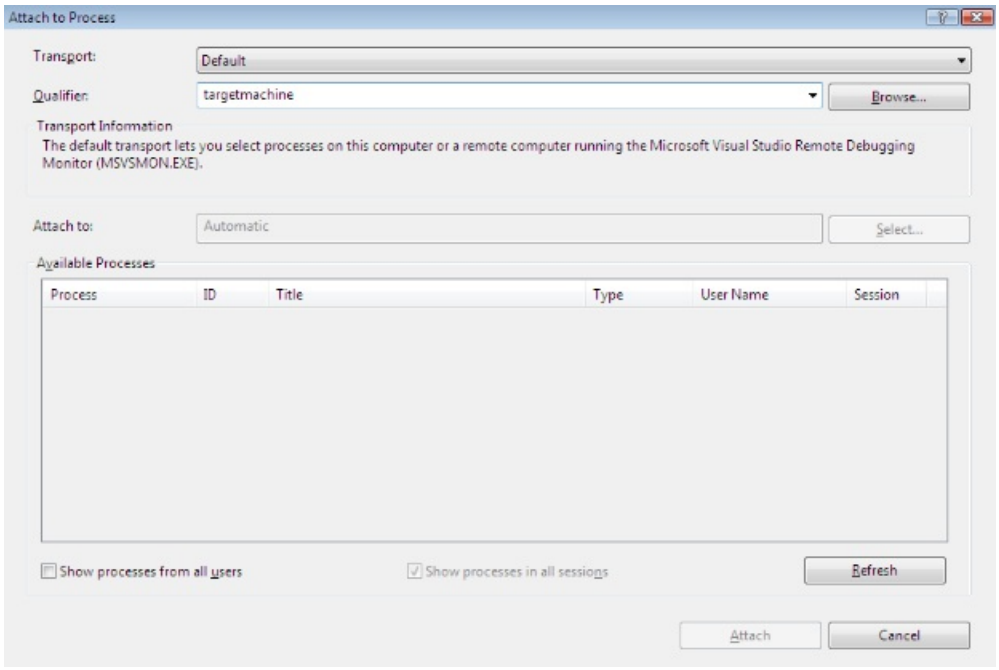
When I was experimenting with this, I wanted to use Reflector VSPro to step into the .NET framework on the target machine. Reflector VSPro stores the generated pdb files in numbered subdirectories under the user's `AppData/Local/Red Gate/.Net Reflector 7/Cache` directory. To get remote debugging to work, I had to copy `mscorlib.pdb` and the pdbs for my test application into the same path on the target machine (or rather, I needed to put the `mscorlib.pdb` in the same place; the two pdbs for the other parts of the application also worked if they were in the same folder as the application I was running). Note that I didn't need to copy the sources, which are looked up on the host machine. Of course, using an identical path can be tricky if you want to run as different usernames on the two machines, but one could imagine ways to make

the copying easier such as batch files or tricks with the file system.

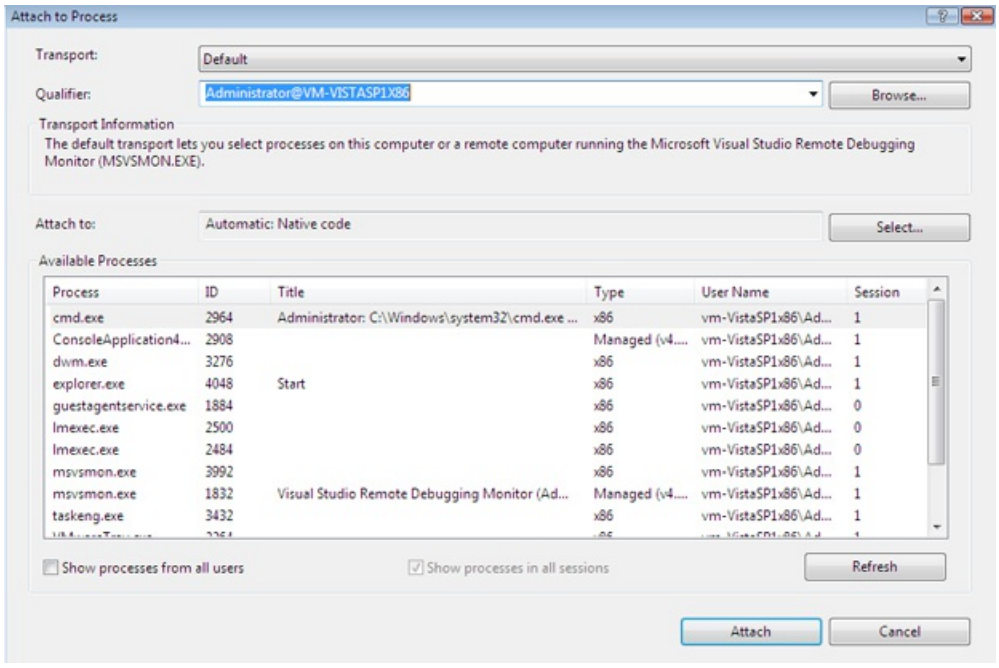
So, using the Visual Studio Debugger makes this all a lot harder than the ideal case where you could simply have all of your VMs running the remote debugger service and then seamlessly connect to debug them.

Debugging

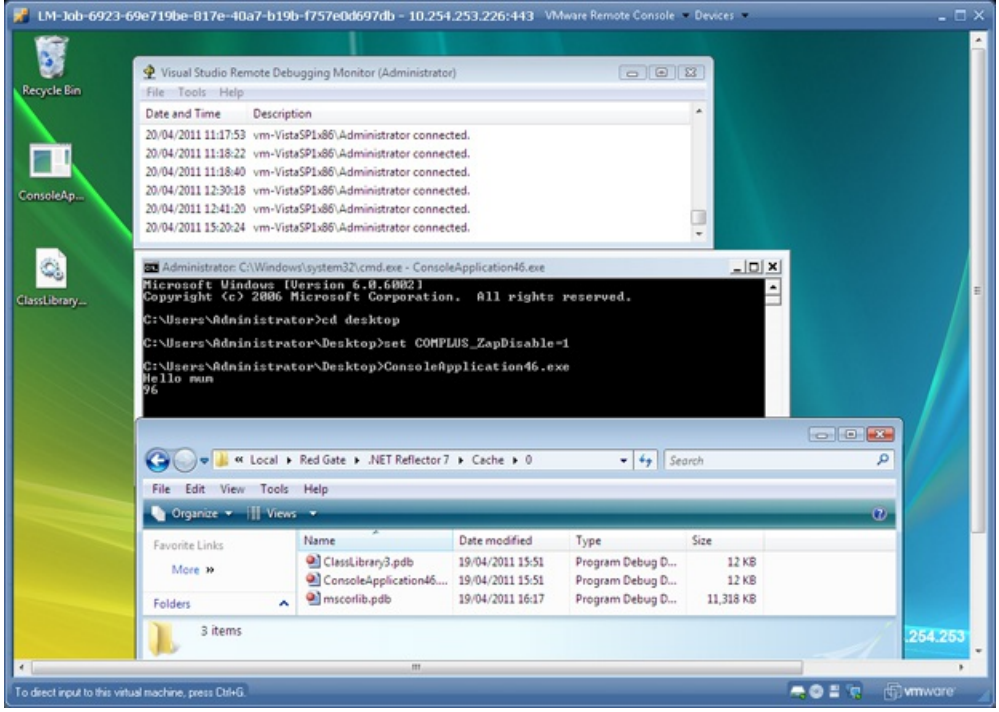
When you've set all this up, you are finally ready to open **Debug, Attach to Process** in Visual Studio, as shown below. In the screenshots in this article, I am running two instances of a virtual machine running Windows Vista, and I'm logged in as administrator on both machines. I used virtual machines so that I could configure their firewalls, something that I don't have permission to do on my usual work machine.



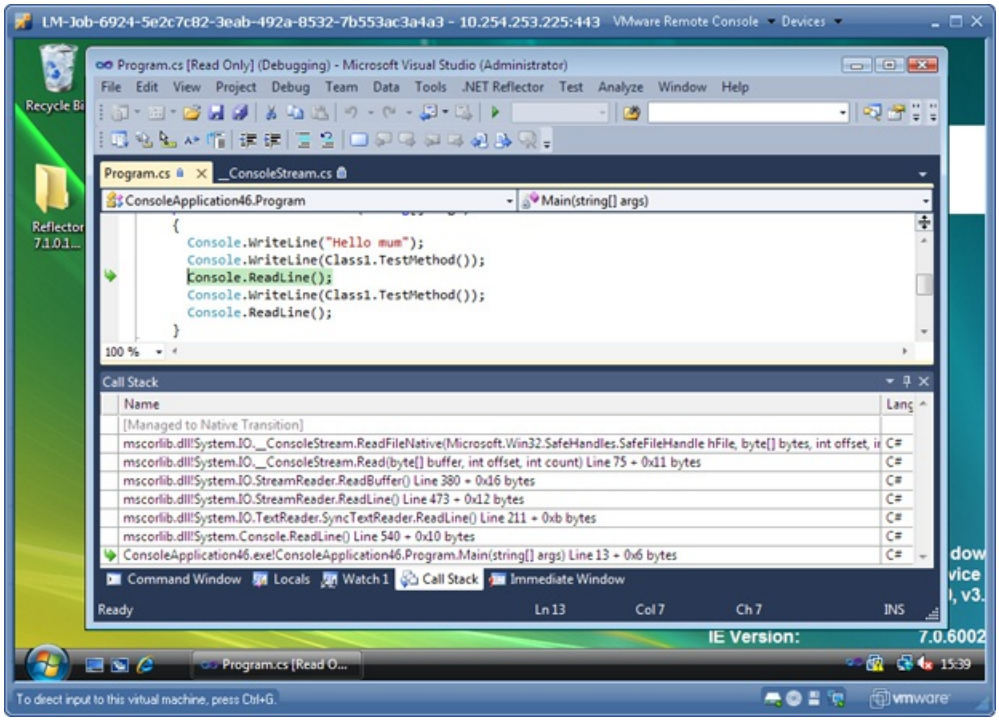
Enter the target machine name, and the available processes are listed. Notice that Visual Studio has substituted the entered name of the target machine (`targetmachine`) with the UPN (`Administrator@VM-VISTASP1X36`).



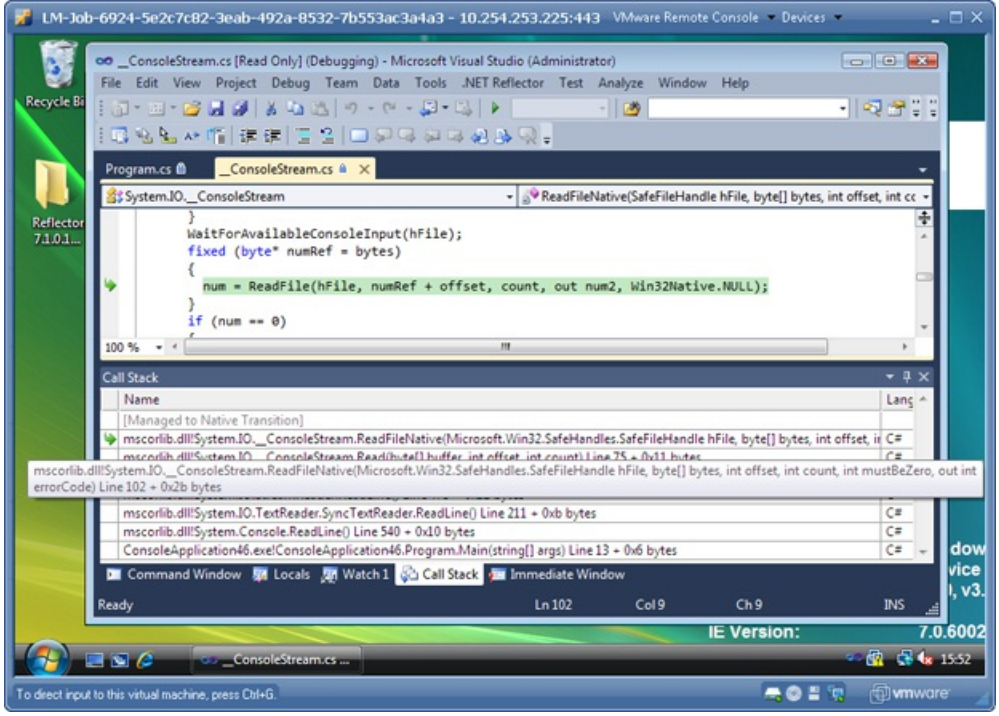
Once you have selected one of the available processes, you will be debugging the application that is running on the target machine. So, with the application running on the target machine:



We can step and debug as normal, though things are a little slower.



And, because we have set up the search location for pdbs and sources, we can look deeper into the Microsoft library (`mscorlib.pdb`):



Authentication of the Connection

We have seen that if you're on a Windows domain and want to run as the same identity on both the host and the target, then everything works really smoothly.

However, when I was experimenting with this, I was running as the administrator on two virtual machines that weren't connected to the same domain. This made things complicated because the two machines had the same name (they were clones of the same initial virtual machine), and this caused the DCOM reverse connection to fail. I had to rename the two machines, `sourcemachine` and `targetmachine` respectively, and for each, add the IP of the other machine to the hosts file in `c:\windows\system32\drivers\etc\hosts`:

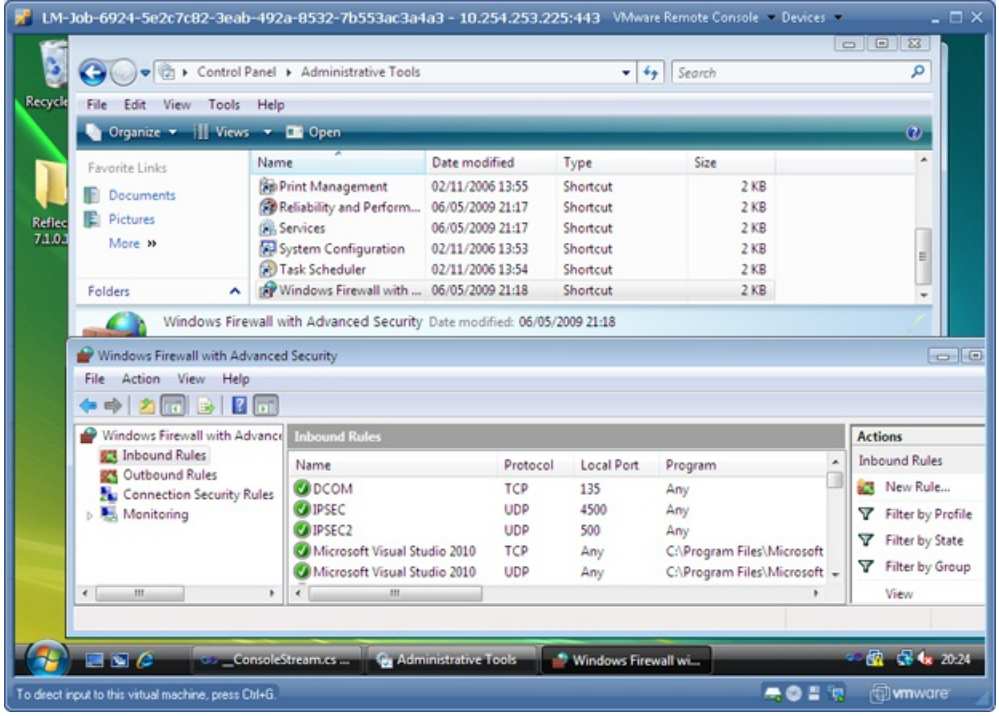
```
#       38.25.63.10     x.acme.com

127.0.0.1       localhost
::1             localhost
10.254.253.94   sourcemachine
```

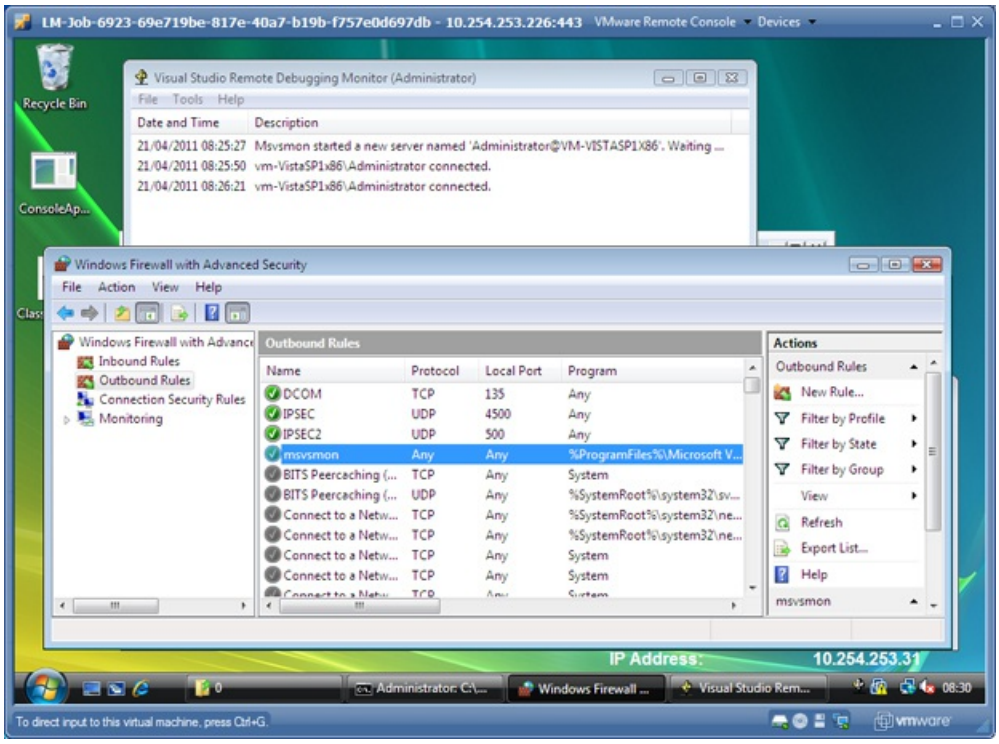
Jerome Hugon goes into some detail about setting up a suitable account to support the debugging in his blog post: Remote debugging.

Windows Firewall

There are various posts on the Internet detailing the permissions that you need to add to the Windows Firewall to allow enough traffic to pass through it to allow remote debugging. In my experiments I needed to allow TCP traffic on port 135, UDP traffic on ports 4500 and 500, and also explicitly allow the `devenv.exe` (Visual Studio) and the `MSVSMON.exe` applications on the relevant computer. My inbound settings are shown below:



And here are the outbound settings I used:



Personally, I'm not sure how confident I'd be punching holes in a firewall in order to enable remote debugging, and hence I don't think I'd use this technology for debugging across the Internet. I didn't have a good example to do any measurements, but you'd imagine that the debugging experience will be a lot slower when there is any distance between the host and the target.

Summary

It is definitely quite simple to use Visual Studio's remote debugging in some scenarios, in particular using it to debug an application running on a virtual machine that isn't behind a firewall. Given the amount of setup required to run it on a customer machine, I doubt that the technology is appropriate for debugging production applications, but it can certainly be useful in scenarios where you control all of the interacting machines.

Related Links

- How to Set up Remote Debugging
- Exploring Windows Azure - is it more than just hosting? by Alex Davies
- Visual Studio Remote Debugging and PDB Files by John Robbins

Don't feed the ducks: can we harness the power of nonsense?

Published Thursday, May 12, 2011 5:11 PM

A fair bit has been written about novelty error messages, particularly web 404 pages. Opinion seems broadly split about whether they're charming or unprofessional, but lots of them are pretty memorable.

In the cases of a 404 page, I'm not sure it's something you want people remembering, but the idea is probably to try and take the edge off a nasty or disappointing user experience. The memorability (because these things are, say, incongruous, or arresting, or funny) is a nice side effect for your brand, of course.

Yesterday, I saw this tweet by Gabriel Smy:

And it really bugged me. I think it's a fantastic idea.

It's either a deliberate joke, or a typo with "si**n**k" instead of "si**c**k"; and it's funny. It has that Fail Whale / Tumbeast memorability. It's also a kind of error message. "Don't feed the ducks" isn't a million miles from "Are you sure you want to move this file to the recycle bin?" It's a warning message, with an informational component, howbeit a daft one.

So how about it for tech comms? Gabriel's sinking duck sign works so well because the joke plays with expectations of rationality. The ridiculousness of the idea of ducks sinking unfolds after you've read and half-digested a superficially plausible message. Structurally, it's a good joke. And it sticks in the mind.

If we're trying to get an idea across that we really want somebody to remember, is it worth making it memorably ridiculous?

The counter-arguments are obvious: it's inappropriate for various brands, audiences, or situations, and it doesn't localize (or work well at all for non-native speakers). But somebody will say that about everything in tech comms. Much of the time they'll be right, too, and the problem devolves to how much we care.

I should work that out, because I'm itching to try this.

by Roger Hart
Filed Under: Technical communications, user assistance

TIME Gentlemen please! The SQL Server temporal datatypes

12 May 2011
by Joe Celko

> If you are still using the old Sybase DateTime datatype, it is a good idea to move your code to the more standard datatypes that were introduced in SQL Server 2008. Joe Celko explains why, and walks through some of the history of the TSQL way of storing and manipulating dates and times.

SQL Server 2008 brought the temporal offerings up to ANSI/ISO Standards. It is my feeling that the new temporal data types will displace the "Sybase/UNIX Code Museum that still dominates T-SQL code. We now have DATE and TIME data types with a full range and high precision.

To re-cap, for decades T-SQL has had the DATETIME data type. It is a crippled eight bytes implementation of what ANSI would call a TIMESTAMP. It combines both a clock time and date with a range from 1753-01-01 to 9999-12-31 with an accuracy of 3.33 milliseconds. That start date has to do with the history of the Gregorian Calendar. I will not get into that, but you might want to Google it when you have a little spare time. The 3.33 millisecond increment has to do with the early 16-bit hardware and the system clocks in those days.

T-SQL has an interesting proprietary temporal data type called SMALLDATETIME. It uses four bytes to hold time and date values ranging from 1900-01-01 to 2079-06-06 with an accuracy of one minute. For most commercial purposes, this range is just fine; very few companies track anything down to the second. Strangely, it is not used as much as DATETIME.

Now, about the reserved word TIMESTAMP. In T-SQL dialect it was an internal increasing numbering that that was used for version control and optimistic concurrency control. It is a table property and not a data type at all. That means each table may have only one TIMESTAMP and they cannot be NULL.

The good news is the T-SQL TIMESTAMP is deprecated. That it will be removed in a future version of T-SQL. This is one of the reasons I tell people to avoid dialect in favor of Standard SQL. In fact, now is the time to start cleaning up your old code before you get caught in a crunch.

Since the T-SQL TIMESTAMP is a table property, you do not have to specify a column name for it. You will get a horrible system generated name instead.

```
CREATE TABLE Foobar
(foo_id INTEGER NOT NULL PRIMARY KEY,
 TIMESTAMP);
```

The replacement for the old T-SQL TIMESTAMP is called ROWVERSION. It is almost a synonym for the old T-SQL TIMESTAMP, but must you specify a column name, for example:

```
CREATE TABLE Foobar2
(foo_id INTEGER NOT NULL PRIMARY KEY,
 row_ver ROWVERSION);
```

The bad news is that you can get duplicate ROWVERSION values by using the SELECT INTO statement in which a ROWVERSION column is in the SELECT list. This is not recommended, even tho it follows a set-oriented model of insertion. Since these things are meta-data and not temporal, this is all I am saying about them.

We now have a DATETIME2(n) data type. Microsoft stole this postfix digit 2 from Oracle and their VARCHAR2() data type. A DATETIME2(n) uses between six to eight bytes to store dates and times as a single unit (the ANSI/ISO TIMESTAMP). The date range is between 0001-01-01 and 9999-12-31 with an accuracy of one hundred nanoseconds (seven decimal places). The FIPS-127 (Federal Information Processing Standards) specs require at least five decimal places for seconds. The nice part is that you can adjust the decimal places in the declaration. A zero will give you whole seconds, and seven will give you maximum precision.

```
CREATE TABLE Foobar3
(foo_id INTEGER NOT NULL PRIMARY KEY,
 foo_date DATETIME2(7) DEFAULT CURRENT_TIMESTAMP NOT NULL);
```

DATE Data Type

The DATE data type use three bytes to store a date only (between 0001-01-01 to 9999-12-31. This is one of the nicest things we have gotten in T-SQL in a long time. When T-SQL only had DATETIME, we spent lots of computing time trimming the time fields to '00:00:00' for comparisons and grouping. This also messed up indexing by putting temporal columns in functions calls.

[Before anyone jumps on me, the ANSI/ISO Standard uses the term fields for the parts of a temporal data types.]

The rounding error in DATETIME also made it hard to use the BETWEEN predicate with temporal ranges. We had to write things like this to ensure we got all of fractional seconds in New Year's Day.

```
@my_datetime >= '2011-01-01' AND @my_datetime < '2011-01-02'
```

now we can use this for a single day

```
CAST (@my_datetime AS DATE) = '2010-01-01'
```

or get a range with.

```
CAST (@my_datetime AS DATE) BETWEEN '2011-01-01' AND '2011-01-02'
```

Notice how natural and simple the CAST(<exp> AS DATE) is to use. The worst way to handle dates was to use the proprietary CONVERT() function, turn the temporal data into a string, edit the string and then cast it back to temporal data. The ANSI/ISO Standards use only one data display format; the ISO-8601 "yyyy-mm-dd" with dashes. This is also what the DATE data type defaults to. This standard also shows up in many other ISO Standards, so it would be a good idea to get rid of your old local dialect dates.

The proprietary DATEADD() function will work with DATE values, but the units have to be YEAR, MONTH or DAY. No, it will not convert 24 hours into a day for you.

TIME Data Types

The TIME data types use between three to five bytes to store a time of day to an accuracy of 100 nanoseconds. Now, we have some language problems when we talk about time. We can talk about a point in time or in a day ("Meet me at 15:00 Hrs."), a duration ("The concert lasted 2 hours.") or an interval ("The movie starts at 15:00 and ends at 17:30 Hrs").

The ISO model of time is a continuum, so you have to use half-open intervals. That means a day starts exactly at 00:00:00 Hrs (midnight) and ends at 23:59:59.999.. Hrs without every getting to the start of the next day. This means that 24:00:00 Hrs today is really 00:00:00 Hrs tomorrow. DB2 will accept this convention and increment the date in a Standard timstamp value; T-SQL will throw an exception.

The proprietary DATEADD() function will work with TIME values, but the units have to be SECOND, MINUTE or HOUR. No, it will not convert a day into 24 hours for you. But the time of day will cycle around if you exceed 24 hours:

```
BEGIN
DECLARE @test_time TIME(0);
SET @test_time = '12:00:00';
SELECT DATEADD(SECOND, 1, @test_time),
    DATEADD(HOUR, 20, @test_time),
    DATEADD(HOUR, -20, @test_time);
END;

Results
================================
12:00:01    08:00:00    16:00:00
```

The handiest trick I have found so far for TIME is to build a reporting range table of time slots in day. For example, if you wanted to set up time slots of one minute, you only need (60 minutes per hour * 24 hours per day) = 1440 rows. That is small enough to fit into main storage.

```
CREATE TABLE Timeslots
(timeslot_nbr INTEGER NOT NULL PRIMARY KEY,
 timeslot_start_time TIME(1) NOT NULL,
```